Eredmények kiértékelése, mérése, kombinálása

Készítette: Hadházi Dániel

Tanulási problémák

- Ellenőrzött (osztályozás, predikció):
  - Adott bemeneti mintákhoz elvárt kimenetek.
- Félig ellenőrzött (valódi problémák):
  - Nincs minden bemeneti mintához ellenőrzött kimenet, de minden mintát felhasználunk.
- Megerősítéses problémák (szabtech):
  - Döntésszekvenciák kumulált minősítése lehetséges csak, viszont az egyes döntéseket kell tanulni.
- Nem ellenőrzött (klaszterezés):
  - Nincs egyetlen tanító mintához sem elvárt kimenet.

(Félig) Ellenőrzött eset

- Cél a bemeneti változók, és a célváltozó értéke közötti leképezést megtanulni.
- Példa:
  - Legyen adott egy osztályozó, ami (egyetemi hallgatót játszva) memorizálja a megadott mintákat.
  - Elvégzi az osztályozó a kitűzött célt?
  - NEM

Eredmények kiértékelése- irányelvek

- Tanítópontok helyesen osztályozása nem releváns bonyolult osztályozóknál.
- Mi az, ami számít:
  - Mennyire sikerült jól megtanulni a problémát.
  - **Mennyire képes általánosítani.**
- Hogyan mérjük mindkettőt:
  - Lehetőleg **minél több**, az osztályozó / regresszor kialakításánál **fel nem használt, valódi eloszlást reprezentáló minták** osztályozási arányával.

Eredmények kiértékelése

- Alapprobléma: csak véges minta áll rendelkezésre, melyből még tanulni is kell.
- Gyakran tanuló eljárások paramétereinek hangolására használjuk:
  - MLP: bátorsági tényező, struktúrák validálása
  - SVM: Hibásan osztályozási költsége (C),
    Érzéketlenségi sáv megválasztása (ε)
    Kernel leképzés hiperparaméterei.

Kiértékelés bizonyossága

- Bernoulli eloszlással modellezzük az egyes minták helyesen osztályozását (p):

$$\Pr\left[-z < \frac{f-p}{\sqrt{p(1-p)/N}} < z\right] = c$$

  - c: konfidencia szint
  - $2 \cdot z(c)$: konfidencia-intervallum szélessége a standardizált normális eloszlásnak.

Kiértékelés bizonyossága

Konfidencia intervallumok szélessége:

f=0.75	C=0.9	C=0.8	C=0.6
N=100	0.128	0.044	0.014
N=1000	0.103	0.035	0.011
N=10000	0.070	0.023	0.007

f=0.9	C=0.9	C=0.8	C=0.6
N=100	0.079	0.029	0.010
N=1000	0.065	0.023	0.008
N=10000	0.045	0.016	0.005

Az előző dián lévő normalizálás csak nagy N esetén állít elő standard normális eloszlást!

Kereszt kiértékelés

- Alapprobléma: Adott **véges mintakészlet alapján** szeretnénk osztályozók szeparálási képességeit minősíteni.
- Osszuk fel a tanítóhalmazokat két részre:
  - **Releváns** mintákból álljon mind a tanító, mind a validáló halmaz.
  - Ezt így igen nehéz lenne megtenni, ezért inkább kereszt kiértékelünk (Cross Validation).

Kereszt kiértékelés

- Osszuk fel *k* diszjunkt részhalmazra a tanító mintákat:
  - Minden kimeneti osztály mintái az egyes partíciókban kummulált relatív gyakoriságuknak megfelelő számban forduljanak elő.
  - A partíciók méretei azonosak legyenek.
- Tanítsunk k darab osztályozót:
  - i-edik esetén az i-edik partíció alapján validáljunk, a maradék pontkészlettel tanítsunk.

Kereszt kiértékelés

- Elrettentő példa:
  - Egy teljesen **zajszerű mintahalmaz**t kell megtanulni: független a kimeneti és a bemenet.
  - Minták **50%-a pozitív, 50%-a negatív**.
  - Osztályozó: minden bemenetre a tanítóminták közül a gyakoribb besorolását választja
  - **Leave-one-out**: minden esetben hibás a validáló elem besorolása => **0% becsült pontosság**
  - **Valójában 50%** a pontossága

Kereszt kiértékelés

- Alkalmazási példák:
  - 10 fold Cross-validation
  - Leave-one-out

0.368 Bootstrap

- Véletlen, visszatevésen alapuló kiválasztással állítsuk elő az N méretű tanítóhalmazt.
- A kimaradó mintákkal teszteljünk.
- Annak a valószínűsége, hogy az i-edik minta kimarad:

$$\left(1-\frac{1}{N}\right)^{N} \approx e^{-1} = 0.368$$

0.368 Bootstrap

- Nem érdemes csak a validációs halmaz alapján minősíteni:
  - Várhatólag csak a minták 63.2%-át látta a tanuló algoritmus.
  - **err=0.632×err(teszt)+0.368×err(tanító)**
- Végezzük el többször a 0.368 bootstrapet, majd annak átlaghibájával minősíthetjük az osztályozót.

0.368 Bootstrap

- Elrettentő példa:
  - Ugyanaz a mintahalmaz, mint amit a Kereszt kiértékelésnél közöltem (kimenet a bemenettől független, 50-50% a mintaosztályok rel. gyakorisága)
  - Az osztályozó memorizálja a tanítómintákat, új mintákra véletlenül dönt: err=0.632×0.5+0.368×0
  - hibásan osztályozás becsült valsége: 0.316
  - Valódi valószínűsége: 0.5

Mennyire jó a kialakított osztályozó?

- Eddig azt néztük, hogyan lehet becsülni egy osztályozó képességeit, de ez mennyire pontos?
- Student t-próbához nyúlunk:
  - *k*×-os kereszt validációval tanítsuk és minősítünk
  - $\frac{\mu_X - \mu}{\sqrt{\sigma_x^2 / k}}$ egy **k-1 szabadságfokú Student** eloszlás
  - *k* növelésével közelíti a std. normál eloszlást

Mennyire tér el egymástól két osztályozó?

- *k*×-os kereszt kiértékeléssel alakítsuk/ minősítsük az osztályozókat (ugyanazon tanító/teszt felosztással
- x_i, y_i jelölje a két osztályozó minősítését az i-edik kereszt kiértékelésnél.
- Vizsgáljuk a minősítések különbségét: $d_i = x_i - y_i$
- o várható értékű, 1 szórású **t statisztika** minősíti a o-hipotézist (u.a. eloszlás mintái):
- $$t = \frac{\mu_d}{\sqrt{\sigma_d^2 / k}}$$
- k-tól függ a t statisztika megbízhatósága.

Hibák minősítése

- Tanulás: osztályozók paramétereinek hangolása egy tanító mintakészlet alapján (nem csak).
- Hogyan működnek az osztályozók:
  - Adott bemeneti mintára osztályba tartozási valószínűségek küszöbölése szerint válaszolnak.
  - Pl. egy rosszul osztályozott mintánál számít, hogy a minta **besorolási bizonyossága** 51%, vagy 90%.
  - Definiáljunk veszteség fgv-eket!

Négyzetes hibaösszeg

- Szakirodalomban SSE, MSE.
- $Err = \sum_i \left(\varphi(x_i) - y_i \right)^2$
- Gradiéns alapú/ analitikus tanításnál használt hiba fgv példák (ezt minimalizáljuk):
  - Adaline – lin. regresszió $Err = \left(\mathbf{X} \cdot \boldsymbol{\theta} - \mathbf{y} \right)^T \left(\mathbf{X} \cdot \boldsymbol{\theta} - \mathbf{y} \right)$
  - MLP
  - LS-SVM
  - C/ε SVM, QP célfüggvénye is kvadratikus (megoldhatóság feltétele itt a konvexség)

Szélsőérték keresés kvadratikus felületen

- Vizsgáljuk a **konvex, kvadratikus** felületű függvényeket:
  - Egyszerű belátni, hogy az előző fejezetben definiált ***Err*** **konvex** felületű.
  - A lokális derivált -1 szerese minden pontban a globális minimum felé mutat.
  - Folytonos függvény -> Minden pontban létezik deriváltja , aminek iránya ellentétes a lokális szélsőérték irányával

Négyzetes veszteség függvény

- Vizsgáljuk meg a küszöbölés előtti állapotot általánosabb osztályozót feltételezve, egyetlen input mintára:
  - Minden kimeneti osztályra megmondja, hogy mekkora valószínűséggel eleme az adott osztálynak a minta ($a_j = \Pr\left[class == j \middle| x_i\right]$, $p_j^* = \mathrm{E}\left(a_j\right)$)
  - Valójában a kimenet egyetlen osztály.
  - Egy mintára a hiba értéke: $Err = E\left[\sum_j \left(p_j - a_j\right)^2\right]$
  - $Err = \sum_j \left(\left(p_j - p_j^*\right)^2 + p_j^*\left(1 - p_j^*\right)\right)$

Információ veszteség függvény

- $Err(x) = -\log_2\left(p\left(class = y \middle| x\right)\right)$
  - Bemenet: x, elvárt kimenet: y
- Logisztikus regresszió kritériumfüggvénye
- Információelméleti megközelítés: hány bitnyi információ szükséges a biztos, helyes döntéshez *p(class=y|x)* ismerete mellé.
- $\mathrm{E}\left[Err(x)\right] = \sum_j p_j^* \cdot \log_2\left(p_j\right)$
- $\underset{p_1, p_2, \ldots p_n}{\operatorname{argmin}} \left\{\mathrm{E}\left[Err(x)\right]\right\} = p_1^*, p_2^*, \ldots p_n^*$

Információ veszteség fgv

- Logisztikus osztályozók **minden** kimeneti osztálynak **pozitív valószínűséget** adnak minden bemenet esetén.
- Ellenkező esetben a hibafüggvényük értéke végtelen lenne, ha egy olyan példát látnak.
- Zero frequency problem:
  - Adott bemenethez tartozó kimeneti értékek eloszlásáról nincs információ a mintahalmazban.

Veszteségfüggvények összehasonlítása

- Négyzetes hibaösszeg:
  - A hiba értéke függ attól, mi a predikált eloszlás a helytelen alternatívák között: Tegyük fel, hogy az x bemenet az i-edik osztályba tartozik. Ekkor

$$\mathrm{Err}(x) = (1 - p_i)^2 + \sum_{j \neq i} p_j^2$$

  - De a maximális hiba korlátos: $\mathrm{Err}(x) \leq 1 + \sum_j p_j^2 \leq 2$

- Logisztikus hiba ennek ellentettje.

Osztályozás jellemzői

- 2 osztályos osztályozás esetén az **osztályozási mátrix**:

		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: érzékenység

- Érzékenység: TPR=TP/(TP+FN)

Érzékenység		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: szenzitivitás

- Specificitás:SPC=TN/(FP+TN)

Szenzitivitás		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: precizitás

- Precizitás:PPV=TP/(TP+FP)

Precizitás		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: negatív prediktív érték

- Negatív prediktív érték:NPV=TN/(TN+FN)

Negatív prediktív érték		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: Kihullás

- Fals pozitív arány:FPR=FP/(FP+TN)

Fals pozitív arány		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: fals riasztás arány

- Fals riasztás arány:FDR=FP/(TP+FP)

Fals riasztás arány		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

Osztályozás jellemzői: pontosság

- Pontosság: ACC=(TP+TN)/(TP+FP+FN+TN)

Pontosság		Előrejelzés	
		IGAZ	HAMIS
Valóság	IGAZ	TP	FN
	HAMIS	FP	TN

- Cél a főátlóbeli elemek maximalizálása.

Kappa statisztika

- Azt vizsgáljuk, hogy mennyivel pontosabb a vizsgált osztályozó egy olyannál, ami csak az domén mintáinak gyakorisága alapján becsül.
- Kappa score:

$$\kappa_{score} = \mathbf{1}^{\mathrm{T}} \cdot \left(\mathrm{diag}\left(\mathbf{C}^{\mathbf{real}} \right) - \mathrm{diag}\left(\mathbf{C}^{\mathbf{naiv}} \right) \right)$$

- Kappa arány:

$$\kappa = \kappa_{score} \Big/ \left(N - \mathbf{1}^{\mathrm{T}} \cdot \mathrm{diag}\left(\mathbf{C}^{\mathbf{naiv}} \right) \right)$$

Hibák súlyozása

- Eddigiekben uniform hibasúlyok voltak
- Miért is fontos a különböző hibák eltérő súlyozása?
  - Tehenészetben mért adatokból próbálták megállapítani, hogy melyik tehén mikor kezd menstruálni.
  - Tehenek menstruációs ciklusa az emberekéhez hasonlóan 30 napos -> **1/30 valószínűséggel** igaz, hogy egy tehén egy adott nap menstruált.
  - Az osztályozó a **„null" megoldással 97%-**os pontosságot ért el (ez így önmagában nagyon jó).

Hibák súlyozása

- Más példák:
  - **Üzembiztonságot felügyelő rendszerek** fals riasztása kisebb költségű, mint ha nem riaszt, amikor kellene.
  - Bank részéről megközelítve: **hitelezni** egy olyannak, aki nem fizeti vissza nagyobb veszteség, mint visszatartani a hitelt egy fals megbízhatósági besorolás miatt.
  - **Egészségügyi szűrések**nél fals negatív eset sokkal rosszabb a fals pozitívnál.

Hibák súlyozása szerinti döntés

- Példa:
  - **Uniform hibasúllyal** kialakítottunk egy **osztályozó**t.
  - **Utólag meghatározott** valaki egy hibákhoz kapcsolódó súlymátrixot.
- Megoldás (k osztályozós eset):
  - Predikált valószínűségeket páronként súlyozva hasonlítsuk össze, majd a páronkénti score-ok alapján válasszuk ki a legvalószínűbbet.
  - Problémás lehet, mert nem biztos, hogy tranzitívak a páronkénti súlyozott valószínűség különbségek

Hibák súlya szerinti tanítás

- Itt a tanítás esetén már rendelkezésre áll a különböző hibák súlya: **felhasználjuk tanításnál:**
  - Gépi tanulásnál legtöbbször az input osztályt tudjuk csak súlyozni.
  - MLP: minták ismételt használata 1 epochon belül
  - Aszimmetrikus C-SVM: megadhatóak osztály súlyok
  - Bayes háló: minták ismételt használata

Súlyozott hibájú osztályozók

- Értelem szerűen minősítésnél is vegyük figyelembe a hibasúlyokat.
- Legtöbbször kiegyensúlyozatlan tanítóhalmaz miatt használjuk:
  - Ha pl. illesztett szűrővel keresünk alakzatot egy képen, majd jellemzők alapján akarjuk a jelölteket osztályozni, akkor a $P/N<<1$ eset áll fel mindig
  - $\alpha*N/P$ legyen a P minták súlya, $\alpha \geq 1$.

Lift chart- példa

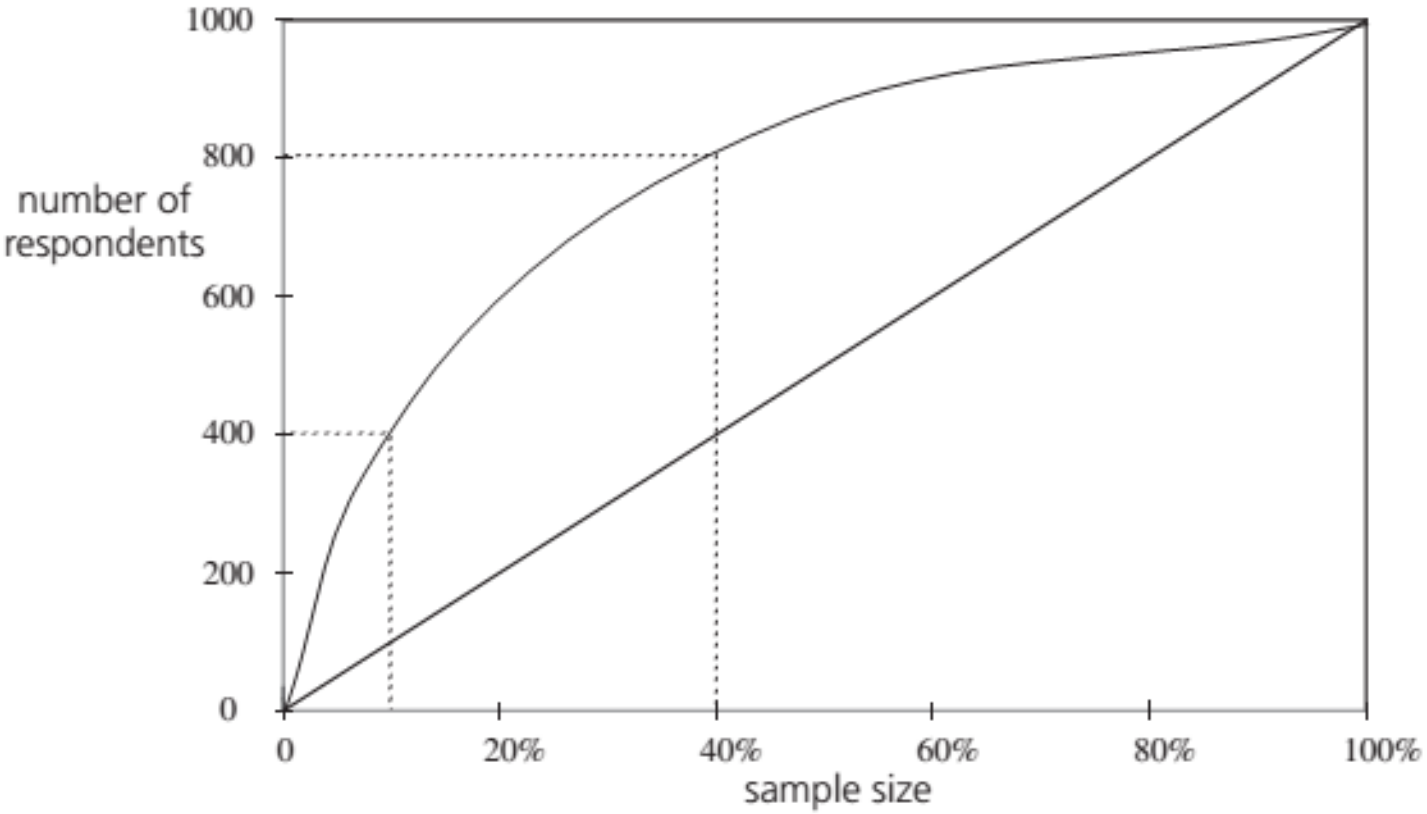
- Spammelik nyereményjátékkal a lakosságot
- Ismert súlyok:
  - Levél költsége (nyomtatás+ kézbesítés+ kenőpénz a hatóságnak, hogy adják ki a névjegyzéket).
  - Ha valaki részt vesz a játékban, az mekkora bevétel.
- T.f.h hogy létezik 100000 fő, akik 5%-a, míg a teljes lakosság 1%-a válaszol csak:
  - Az 100000 fős csoport **lift faktora 5**.

Lift chart példa

- Megoldás:

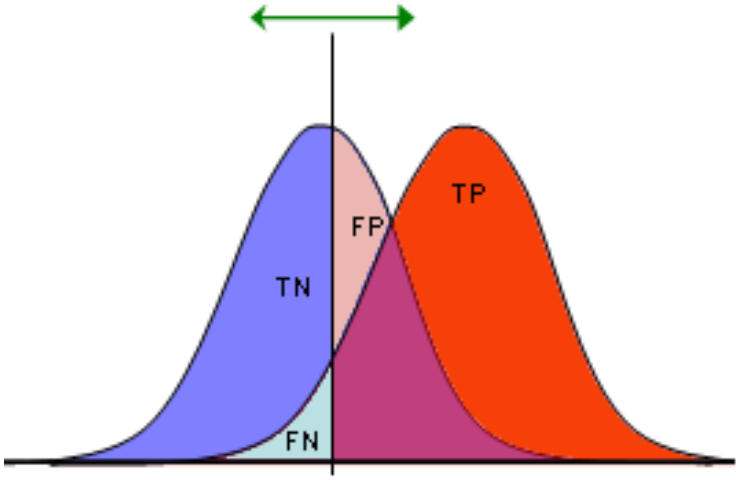
1. Vegyünk egy **mintahalmazt hasonló esetről** (emberek jellemzői a bemenetek – részt vett-e a kimenet).
2. Tanítsunk egy olyan **osztályozó**t, aminek részvételi **valószínűségek a kimenetei**.
3. Rangsoroljuk az embereket ez alapján, majd a **legjobb *k%*-nak küldjük ki** a leveleket.

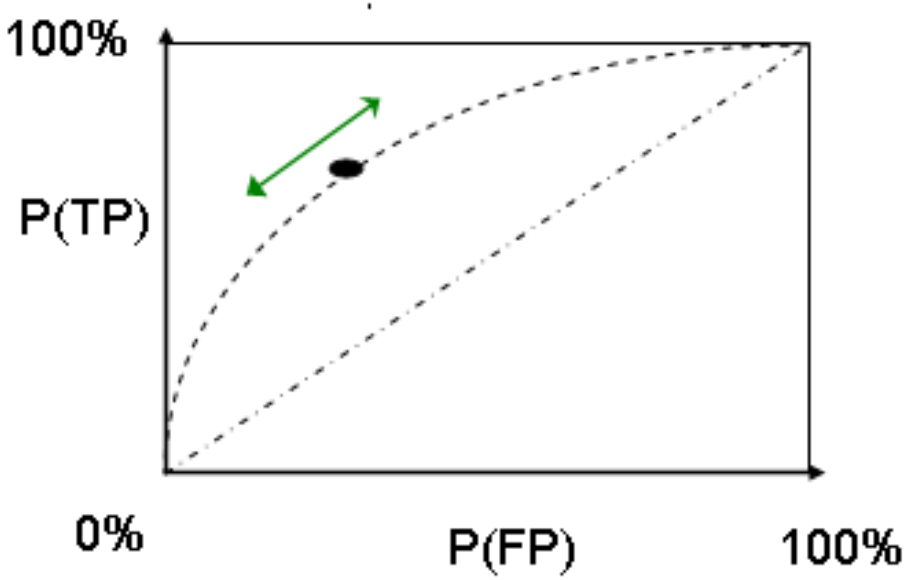
Lift chart görbe



ROC görbe

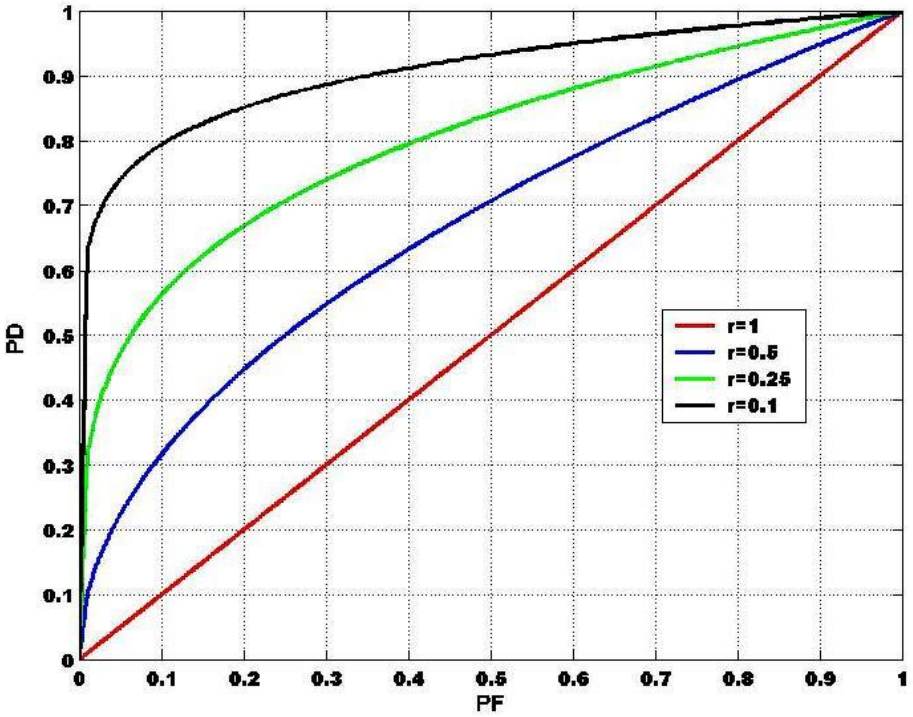
- Újfent valószínűségi kimenetekkel rendelkező 2 osztályos osztályozónk van.
- FPR=FP/(FP+TN) függvényében ábrázoljuk a TPR=TP/(TP+FN)-t.





ROC görbe

- Lehetséges osztályozók munkapont független rangsorolása:



ROC görbe

- Munkapont független minősítést (AUC):

$$AUC = \int_0^1 TPR(x)dx$$

- Munkapont kijelölése (hol küszöböljük a kiadott valószínűségeket):

$$\arg\min_k \left\{ TPR(k) \cdot C\left[-|+\right] + FPR(k) \cdot C\left[+|-\right] \right\}$$

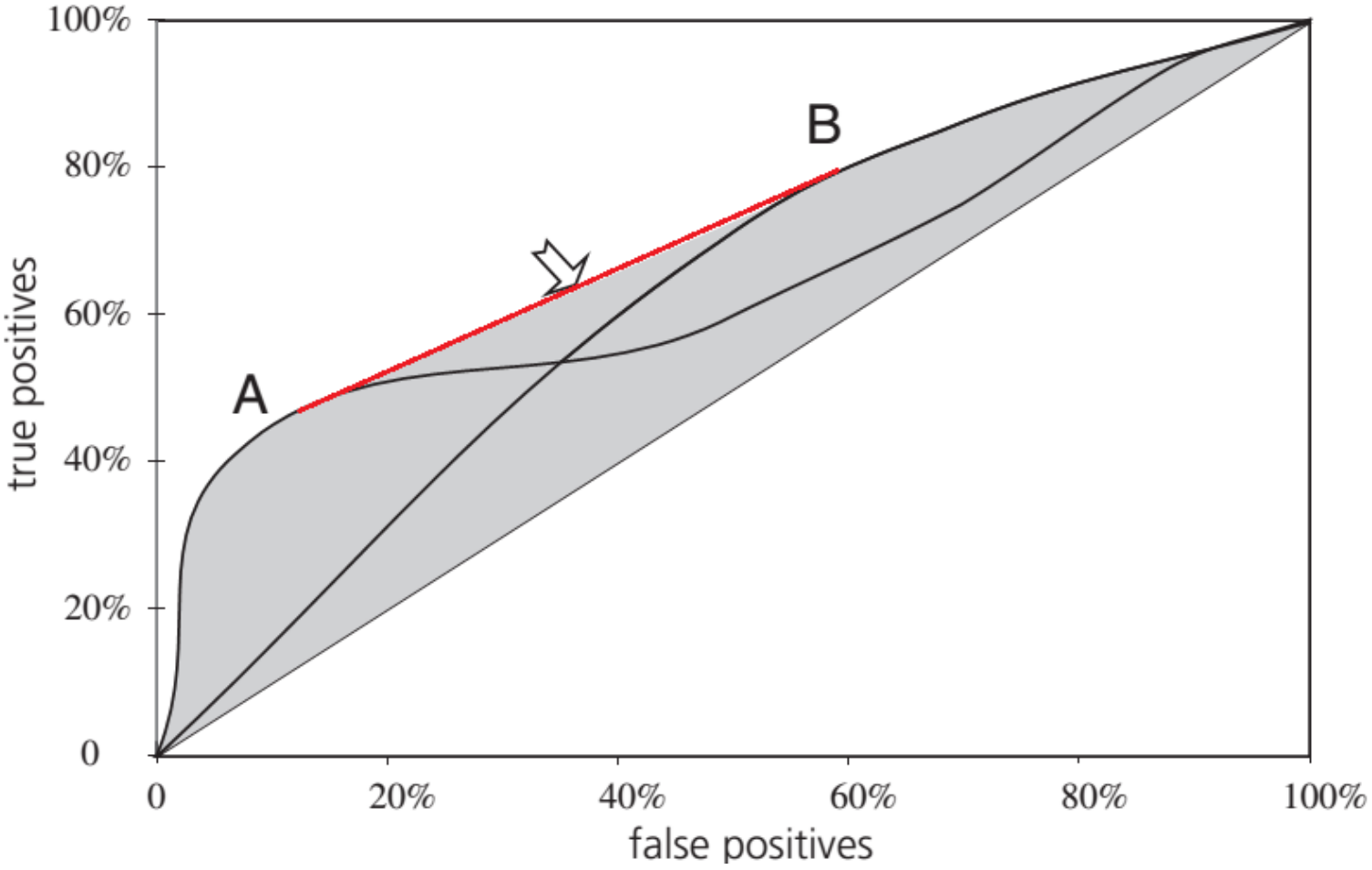
ROC görbe

- Tetszőleges munkatartományon belüli minősítés:

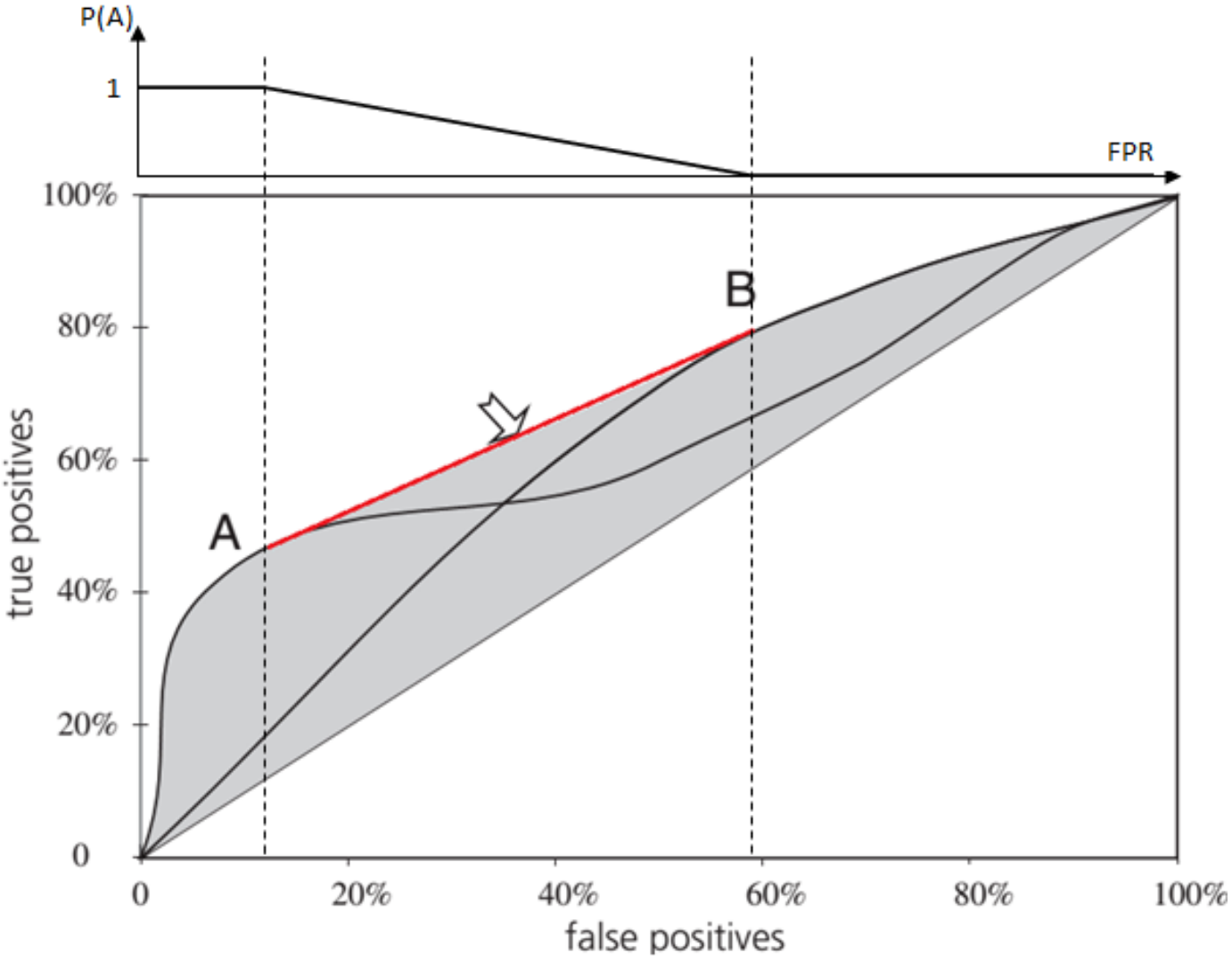
$$NAUC_{(FPR1,FPR2)} = \frac{\int_{FPR1}^{FPR2} TPR(x)dx}{FPR2 - FPR1}$$

$$NAUC_{(TPR1,TPR2)} = \frac{TPR2 - TPR1 - \int_{TPR1}^{TPR2} FPR(x)dx}{TPR2 - TPR1}$$

Osztályozók kombinálása



Osztályozók kombinálása



Recall-precision görbe

- **Keresőket minősítünk**:
  - Egyik 100 találatot ad, abból 40 releváns
  - Másik 400-at, amiből 80 releváns.
- **Recall:** releváns dokumentumok mekkora része eleme a találatoknak *(TP/(TP+FN))*
- **Precision**: a találatok mekkora része releváns
- Ez is ábrázolható grafikusan. *(TP/(TP+FP))*
- Munkapont kiválasztása talán ezzel a módszernél a legkézenfekvőbb.

Egyéb mérőszámok

- k pontos átlagos recall:
  - k=3 esetén 20%,50%,80%-os recall-hoz tartozó precision átlaga
- F-mérték:

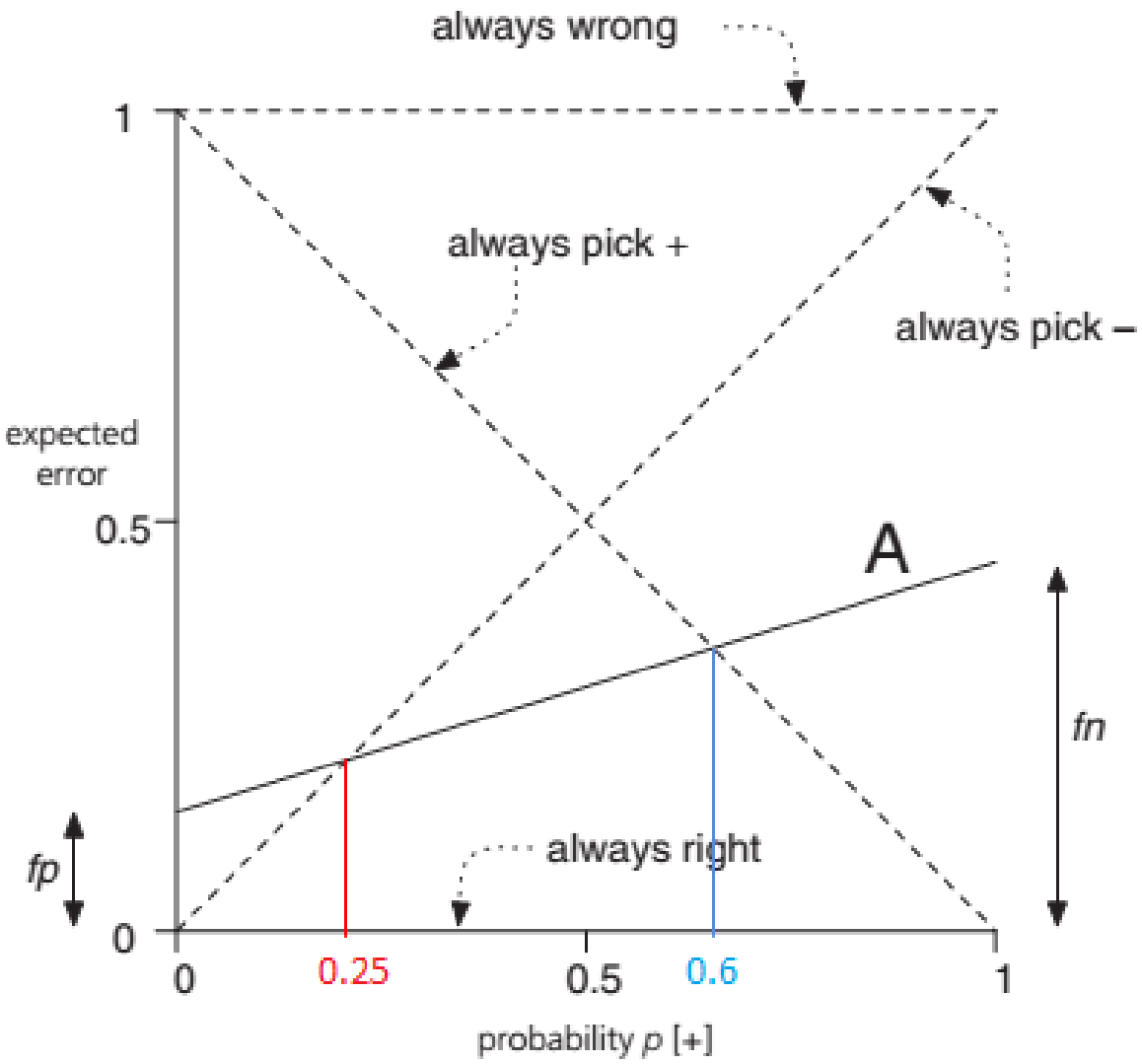
$$\frac{2 \cdot recall \cdot precision}{recall + precision} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN}$$

- ROC- AUC
- Helyesen osztályzási arány: (TP+TN)/(P+N)

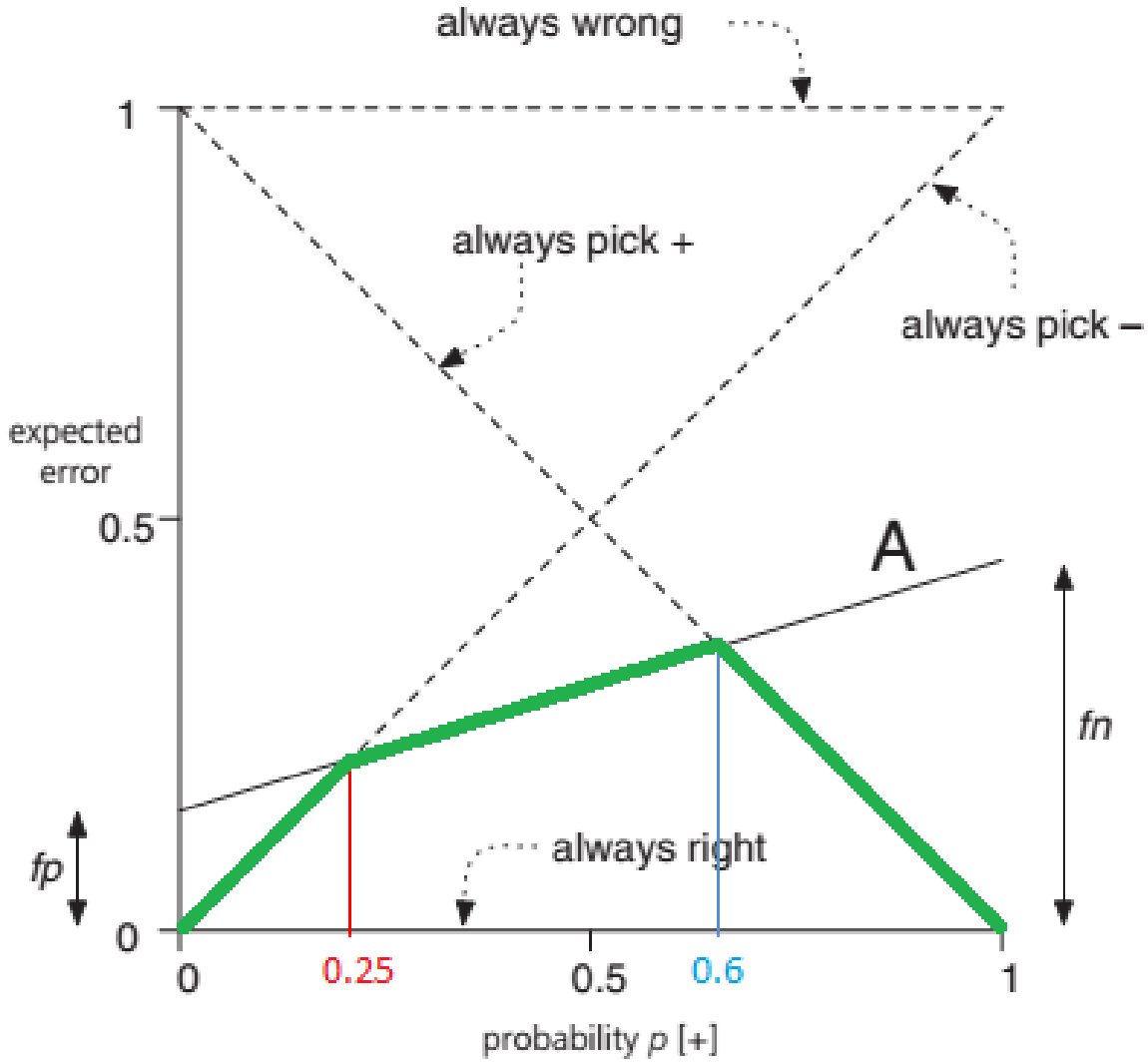
Költség görbék

- Eddigi görbék:
  - Hibasúly függő munkapont választás
  - De nem hibasúly függő kiértékelés
- Költség görbék:
  - **Egy osztályozóhoz** egy egyenest rendelünk: várható osztályozási hiba a teszthalmaz mintáinak eloszlásának a függvényében.
  - Tehát itt küszöböljük a predikált valségeket

Költség görbék példa



Költség görbék példa



Súlyozásos költség görbék

- Hol marad a hibák súlyozása?
- Valószínűségi költség: 0-1 között változik

$$p_C[+] = \frac{p[+]\cdot C[-|+]}{p[+]\cdot C[-|+] + p[-]\cdot C[+|-]}$$

- Normalizált várható költség:

$$\mathrm{nec}(p_c[+]) = fn\cdot p_c[+] + fp\cdot(1-p_c[+])$$

- C[+|+]=C[-|-]=0 feltételezéssel éltünk.

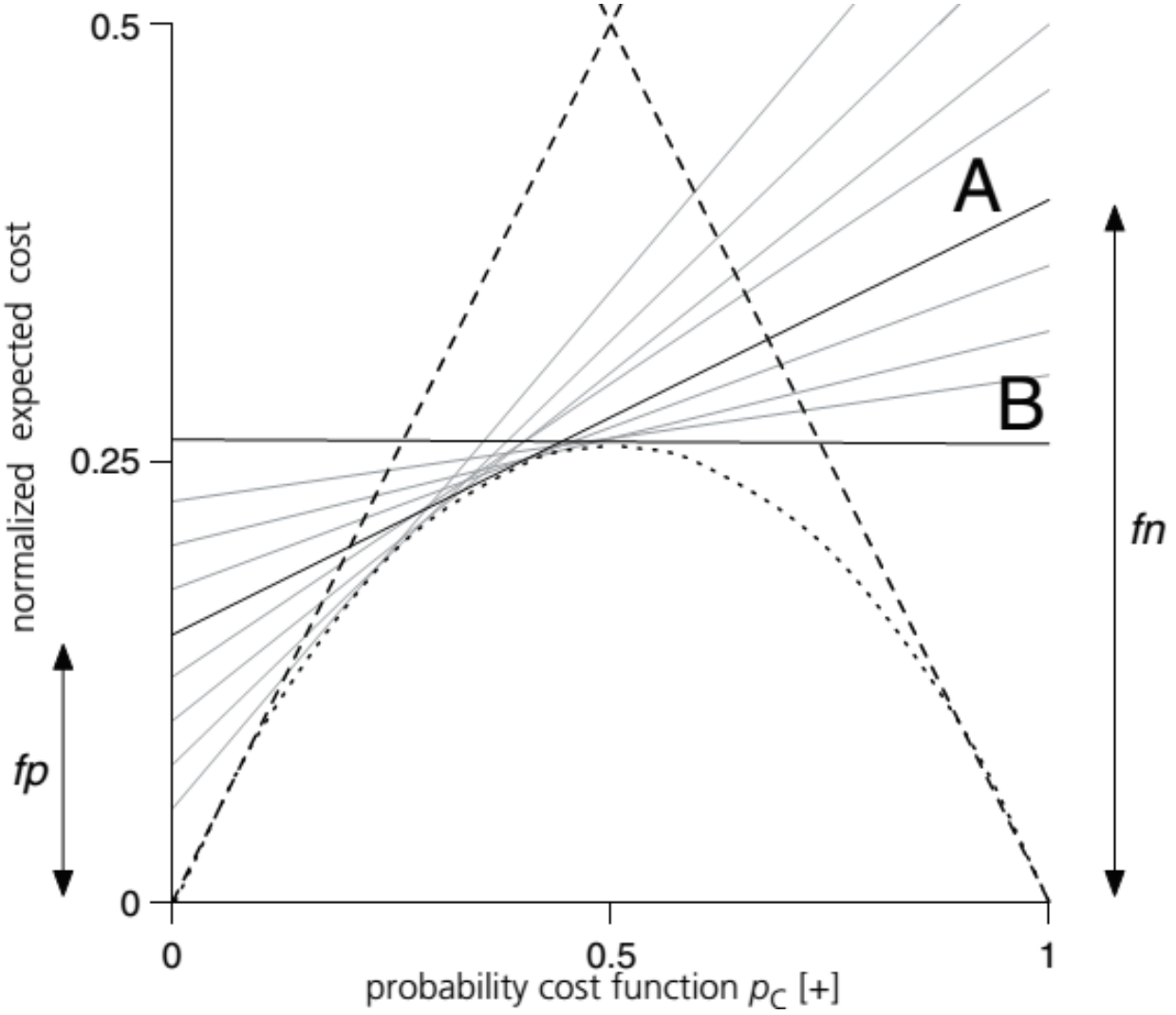
Súlyozásos költség görbék

- Mi is a normalizált várható költség értéke:

$$\mathrm{nec}(p_c[+]) = \frac{fn \cdot p[+] \cdot C[-|+] + fp \cdot p[-] \cdot C[+|-]}{p[+] \cdot C[-|+] + p[-] \cdot C[+|-]}$$

- Piros: FN besorolásból eredő várható hiba
- Zöld: FP besorolásból eredő várható hiba
- Kék: Elérhető legnagyobb várható hiba (minden elemet rosszul osztályzunk)

Súlyozásos költség görbék



Hiba kiértékelése predikciónál

Performance measure	Formula
mean-squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}$
root mean-squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{n}}$
mean absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{n}$
relative squared error	$\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}$, where $\bar{a}=\frac{1}{n}\sum_i a_i$
root relative squared error	$\sqrt{\frac{(p_1-a_1)^2+\ldots+(p_n-a_n)^2}{(a_1-\bar{a})^2+\ldots+(a_n-\bar{a})^2}}$
relative absolute error	$\frac{\lvert p_1-a_1\rvert+\ldots+\lvert p_n-a_n\rvert}{\lvert a_1-\bar{a}\rvert+\ldots+\lvert a_n-\bar{a}\rvert}$
correlation coefficient	$\frac{S_{PA}}{\sqrt{S_P S_A}}$, where $S_{PA}=\frac{\sum_i (p_i-\bar{p})(a_i-\bar{a})}{n-1}$, $S_p=\frac{\sum_i (p_i-\bar{p})^2}{n-1}$, and $S_A=\frac{\sum_i (a_i-\bar{a})^2}{n-1}$

Hiba kiértékelése predikciónál

- Alapvetően a kritériumfüggvények által definiált sorrendezés egymástól relevánsan eltérő osztályozókra megegyezik.

	A	B	C	D
root mean-squared error	67.8	91.7	63.3	57.4
mean absolute error	41.3	38.5	33.4	29.2
root relative squared error	42.2%	57.2%	39.4%	35.8%
relative absolute error	43.1%	40.1%	34.8%	30.4%
correlation coefficient	0.88	0.88	0.89	0.91

Minimum description length

- Gépi tanulás célja:
  - Leképzést lehető legkisebb hibával annak mintáiból megtanulni.
  - De egy véges ponthalmazra végtelen sok függvény illeszkedhet.
  - Zajjal terheltek a bemenetek, illetve a kimenetek
  - Occam elv: Azonos valószínűségű magyarázatokból az egyszerűbbet preferáljuk.
  - **Statisztikai tanuláselmélet** tárgyalja behatóbban

Minimum Decsiption Length

- MDL szerinti legjobb teória:
  - **Legkisebb** méretű a „mögöttes **modell**”
  - A tanításhoz felhasznált minták esetén átlagosan **legkevesebb plusz információ**ra van szükség.
- MDL elv analógiája:
  - Zajmentes csatornán kell a tanítómintákat átvinni.
  - Keressük azt a módszert, amivel a **lehető legkevesebb bit**et kell csak forgalmazni.

MDL

- Szélsőséges esetek:
  - Memorizáló modell
  - Null modell
- Ezen esetek leírásai nagyok
- Nincs szükség szeparált teszthalmazra:
  - Statisztikai tanuláselmélet szerint **R=Remp+Q**
  - A mintakészlet átviteléhez szükséges információ mennyiségét vizsgáljuk csak.
  - Ellenben nem igazán alkalmazható jól.

Statisztikai tanuláselmélet VC dim

- Egy szakértő VC dimenziója azon mintahalmaz **maximális mérete**, mely mintái:
  - **Tetszőleges**, de nem elfajuló **elhelyezkedésű**
  - **Tetszőlegesen** két **osztályba** sorolásúak
  - És **létezik olyan paraméterezése** a szakértőnek, amivel **helyesen** képes a halmazt **osztályoz**ni.
- Mekkora a VC dimenziója egy lineáris szakértőnek?

MDL elvi levezetése

- Bayes szabály:

$$\Pr[T|E] = \frac{\Pr[E|T]\Pr[T]}{\Pr[E]}$$

- Log-lieklihood elven maximalizáljuk:

$$-\log(\Pr[T|E]) = -\log(\Pr[E|T]) - \log(\Pr[T]) + \log(\Pr[E])$$

- Tehát minimalizáljuk az alábbi kifejezést:

$$-\log(\Pr[E|T]) - \log(\Pr[T])$$

MDL alkalmazásának nehézségei

- Pr[T] meghatározása lehetetlen:
  - Függ a modell kódjától
  - Mekkora komplexitása van egy műveletnek?
  - Mekkora a komplexitása egy művelet sorozatnak?
  - Gyakorlatban alacsony szintű (C/ Pascal) program kódjának hosszával közelítik.

MDL alkalmazásának nehézségei

- Pr[E|T] körüli problémák:
  - Szekvenciaként értelmezzük a tanító mintákat, vagy halmazként?
  - Ha szekvenciaként értelmezzük, akkor kihasználhatjuk a hibák autokorreláltságát?
- A megválaszolhatatlan kérdések miatt gyakorlatban inkább **Vapnik statisztikai tanuláselméletét** alkalmazzuk.

Klaszterezés MDL elv alapján

- Cél: **mintapontok** koordinátáit megengedett ε maximális pontatlansággal **átvinni a csatornán** minél kevesebb bitet forgalmazva.
- Mit küldjünk el a csatornán:
  - Klaszterek centroidját
  - Melyik minta melyik klaszterbe tartozik
  - Minták attribútumainak eltérése a klaszter középponttól ezen **eltérések eloszlása szerint tömörítve**.

Klaszterezés MDL elv alapján

- Klaszteren belüli eloszlást hogy kódoljuk?
  - Átküldjük pár, klaszteren belüli minta attribútumait.
  - Ezekre illeszti mindkét oldal a tömörítéshez felhasznált eloszlás paramétereit. (Ennek konkrét kivitelezése aktuális kutatások témája).
- Jelentős tömörítés elérhető, ha a módszer jól képes klaszterezni.
- Ha viszont a módszer elbukik adott mintán, akkor általában jobban megéri a „o modell”.

Adaptive Boosting

- Moduláris szakértőket hozunk létre:
  - Az egyes szakértőket különböző eloszlású, de azonos halmazból generált mintakészlettel tanítjuk
- Elméleti háttér:
  - Egy olyan osztályba sorolást kell megtanulni, mely szeparáló felülete **jól közelíthető egyszerűbb szeparáló felületek súlyozott átlagával**.
  - Összességében viszont egy komplex, nem lineáris leképzéssel állunk szemben.

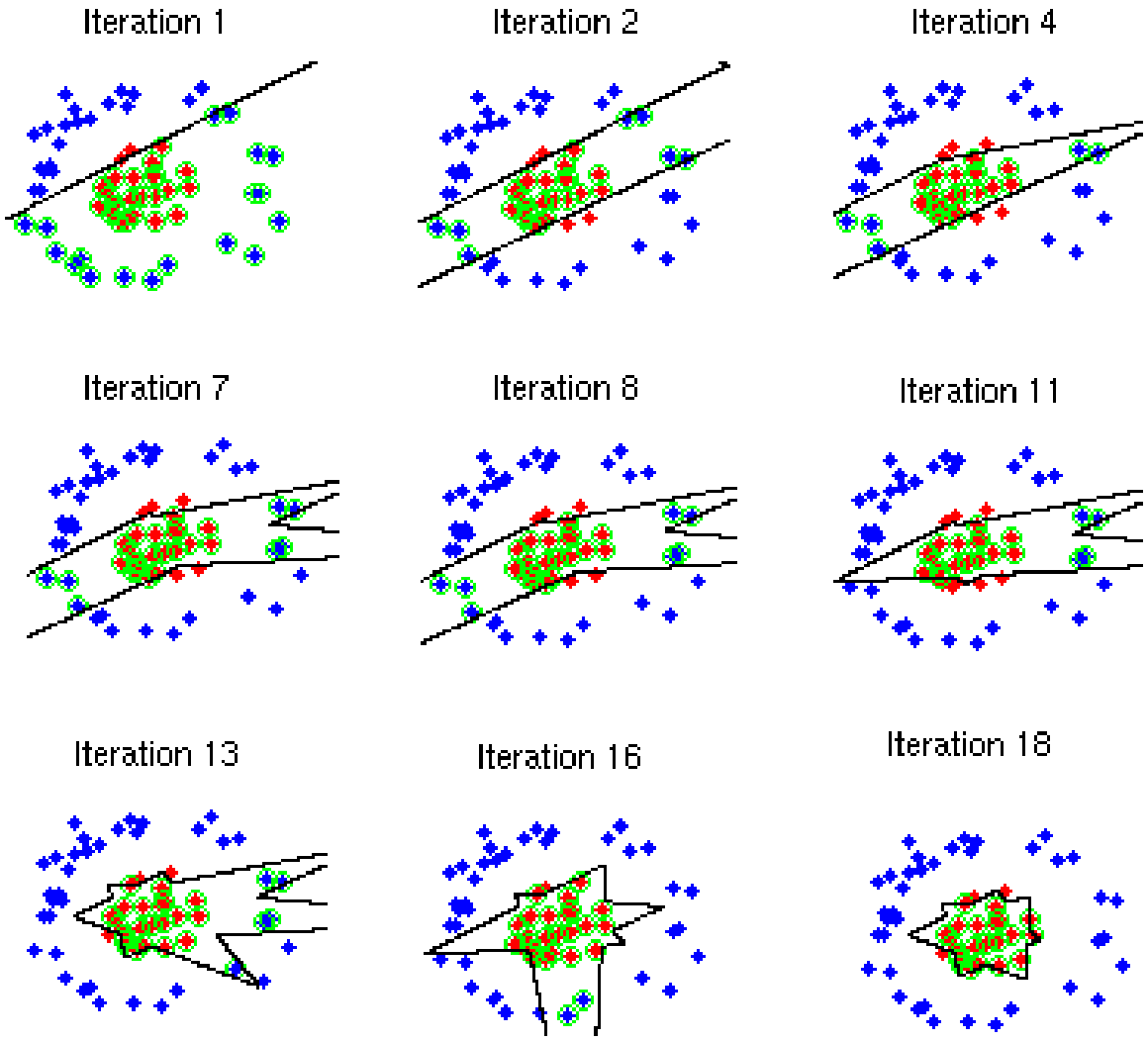
Adaptive Boosting

- Az általánosítási hiba várható értéke kisebb a hálónak, mint egy azonos határoló felületű szakértőnek:
  - Pl. k darab lineáris szakértő egyszerűbb egy erősen nem lineárisnál (pl. nem tudjuk megmondani előre, hogy a k szakértő kimenete egy 4-ed fokú polinommal közelíthető).

Adaptive Boosting

1. $D_l := 1/L \quad \forall l - re$
2. t:=1
3. SZ(t)-t alakítsuk ki D súlyokkal
4. $\varepsilon_t := \Pr\left[\mathrm{y_t}\left(x_l\right) \neq \mathrm{d}_l \right]$
5. $\alpha_t := \frac{1}{2} \ln\left(\frac{1-\varepsilon_t}{\varepsilon_t} \right)$
6. $D_l := D_l \cdot \exp((\left|\mathrm{y_t}\left(x_l\right) - \mathrm{d}_l\right| - 1) \cdot \alpha_t \quad \forall l - re$
7. $D_l := D_l \Big/ \sum_k D_k \quad \forall l - re$
8. t:=t+1

Adaptive Boosting



Mixture of experts

- Itt moduláris szakértőt alakítunk ki.
- Viszont itt a domént partícionáljuk különböző területekre:
  - Egyes részdoménekben elég egyszerűbb osztályozó
  - Összességében itt is nemlineáris szeparáló felülettel küzdünk.
  - Jobb az általánosító képessége az így kialakított szakértő együttesnek.

Diákhoz felhasznált irodalom

[1]: Ian H. Witten, Eibe Frank, Mark A. Hall: **Data Mining** Practical Machine Learning Tools and Techniques 3rd edition pp 143-185

[2]: Horváth G. (szerk): **Neurális hálózatok** 2006, 2. fejezet

[3]: Schapire, R. E., **A Brief Introduction to Boostin**g, (IJCAI-99) Vol. 2. pp. 1401-1406

[4]:Jacobs, R.A. , Jordan, M. I. , Nowlan, S. J. , Hinton, G. E. **Adaptive Mixture of Local Experts, Neural Computation**, Vol. 3. pp. 79-89. 1991.

Köszönöm a figyelmet!