Osztályozás

Fodor Gábor

2010. március 17.

Bevezetés

- Felügyelt tanulás (Supervised learning)
- Magyarázó attribútumok, magyarázandó attribútum
- Tanító pontok, teszthalmaz
- Regresszió és Osztályozás
- Előfeldolgozás (Hiányos adatok, adattisztítás, adattranszformáció, releváns adatok)
- Hiba mértékek (Accuracy, Precision, Recall, ROC, AUC, Cost)

Definíciók

Def. (Döntési szabály)

Az A attribútumhalmaz felett értelmezett döntési szabály alatt olyan $R : \phi(A) \to Y = y$ logikai implikációt értünk, amelyek feltételrészében az attribútumokra vonatkozó feltételek logikai kapcsolatai állnak, a következményrészben pedig az osztályattribútumra vonatkozó ítélet.

Def. (Illeszkedés)

Az $R : \phi(A) \to Y = y$ döntési szabályra illeszkedik a t objektum, ha a feltételrész attribútumváltozóiba t megfelelő értékeit helyettesítve igaz értéket kapunk.

Def. (Fedés)

Az $R : \phi(A) \to Y = y$ szabály lefedi az T objektumhalmazt, ha minden objektum illeszkedik a szabályra. Adott τ tanító halmaz esetén az R által fedett tanítópontok halmazát $\text{cover}_\tau(R)$-rel jelöljük.

Döntési szabályok

- szabályhalmaz és szabálysorozat
- egyértelműség
- teljesség
- kifejezőerő
- döntési táblázat

1R algoritmus

Pofonegyszerű osztályozó algoritmus, kiválaszt egy attribútumot, majd annyi szabályt állít elő, ahány különböző értéket vesz fel az attribútumunk a tanító adathalmazban.
Az $A = a \to Y = y_i$ szabály következményében szereplő y_i osztály értelemszerűen a leggyakoribb lesz az A attribútumában a-t felvevő tanítópontok közül.
Az 1R egyértelmű szabályhalmazt állít elő.
Valós attribútumok problémája
,,Egyszerűsége ellenére elég jól muzsikál a gyakorlatban."
0R osztályozó

A Prism módszer

Alapfeltétel: nincsenek olyan tanítópontok, melyek fontos magyarázó attribútumai megegyeznek, de osztályattribútumukban különböznek. (!)
separate and conquer

Csak 100%-os pontosságú szabályokat állít elő.

Algorithm 8 Prism

Require: $\mathcal{T}$: tanítópontok halmaza,
Y: osztályattribútum változó,

```
for all y ∈ osztályattribútum értékre do
  E ← az y osztályba tartozó tanítópontok
  φ ← ∅
  while E ≠ ∅ do
    R ← φ → Y = y
    while Er_T(R) ≠ 0 do
      hiba ← 1
      for all (A, a) attribútum-érték párra do
        if Er(φ AND A = a → Y = y) < hiba then
          hiba ← Er(φ AND A = a → Y = y)
          A* ← A
          a* ← a
        end if
      end for
      φ ← φ AND A* = a*
    end while
    T ← T \ cover(R)
  end while
end for
```

Általában

Könnyen értelmezhető, egyértelmű szabályhalmazok
Faépítés rekurzív vágásokkal(kérdésekkel)
Leállás:

- Attribútumhiány
- Mélységi korlát
- Nincs jó vágás

Főbb algoritmuscsaládok:

- Interactive Dichotomatizer 3 (ID3)
- Classification and Regression Trees (CART,C&RT)
- Chi-squared Automatic Interaction Detection (CHAID)

Egy kis információelmélet

X, Y diszkrét v.v. k, l lehetséges értékkel
Ekkor Y entrópiája:

$$H(Y) = -\sum_{i=1}^{l} \mathbb{P}(Y = i) \log \mathbb{P}(Y = i)$$

Tegyük fel X megfigyelt változó értéke x_j, ekkor Y-nal kapcsolatos bizonytalanságunk:

$$H(Y|X = x_j) = -\sum_{i=1}^{l} \mathbb{P}(Y = i|X = x_j) \log \mathbb{P}(Y = i|X = x_j)$$

X ismeretében a várható bizonytalanságunk:

$$H(Y|X) = \sum_{j=1}^{k} \mathbb{P}(X = x_j) H(Y|X = x_j)$$

Kölcsönös információ $I(Y, X) = H(Y) - H(Y|X)$

ID3

Az egyik legősibb és legismertebb osztályozó algoritmus
Y osztályozásakor azt az X attribútumot választja, melyre $I(X, Y)$ maximális
Hátrány: terebélyes fa
Javítási ötlet nyereségaránnyal

$$gainratio(X) = I(X, Y)/H(X)$$

Egy attribútum szerint legfeljebb egyszer vágunk.
Bináris fa Feltételek a csomópontokban: Sorrend, kategória, intervallum

Vágási függvények

X diszkrét v.v. k lehetséges értékkel, $p_i := \mathbb{P}(X = x_i)$, $\mathbf{p} = (p_1, p_2, \ldots, p_k)$
Egy $\Phi : [0,1]^k \to \mathbb{R}$ vágási függvényre vonatkozó **Taylor-Silverman** kritériumok:

1. $\Phi(\mathbf{p}) \geq 0$
2. Φ az elfajult eloszlásra minimális
3. Φ az egyenletes eloszlásra maximális
4. $\Phi(\mathbf{p})$ a $\mathbf{p}$ komponenseire nézve szimmetrikus.
5. Φ differenciálható

CART

Entrópia helyett Gini-index:

$$Gini(\mathbf{p}) = 1 - \sum_{i=1}^{k} p_i^2$$

Ferdén is tudnak vágni (lineáris kombináció)
Mindig bináris döntés

Egy kis statisztika

$A_1, \ldots, A_r$ teljes eseményrendszer

$$H_0 : \mathbb{P}(A_i) = p_i \qquad (i = 1, \ldots, r),$$

n független megfigyelés során jelölje ν_i a megfelelő A_i gyakoriságát!
Ekkor H_0 fennállásakor $(\nu_1, \ldots, \nu_r)$ polinomiális eloszlású.
$n_1 + \cdots + n_r = n$ esetén:

$$\mathbb{P}_{H_0}(\nu_1 = n_1, \ldots, \nu_r = n_r) = \frac{n!}{n_1! \ldots n_r!} p_1^{n_1} \ldots p_r^{n_r}$$

T.

Ha $(\nu_1, \nu_2, \ldots, \nu_r)$ polinomiális eloszlású n és $p_1, \ldots, p_r (p_i > 0)$ paraméterekkel akkor $n \to \infty$ esetén

$$\sum_{i=1}^{r} \frac{(\nu_i - np_i)^2}{np_i} \to \chi^2(r-1)$$

CHAID

Három lépés

- Minden magyarázó változóra a statisztikailag leginkább független kategóriák páronkénti *egyesítése*
- A leginkább függő attribútum kategóriái szerinti *felosztás*
- A rekurzió folytatása valamely *megállítási* kritériumig

Függetlenségvizsgálat χ^2 próbával diszkrét esetben

X, Y diszkrét, $A_i = \{X = x_i\}$, $B_j = \{Y = y_j\}$, $p_i = \mathbb{P}(A_i)$, $q_j = \mathbb{P}(B_j)$
$\nu_{ij} = |\{k : X_k = x_i, Y_k = y_j\}|$
$H_0 : X$ és Y függetlenek: $\mathbb{P}(A_i \cap B_j) = \mathbb{P}(A_i)\mathbb{P}(B_j) = p_i q_j$

$$\chi^2 = \sum_i \sum_j \frac{(\nu_{ij} - n p_i q_j)^2}{n p_i q_j}$$

Bayes-hálók bevezető

G (DAG a diszkrét attribútumokon mint csúcsokon) a változók közötti függőségi viszonyokat kódolja.
Lokális Markov-feltétel: Bármely attribútum független nem leszármazottaitól, ha ismert szüleinek értéke.

T. (Láncszabály Bayes-hálókra)

$$\mathbb{P}(\mathbf{X}) = \prod_{i=1}^{n} \mathbb{P}(X_i | Par_i)$$

Következtetés a hálóban

Def. (Markov-takaró)

Egy változó Markov-takarója a szüleinek, gyermekeinek és a gyermekei szüleinek halmaza.

Feltételes valószínűségi tábla (CPT)

A tanulás nehézségei

Paramétertanulás, struktúratanulás
Melyek a jó struktúrák?
Kritériumfüggvények:

$$BIC^1(B, D) = \sum_{i=1}^{N} log(\mathbb{P}(d_i|B)) - \frac{logN}{2}|\Theta|$$

$$AIC^2(B, D) = \sum_{i=1}^{N} log(\mathbb{P}(d_i|B)) - |\Theta|$$

Az óriási keresési tér szűkítése

- Topológikus sorrend felállítása
- Szülőhalmazok méretének korlátozása

[1] Bayesian Information Criterion
[2] Akaike Information Criterion

Mohó keresések

Tetszőleges kiindulási gráf (üres, szakértői, random)

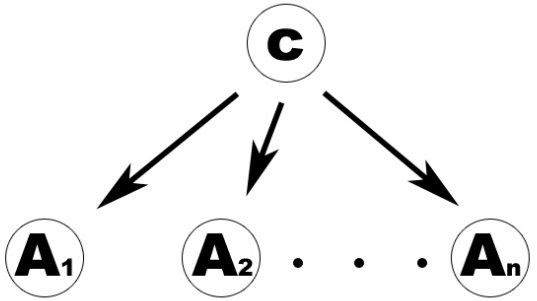
- éltörlés
- élhozzáadás
- élfordítás

WEKA algoritmusok

- K2
- HillClimbing
- RepeatedHillClimbing
- Simulated Annealing

Naive Bayes Classifier (NB)

Durva függetlenségi feltétel, rögzített struktúra
C osztályattribútum
$A_1, A_2, \ldots, A_n$ magyarázó változók



Bayes-tétel miatt

$$\mathbb{P}(C|A_1, A_2, \ldots, A_n) = \frac{\mathbb{P}(C)\mathbb{P}(A_1, A_2, \ldots, A_n|C)}{\mathbb{P}(A_1, A_2, \ldots, A_n)}$$

Függetlenségi feltételünk alapján

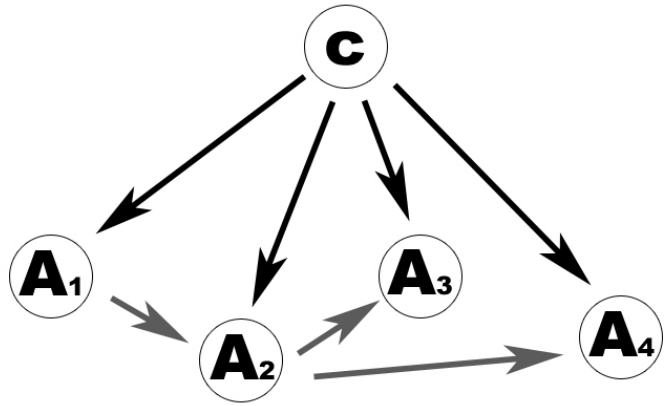
$$\mathbb{P}(A_1, \ldots, A_n|C) = \prod_{i=1}^{n} \mathbb{P}(A_i|C)$$

ML döntés

$$classify(a_1, \ldots, a_n) = \arg\max_c \left(\mathbb{P}(C = c) \prod_{i=1}^{n} \mathbb{P}(A_i = a_i|C = c\right)$$

Tree Augmented Naive Bayes Model(TAN)

Bonyolultabb, de kezelhető struktúra
C árva
$A_1, A_2, \ldots, A_n$ mind C gyermekei
$A_1, A_2, \ldots, A_n$ pontokon irányított fa



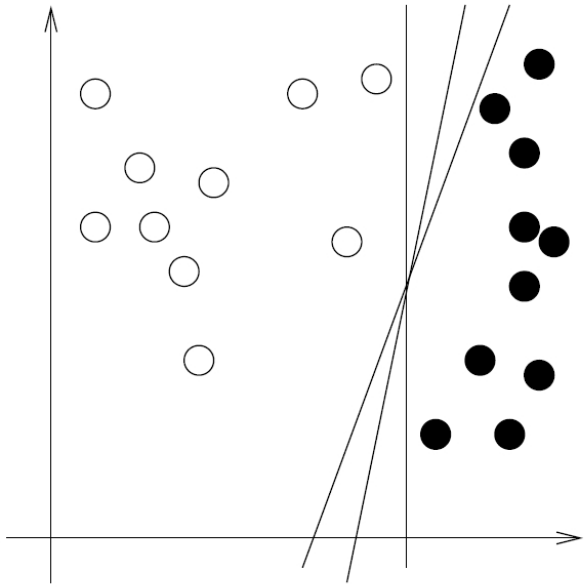
A tanulás működik polinomidőben!

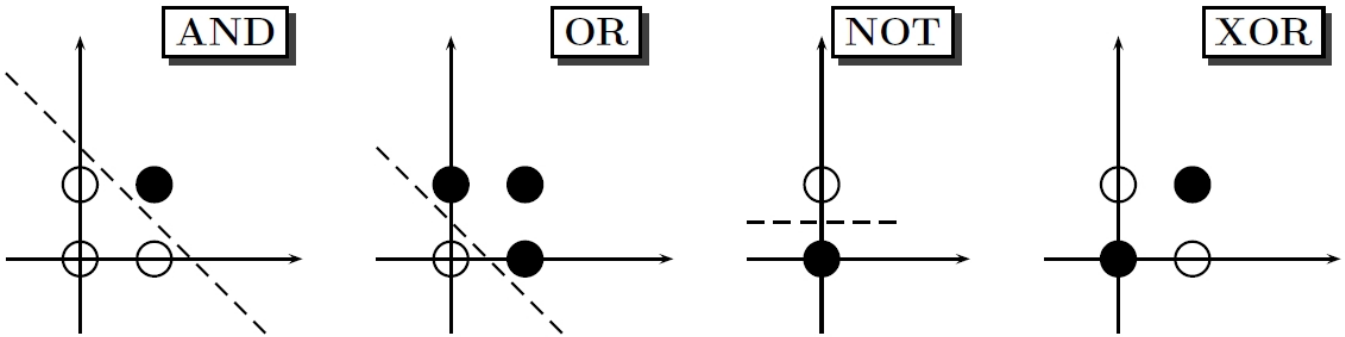
1. Meghatározzuk az adatok segítségével $\hat{I}(A_i, A_j|C)$-t minden (i, j) párra, ezekkel súlyozzuk egy n-pontú teljes gráf éleit.
2. Ebben a gráfban keresünk egy maximális feszítőfát, erre ismertek $O(n^2 \log n)$ idejű algoritmusok.
3. Kiválasztunk egy gyökeret és ennek megfelelően irányítjuk a feszítőfa éleit.
4. Végül hozzáadjuk a gráfhoz a C csúcsot és behúzzuk a maradék éleket.

Lineáris szeparálás

Két osztály lineárisan szeparálható, ha egy hipersík segítségével el tudjuk különíteni a két osztály pontjait.

$$w_1 a_1 + w_2 a_2 + \cdots + w_n a_n = 0$$





Perceptron

- A neurális hálók ősének tekinthető
- Minden attribútum valós
- Ha a lineáris kombináció pozitív első osztály
- Feladatunk megfelelő (nem optimális!) w súlyok keresése

Algorithm 7 Perceptron tanulási szabály

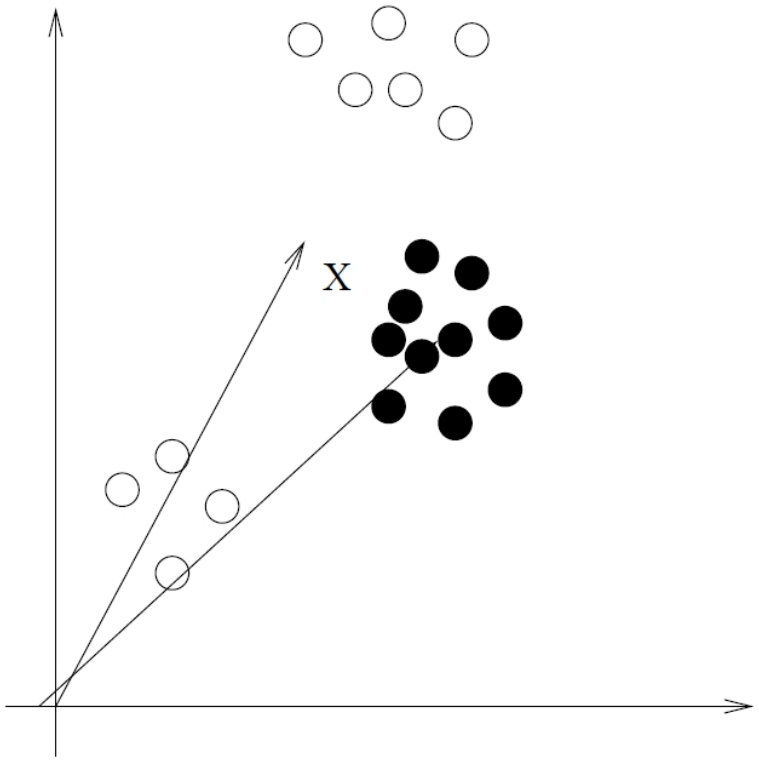
Require: $\mathcal{T}$: tanítópontok halmaza

$\vec{w} = (0,0,\ldots,0)$
while van rosszul osztályozott $t \in \mathcal{T}$ **do**
  for all minden $\vec{t} \in \mathcal{T}$ **do**
    if $\vec{t}$ rosszul van osztályozva **then**
      if $\vec{t}$ az első osztályba tartozik **then**
        $\vec{w} = \vec{w} + \vec{t}$
      else
        $\vec{w} = \vec{w} - \vec{t}$
      end if
    end if
  end for
end while

Winnow módszer csupa bináris attribútumra

Rocchio

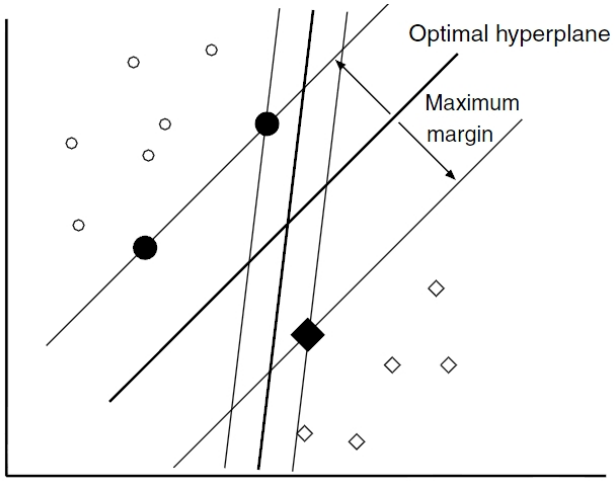
- Klasszikus IR algoritmus
- Minden attribútum valós
- Minden osztályhoz prototípusvektor (D_c mintaátlag)
- Kicsiny számításigény, gyors tanulás (online környezetben is)



$$c = \beta Avg_{d_j \in C} d_j - \gamma Avg_{d_j \notin C} d_j$$

Hard-Margin SVM

Bináris osztályozás $\{-1, +1\}$
Tfh. lineárisan szeparálhatók az osztályok!
A szeparáló sík egyenlete: $D(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b = 0$
Kis átalakításokkal:



$$y_k(\mathbf{w}^T\mathbf{x}_k + b) > 1$$

$\mathbf{x}$ pont távolsága $D(\mathbf{x})$-től: $|D(\mathbf{x})|/||\mathbf{w}||$

$$\frac{y_k(D(\mathbf{x}_k))}{||\mathbf{w}||} \geq \delta$$

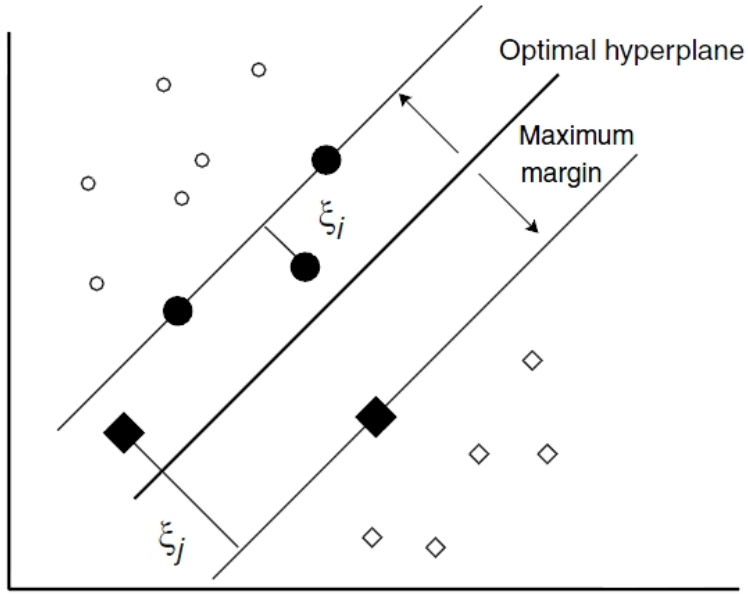
Célunk $\frac{1}{2}||\mathbf{w}||^2$-t minimalizálni, $y_k(\mathbf{w}^T\mathbf{x}_k + b) \geq 1$ korlátok mellett.
(Kvadratikus optimalizálási feladat, KKT, Lagrange multiplikátorok)

Soft-Margin SVM

A feltételek enyhítése ξ_i nemnegatív segédváltozókkal:

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i$$

A segédváltozók miatt mindig létezik megengedett megoldás.



$$\frac{1}{2}||\mathbf{w}||^2 + C\sum_i \xi_i^p \rightarrow min$$

$$y_k(\mathbf{w}^T\mathbf{x}_k + b) \geq 1 \qquad i = 1, 2, \ldots$$

Nemlinearitás kezelése magfüggvényekkel

- Nemlineáris transzformáció (magasabb dimenzióba)
- A transzformált térben az optimális szeparáló sík meghatározása

$$D(\mathbf{x}) = \mathbf{w}^T \mathbf{g}(\mathbf{x}) + b$$

$$H(\mathbf{x}, \mathbf{x}') = \mathbf{g}^T(\mathbf{x})\mathbf{g}(\mathbf{x})$$

Lineáris magfüggvények	$H(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{x}'$
Polinomiális magfüggvények	$H(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + 1)$
RBF magfüggvények	$H(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\|\mathbf{x} - \mathbf{x}'\|\|)$

SVM vs. NN

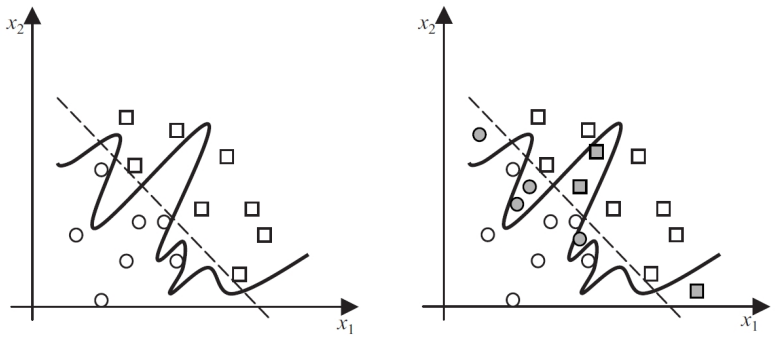
Előnyök

1. Maximált általánosítóképesség
2. Nincs lokális optimum
3. Hatékonyság kiugró (outlier) értékek esetén is

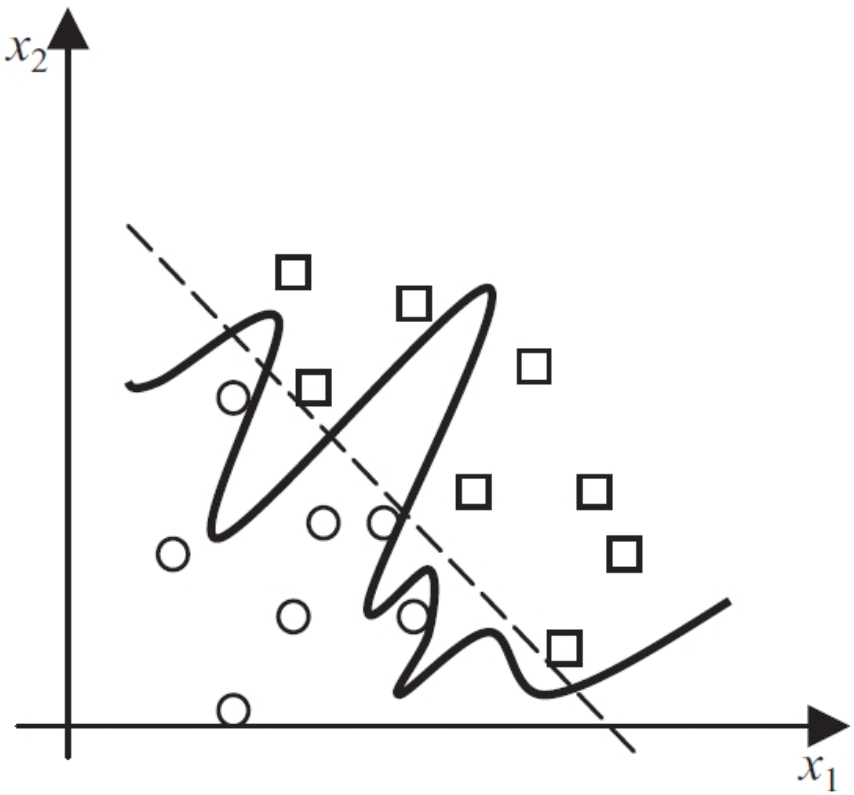
Hátrányok

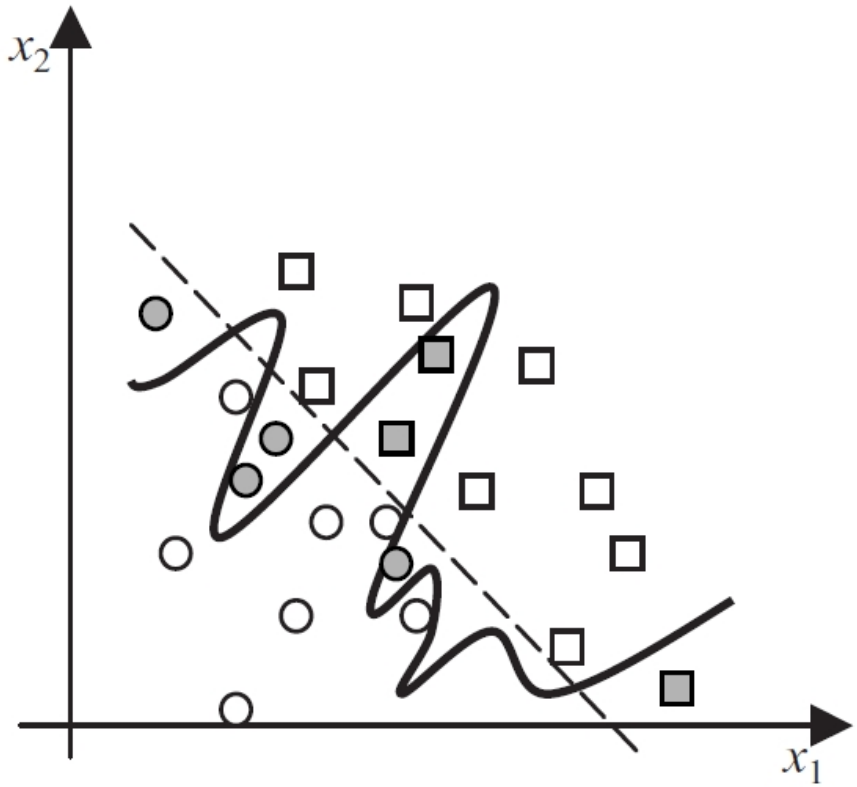
1. Bináris döntés
2. Lassú tanulás
3. Paraméterek kezelése

Mindkét módszer univerzális függvényapproximátor



overfitting





RandomForest

M magyarázó változó, N adatsor,
Minden egyes döntési fának választunk (visszatevéses mintavételezéssel-bootstrap) egy N méretű mintát.
Minden csomópontban random $m(<< M)$ attribútum közül kiválasztjuk azt, amelyik szerint vágunk.
Végül az erdőt összeszavaztatjuk többségi szavazással.

Előnyök

- Sok attribútummal is elbír
- Pontos osztályozás
- Gyors tanulás
- *Túltanulás elkerülése*

Hátrányok

- Független attribútumok
- *Torz mintavételezés*

Bagging, Stacking

Bootstrap **agg**regat**ing**
Szintén Leo Breiman 1994-ből, nemcsak döntési fát, tetszőleges tanuló algoritmust alkalmazhatunk.
Túltanulás elkerülése
Stabil modelleken nem segít.

Stacking
n belső modell kimenetét adjuk egy összeszavaztató modellnek

Boosting

AdaBoost Freund és Schapire 1995

Cél: egyszerű modellek adaptív alkalmazásával pontos eredmény

T körben tanítunk egy-egy h_t modellt, a $D_t(i)$ eloszlással mintavételezett tanítóhalmazon.

A modell hibája:

$$\epsilon_t = \sum_{i:h_t(x_i)\neq y_i} D_t(i)$$

$$\alpha_t := \frac{1}{2}\ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$$

Frissítés:

$$D_i(t) = \frac{D_i(t)\exp(-\alpha_t h_t(x_i) y_i)}{Z_t}$$

Végső döntésünk:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$$

Adatbányászati eszközök

C5.0

C&R Tree

CHAID

Bayes Ne:

SVM

OneR
DecisionTree
CHAID
ID3
EvoSVM
LibSVMLearner
PsoSVM
RandomForest
AdaBoost
Bagging
Stacking

Prism
OneR
ZeroR
BayesNet
ID3
Winnow
VotedPerceptron
MultilayerPerceptron
RandomForest
AdaBoostM1
Bagging
LogitBoost
Stacking

Források


Bodon Ferenc,
Adatbányászati algoritmusok,
(2010)


Nir Friedman, Dan Geiger, Moises Goldszmidt,
Bayesian Network Classifiers,
(1997)


R. R. Bouckaert, E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, D. Scuse,
WEKA Manual for Version 3-7-1,
(2010)


Shiego Abe,
Support Vector Machines for Pattern Classification,
(2005)