Odločitvena drevesa in pravila

Ljupčo Todorovski

Univerza v Ljubljani, Fakulteta za upravo
Institut Jožef Stefan, Odsek za tehnologije znanja (E-8)

Marec 2020

Pregled predavanja

Odločitvena drevesa

- Algoritem za učenje dreves *TDIDT*
- Sprotno in naknadno rezanje dreves
- Kontrola pristranskosti in variance

Odločitvena pravila

- Algoritem za učenje posameznih pravil
- Prekrivni algoritem za učenje množice pravil
- Kontrola pristranskosti in variance

Zgradba odločitvenih dreves

Notranje vozlišče

Testira vrednost izbrane napovedne spremenljivke X, npr. $X_1 < 0.5171$

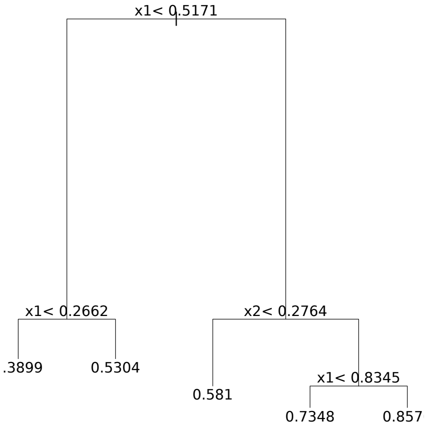
Veje

Ustrezajo izidom testa, npr. *True* in *False*

Končno vozlišče

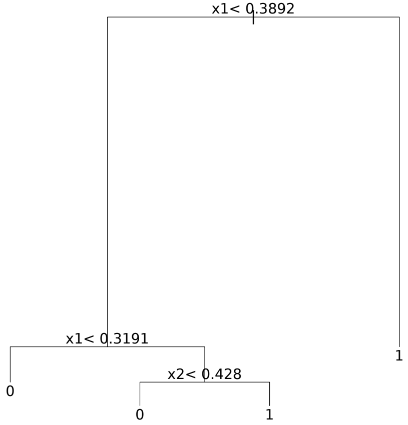
Poda napoved ciljne spremenljivke Y, npr. $\hat{Y} = 0.39$

Primer regresijskega drevesa



$$Y = (1 + X_1 + X_1 X_2)/3 + \mathcal{N}(0, 0.05),\ D_i = [0, 1], i = 1..2, D_Y = [0, 1]$$

Primer klasifikacijskega drevesa

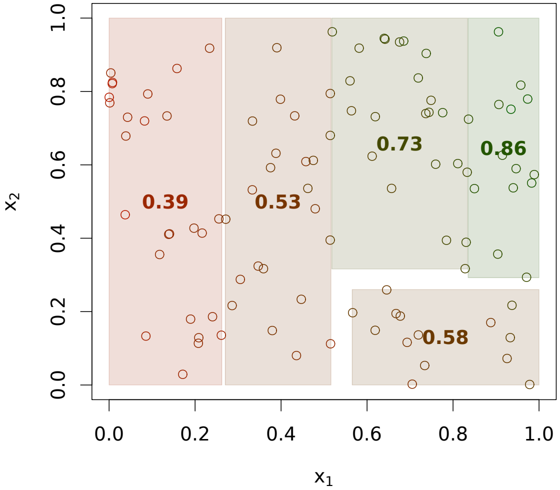


$Y = I((1 + X_1 + X_1X_2)/3 \geq 0.5), D_i = [0,1], i = 1..2, D_Y = \{0,1\}$
zamenjamo vrednost Y 0.05 naključno izbranim primerom

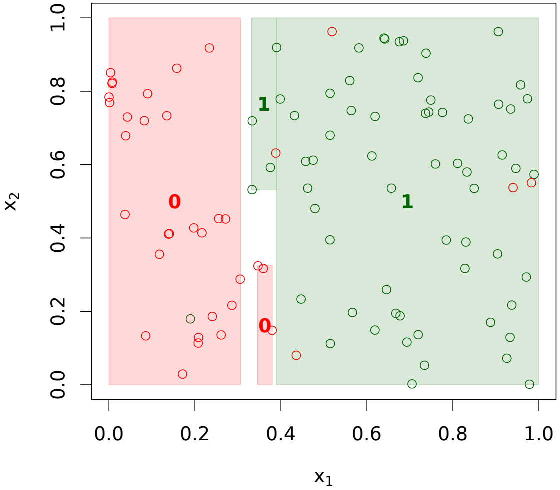
Napoved odločitvenega drevesa za podan primer

- Začni v korenskem vozlišču
- Ponavljaj dokler si v notranjem vozlišču
    - Opravi test notranjega vozlišča
    - Sledi veji, ki ustreza izidu opravljenega testa do novega vozlišča
- Uporabi končno vozlišče za napoved

Napovedi regresijskega drevesa



Napovedi klasifikacijskega drevesa



Algoritem $TDIDT(S) = DecisionTree$

TDIDT: Top-Down Induction of Decision Trees

function $TDIDT(S)$
  if $Impurity(S) = 0$
    return $DecisionTree(leaf : S)$
  $Split = SelectOptimal(S)$
  $\{S_1, S_2, \ldots S_s\} = Partition(S, Split)$
  for $i = 1$ **to** s **do**
    $t_i = TDIDT(S_i)$
  return $DecisionTree(node : Split, descendants : (t_1, t_2, \ldots t_s))$

Pomembne komponente algoritma *TDIDT*

- $Partition(S, Split)$: test *Split* in razbitje množice S
- $Impurity(S)$: nečistoča množice S
- $SelectOptimal(S)$: izbira optimalnega testa za množico S

Testi in razbitje množice *Partition*$(S, Split)$

Diskretna spremenljivka $X_j, D_j = \{v_1, v_2, \ldots v_m\}$

1. $Split\ X_j$: $S_i = \{(\boldsymbol{x}, y) \in S : x_j = v_i\}$, en možen test
2. $Split\ X_j \in V$, $V \subset D_j$, $V \neq \emptyset$: $S_1 = \{(\boldsymbol{x}, y) \in S : x_j \in V\}$, $S_2 = S \setminus S_1$, $(2^{|D_j(S)|} - 2)/2 = 2^{|D_j(S)|-1} - 1$ možnih testov

Numerična spremenljivka $X_j, D_j \subseteq \mathbb{R}$

$Split\ X_j < v$: $S_1 = \{(\boldsymbol{x}, y) \in S : x_j < v\}$, $S_2 = S \setminus S_1$, $|D_j(S)| - 1$ možnih testov (vrednosti praga v)

$D_j(S) \subseteq D_j$ je množica vrednosti X_j v množici S.

Napoved končnega vozlišča *leaf* : *S*

Regresija, $D_Y \subseteq \mathbb{R}$: povprečje

$$Prediction(leaf : S) = \frac{1}{|S|} \sum_{(\boldsymbol{x},y)\in S} y$$

Klasifikacija, $D_Y = \{v_1, v_2, \ldots v_c\}$: večinski razred

$$Prediction(leaf : S) = \arg\max_{v_i \in D_Y} p(Y = v_i|S)$$

$p(Y = v_i|S)$ je verjetnost, da primer iz množice S pripada razredu v_i.

Funkcija nečistoče *Impurity*(S)

Funkcija nečistoče meri varianco vrednosti ciljne spremenljivke Y v S.

Regresija, $D_Y \subseteq \mathbb{R}$

$$Impurity(S) = \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} (y - \bar{y})^2, \quad \bar{y} = \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S} y$$

Klasifikacija, $D_Y = \{v_1, v_2, \ldots v_c\}$

$$Impurity(S) = \phi(p_1, p_2, \ldots p_c)$$

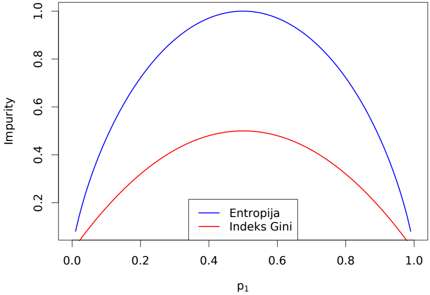
Verjetnosti $p_i = p(Y = v_i | S)$

Zaželene lastnosti funkcije nečistoče $\phi(p_1, p_2, \ldots p_c)$

1. Doseže maksimalno vrednost pri enakomerni porazdelitvi $\forall i : p_i = 1/c$
2. Doseže minimalno vrednost pri porazdelitvah, kjer $\exists ! i : p_i = 1$
3. Simetrična: neobčutljiva na vrstni red parametrov
4. Konkavna, zvezna in zvezno odvedljiva

Dve pogosto uporabljani funkciji

1. Entropija $\phi = -\sum_{i=1}^{c} p_i \log_2 p_i$, dokaz prvih dveh lastnosti na tabli
2. Indeks Gini $\phi = 1 - \sum_{i=1}^{c} p_i^2$, dokaz prvih dveh lastnosti doma



Izbira optimalnega testa $Split = SelectOptimal(S)$

Ciljna funkcija za optimizacijo zmanjšanje nečistoče *IR*

$$IR(Split, S) = Impurity(S) - \sum_{i=1}^{s} \frac{|S_i|}{|S|} Impurity(S_i)$$

- Test *Split* povzroči razbitje S na $\{S_1, S_2, \dots S_s\}$
- IR = *Impurity Reduction*

Test izberemo z optimizacijo

$$\max_{Split} IR(Split, S) \equiv \min_{Split} \sum_{i=1}^{s} |S_i| Impurity(S_i)$$

Optimizacijski algoritmi za izbiro testa

1. Naštejemo vse možne teste
2. Vzorčimo množico vseh možnih testov

Ali lahko prepoznam delfina?

primer	dolzina	skrge	kljun	zob	delfin
x_1	3	ne	da	veliko	da
x_2	4	ne	da	veliko	da
x_3	3	ne	da	malo	da
x_4	5	ne	da	veliko	da
x_5	5	ne	da	malo	da
x_6	5	da	da	veliko	ne
x_7	4	da	da	veliko	ne
x_8	5	da	ne	veliko	ne
x_9	4	da	ne	veliko	ne
x_{10}	4	ne	da	malo	ne

Opis naloge strojnega učenja

Primeri

10 opazovanih živali, od tega 5 delfinov

Spremenljivke

- Ciljna spremenljivka $Y = \mathit{deflin}$, $D_Y = \{\mathit{da}, \mathit{ne}\}$
- Ena numerična napovedna spremenljivka $X_1 = \mathit{dolzina}$, $D_1 = \mathbb{R}^+$
- Tri diskretne napovedne spremenljivke
  - $X_2 = \mathit{skrge}$, $D_2 = \{\mathit{da}, \mathit{ne}\}$
  - $X_3 = \mathit{kljun}$, $D_2 = \{\mathit{da}, \mathit{ne}\}$
  - $X_4 = \mathit{zob}$, $D_2 = \{\mathit{veliko}, \mathit{malo}\}$

Napovedni model, ki razpoznava delfine: klasifikacijsko drevo

Nečistost prvotne množice in možni testi

$Impurity(S), S = \{x_1, x_2, \ldots x_{10}\}$

- $p(delfin = da) = p(delfin = ne) = 5/10 = 0.5$
- $Gini(S) = 1 - (0.5^2 + 0.5^2) = 0.5$

Pet možnih testov

1. $dolzina < 4$
2. $dolzina < 5$
3. *skrge*
4. *kljun*
5. *zob*

Test $dolzina < 4$

$Impurity(S_1), S_1 = \{x_1, x_3\}$

- $p(delfin = da) = 2/2 = 1,\ p(delfin = ne) = 0/2 = 0$
- $Gini(S_1) = 1 - (1^2 + 0^2) = 0$

$Impurity(S_2), S_2 = \{x_2, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$

- $p(delfin = da) = 3/8 = 0.375,\ p(delfin = ne) = 5/8 = 0.625$
- $Gini(S_2) = 1 - (0.375^2 + 0.625^2) = 0.46875$

$$IR(dolzina < 4) = 0.5 - (\frac{2}{10}0 + \frac{8}{10}0.46875) = 0.125$$

Test $dolzina < 5$

$Impurity(S_1), S_1 = \{x_1, x_2, x_3, x_7, x_9, x_{10}\}$

- $p(delfin = da) = 3/6 = 0.5,\ p(delfin = ne) = 3/6 = 0.5$
- $Gini(S_1) = 1 - (0.5^2 + 0.5^2) = 0.5$

$Impurity(S_2), S_2 = \{x_4, x_5, x_6, x_8\}$

- $p(delfin = da) = 2/4 = 0.5,\ p(delfin = ne) = 2/4 = 0.5$
- $Gini(S_2) = 1 - (0.5^2 + 0.5^2) = 0.5$

$$IR(dolzina < 5) = 0.5 - (\frac{6}{10}0.5 + \frac{4}{10}0.5) = 0$$

Test *skrge*

$Impurity(S_1), S_1 = \{x_6, x_7, x_8, x_9\}$

- $p(delfin = da) = 0/4 = 0,\ p(delfin = ne) = 4/4 = 1$
- $Gini(S_1) = 1 - (0^2 + 1^2) = 0$

$Impurity(S_2), S_2 = \{x_1, x_2, x_3, x_4, x_5, x_{10}\}$

- $p(delfin = da) = 5/6 = 0.833,\ p(delfin = ne) = 1/6 = 0.167$
- $Gini(S_2) = 1 - (0.833^2 + 0.167^2) = 0.278$

$$IR(skrge) = 0.5 - (\frac{4}{10}0 + \frac{6}{10}0.278) = 0.333$$

Test *kljun*

$Impurity(S_1), S_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_{10}\}$

- $p(delfin = da) = 5/8 = 0.625$, $p(delfin = ne) = 3/8 = 0.375$
- $Gini(S_1) = 1 - (0.625^2 + 0.375^2) = 0.46875$

$Impurity(S_2), S_2 = \{x_8, x_9\}$

- $p(delfin = da) = 0/2 = 0$, $p(delfin = ne) = 2/2 = 1$
- $Gini(S_2) = 1 - (0^2 + 1^2) = 0$

$$IR(kljun) = 0.5 - (\frac{8}{10}0.46875 + \frac{2}{10}0) = 0.125$$

Test *zob*

$Impurity(S_1), S_1 = \{x_1, x_2, x_4, x_6, x_7, x_8, x_9\}$

- $p(delfin = da) = 3/7 = 0.4286$, $p(delfin = ne) = 4/7 = 0.5714$
- $Gini(S_1) = 1 - (0.4286^2 + 0.5714^2) = 0.4898$

$Impurity(S_2), S_2 = \{x_3, x_5, x_{10}\}$

- $p(delfin = da) = 2/3 = 0.667$, $p(delfin = ne) = 1/3 = 0.333$
- $Gini(S_2) = 1 - (0.667^2 + 0.333^2) = 0.444$

$$IR(zob) = 0.5 - (\frac{7}{10}0.4898 + \frac{3}{10}0.444) = 0.0238$$

Izbira optimalnega testa: *skrge*

1. *dolzina* < 4: 0.125
2. *dolzina* < 5: 0
3. *skrge*: **0.333**
4. *kljun*: 0.125
5. *zob*: 0.0238

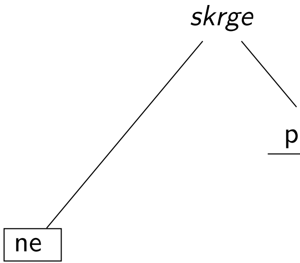
Delni rezultat

skrge

primer	...	delfin
x_6	...	ne
x_7	...	ne
x_8	...	ne
x_9	...	ne

primer	...	delfin
x_1	...	da
x_2	...	da
x_3	...	da
x_4	...	da
x_5	...	da
x_{10}	...	ne

Končno vozlišče



primer	...	delfin
x_1	...	da
x_2	...	da
x_3	...	da
x_4	...	da
x_5	...	da
x_{10}	...	ne

Rekurzija: nečistost množice in možni testi

$Impurity(S), S = \{x_1, x_2, x_3, x_4, x_5, x_{10}\}$

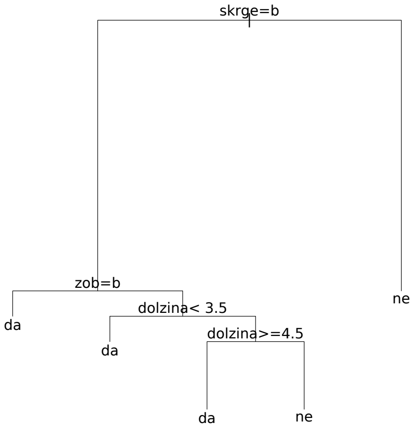
- $p(delfin = da) = 5/6 = 0.833,\ p(delfin = ne) = 1/6 = 0.167$
- $Gini(S) = 1 - (0.833^2 + 0.167^2) = 0.278$

Štirje možni testi

1. $dolzina < 4$
2. $dolzina < 5$
3. *kljun*
4. *zob*

Končni rezultat

Legenda: $skrge = b$ pomeni $skrge = ne$, $zob = b$ pomeni $zob = veliko$



Kontrola pristranskosti in variance

Velikost drevesa

- Kontroliramo tako, da dopuščamo *Impurity* > 0 v končnih vozliščih
- Rezanje dreves

Sprotno in naknadno rezanje dreves

- Sprotno: navzdol omejimo število primerov v (končnih) vozliščih (parameter minsplit, *ms*)
- Naknadno: pretvarjanje notranjih vozlišč v končna v rezultatu algoritma TDIDT (parameter stopnja rezanja, α ali *cp*)

Naknadno rezanje dreves

Rezanje poddrevesa v vozlišču t

Notranje vozlišče t spremenimo v končno vozlišče t_L.

Napaka poddrevesa t

$$Err_\alpha(t) = Err(t) + \alpha|t|$$

- $Err(t)$: napaka poddrevesa v t na učnih primerih iz t
- $|t|$: število končnih vozlišč, ki so nasledniki t

Odločitev o rezanju: $Err_\alpha(t_L) \leq Err_\alpha(t)$

Parameter α: regulator moči, stopnja rezanja (R implementacija: *cp*)

Primer rezanja poddrevesa $Err(t : dolzina < 3.5)$

Napaka poddrevesa

$Err_{\alpha}(t) = 0 + 3\,\alpha$, poddrevo ima napako 0 in tri končna vozlišča.

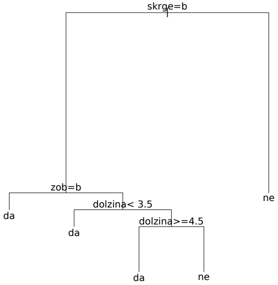
Napaka končnega vozlišča

$Err_{\alpha}(t_L) = 1/3 + 1\,\alpha$, učni primeri $\{x_3, x_5, x_{10}\}$: dva pozitivna in en negativen (napaka 1/3).

Prelomna točka $Err_{\alpha}(t) = Err_{\alpha}(t_L)$

Če je $\alpha \geq 1/6$, potem bo poddrevo porezano; sicer pa ne.

Legenda: $skrge = b$ pomeni $skrge = ne$, $zob = b$ pomeni $zob = veliko$



Primer rezanja poddrevesa $Err(t : zob \in \{veliko\})$

Napaka poddrevesa

$Err_\alpha(t) = 0 + 4\,\alpha$, poddrevo ima napako 0 in štiri končna vozlišča.

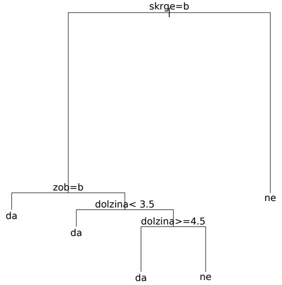
Napaka končnega vozlišča

$Err_\alpha(t_L) = 1/6 + 1\,\alpha$, šest učnih primerov: pet pozitivnih in en negativen (napaka 1/6).

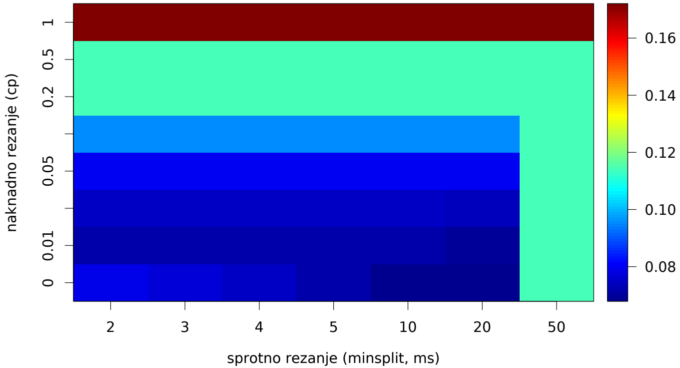
Prelomna točka $Err_\alpha(t) = Err_\alpha(t_L)$

Če je $\alpha \geq 1/18$, potem bo poddrevo porezano; sicer pa ne.

Legenda: $skrge = b$ pomeni $skrge = ne$, $zob = b$ pomeni $zob = veliko$



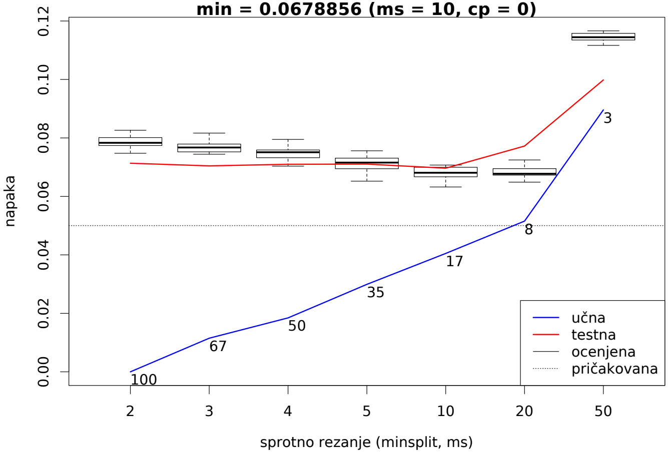
Regresijska drevesa: *ms* in *cp*



$$Y = (1 + X_1 + X_1 X_2)/3 + \mathcal{N}(0, 0.05),\ D_i = [0,1], i = 1..2, D_Y = [0,1]$$

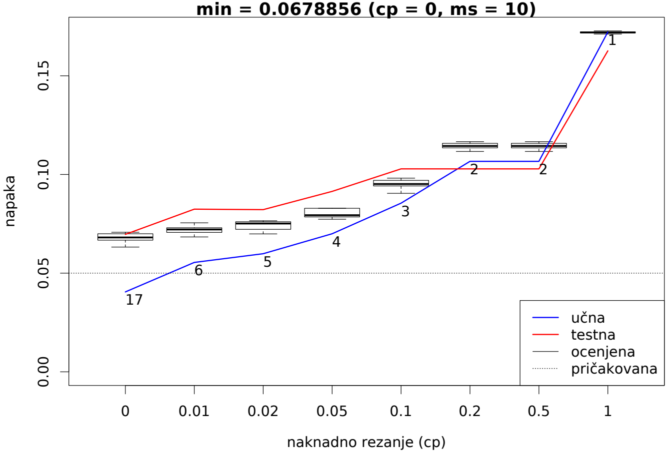
Regresijska drevesa ($cp = 0$): ms

Številke: število končnih vozlišč v drevesu



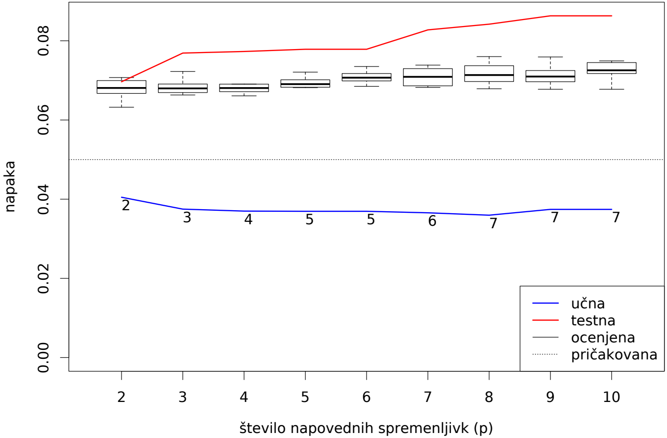
Regresijska drevesa ($ms = 10$): cp

Številke: število končnih vozlišč v drevesu

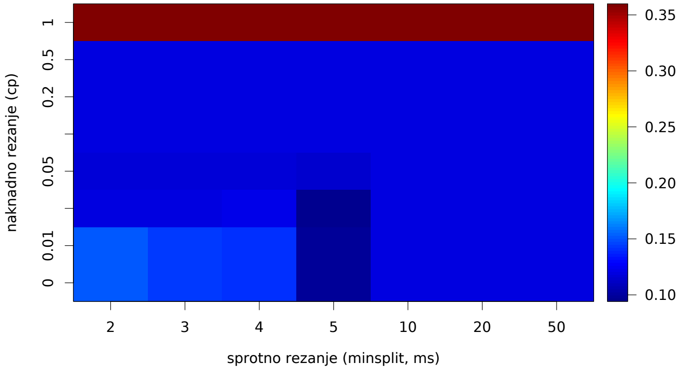


Regresijska drevesa: število napovednih spremenljivk

Številke: število napovednih spremenljivk v drevesu



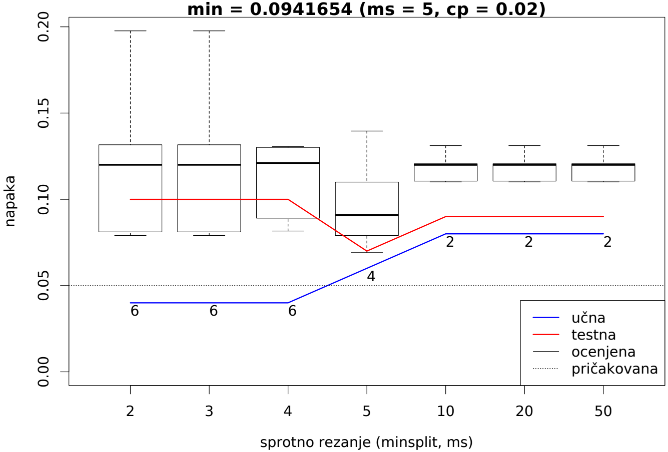
Klasifikacijska drevesa: *ms* in *cp*



$Y = I((1 + X_1 + X_1X_2)/3 \geq 0.5), D_i = [0, 1], i = 1..2, D_Y = \{0, 1\}$
zamenjamo vrednost Y 0.05 naključno izbranim primerom

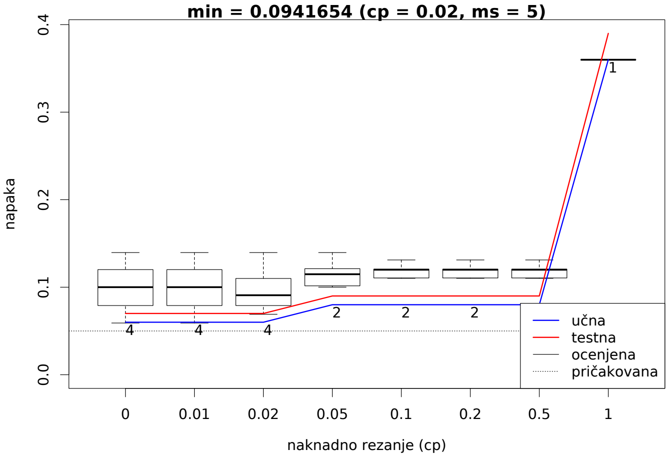
Klasifikacijska drevesa ($cp = 0.02$): ms

Številke: število končnih vozlišč v drevesu



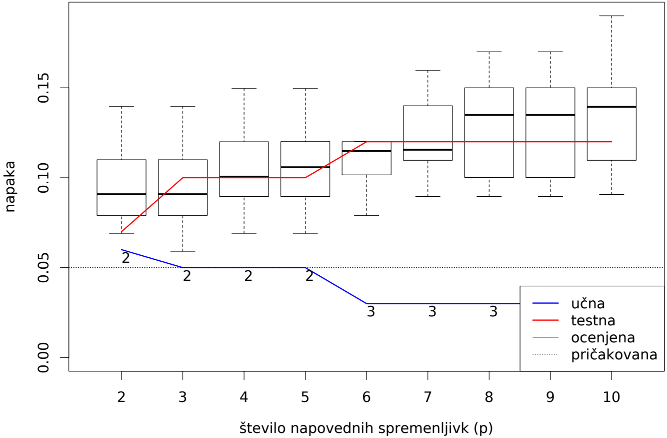
Klasifikacijska drevesa ($ms = 5$): cv

Številke: število končnih vozlišč v drevesu



Klasifikacijska drevesa: število napovednih spremenljivk

Številke: število napovednih spremenljivk v drevesu



Časovna zahtevnost algoritmov za učenje dreves

Učenje drevesa iz podatkovne množice S

- Eno vozlišče: $O(p|S|)$, p je število napovednih spremenljivk
- Celotno drevo: $O(p|S| \log |S|)$, ker je $\log |S|$ pričakovana globina

Naknadno rezanje

- Eno vozlišče: $O(\log |S|)$
- Celotno drevo: $O(\log |S| \log |S|)$

Znani algoritmi in implementacije

CART (Breiman in ost. 1984)

Na voljo knjižnica za R.

C4.5 (Quinlan 1995), pozneje C5.0

Na voljo v zbirki Weka kot J4.8 (C4.5), na voljo knjižnica za R (C5.0).

Kaj so odločitvena pravila?

Eno pravilo r : *IF Pogoj THEN Napoved*

- Pogoj je konjunkcija testov napovednih spremenljivk X, npr. $X_1 > 3 \ \wedge \ X_3 = ne$
- Napoved poda napovedano vrednost ciljne spremenljivke Y, npr. $\hat{Y} = da$

Urejena (tudi zverižena) množica pravil

Napoved za $\boldsymbol{x}$ poda prvo pravilo r, za katerega velja: $Pogoj(\boldsymbol{x}) = True$.

Neurejena množica pravil

Napovedi za $\boldsymbol{x}$ podajo vsa pravila r, za katera velja $Pogoj(\boldsymbol{x}) = True$, množica napove povprečje (regresija) ali večinsko napoved (klasifikacija).

Primera množice pravil

Urejena množica pravil

$$\begin{array}{lll} IF\ skrge = da & THEN\ delfin = ne & ELSE \\ IF\ zob = veliko & THEN\ delfin = da & ELSE \\ IF\ dolzina = 4 & THEN\ delfin = ne & ELSE \\ delfin = da & & \end{array}$$

Neurejena množica pravil

$$\begin{array}{ll} IF\ dolzina = 3 & THEN\ delfin = da \\ IF\ skrge = ne\ \wedge\ zob = veliko & THEN\ delfin = da \end{array}$$

Učenje enega pravila $LearnRule(S) = DecisionRule$

function $LearnRule(S)$
  $Cond = True$
  while $Impurity(S, Cond) > 0$ **do**
    $L_{opt} = \arg\min_L Impurity(S, Cond \wedge L)$
    $Cond = Cond \wedge L_{opt}$
  return $DecisionRule : IF\ Cond\ THEN\ Y = Prediction(S, Cond)$

Napoved pravila *Class*$(S, Cond)$

$$S_{Cond} = \{(\boldsymbol{x}, y) \in S : Cond(\boldsymbol{x}) = True\}$$

Regresija, $D_Y \subseteq \mathbb{R}$: povprečje

$$Prediction(S, Cond) = \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S_{Cond}} y$$

Klasifikacija, $D_Y = \{v_1, v_2, \ldots v_c\}$: večinski razred

$$Prediction(S, Cond) = \arg\max_{v \in D_Y} p(Y = v | S_{cond})$$

Funkcija nečistoče $Impurity(S, Cond)$

Funkcija nečistoče meri varianco vrednosti ciljne spremenljivke Y v množici *pokritih* primerov $S_{Cond} = \{(\boldsymbol{x}, y) \in S : Cond(\boldsymbol{x}) = True\}$.

Regresija, $D_Y \subseteq \mathbb{R}$

$$Impurity(S, Cond) = \frac{1}{|S|} \sum_{(\boldsymbol{x},y) \in S_{Cond}} (y - \bar{y})^2$$

Klasifikacija, $D_Y = \{v_1, v_2, \ldots v_c\}$

$$Impurity(S, Cond) = 1 - \max_{v \in D_Y} p(Y = v | S_{Cond})$$

Prekrivni algoritem $Covering(S) = RuleSet$

Za učenje urejene množice pravil

function $Covering(S)$
  $RuleSet = []$
  while $S \neq \emptyset$ **do**
    $Rule = LearnRule(S)$
    $S = S \setminus S_{Rule.Cond}$
    append $Rule$ **to** $RuleSet$
  return $RuleSet$

Spremembe za neurejene množice pravil

- V vsaki iteraciji izberemo razred v za katerega se učimo pravilo
- Posebna varianta $LearnRule(S, v)$, ki se uči pravilo za izbran razred v
- $S = S \setminus S_{Rule.Cond \wedge Y = v}$

Ali lahko prepoznam delfina?

primer	dolzina	skrge	kljun	zob	delfin
x_1	3	ne	da	veliko	da
x_2	4	ne	da	veliko	da
x_3	3	ne	da	malo	da
x_4	5	ne	da	veliko	da
x_5	5	ne	da	malo	da
x_6	5	da	da	veliko	ne
x_7	4	da	da	veliko	ne
x_8	5	da	ne	veliko	ne
x_9	4	da	ne	veliko	ne
x_{10}	4	ne	da	malo	ne

Prva iteracija algoritma *LearnRule*(S)

Na začetku $Cond = True$, $S = \{x_1, x_2, \ldots x_{10}\}$

L	S_{Cond}	p_{da}	p_{ne}	$Impurity(S, Cond)$
$dolzina = 3$	$\{x_1, x_3\}$	1	0	0
$dolzina = 4$	$\{x_2, x_6, x_9, x_{10}\}$	0.25	0.75	0.25
$dolzina = 5$	$\{x_4, x_5, x_6, x_8\}$	0.5	0.5	0.5
$skrge = da$	$\{x_6, x_7, x_8, x_9\}$	0	1	0
$skrge = ne$	$\{x_1, \ldots x_5, x_{10}\}$	0.83	0.17	0.17
$kljun = da$	$\{x_1, \ldots x_7, x_{10}\}$	0.625	0.375	0.375
$kljun = ne$	$\{x_8, x_9\}$	0	1	0
$zob = veliko$	$\{x_1, x_2, x_4, x_6, \ldots x_9\}$	0.43	0.57	0.43
$zob = malo$	$\{x_3, x_5, x_{10}\}$	0.67	0.33	0.33

Prva iteracija algoritma *Covering*(S)

Izbranoo pravilo *IF skrge* $=$ *da THEN delfin* $=$ *ne*

- Ker od vseh pravil z *Impurity* $= 0$ pokrije največ primerov (4)
- Ostane torej šest primerov $S = \{x_1, \ldots x_5, x_{10}\}$

Prva iteracija algoritma $LearnRule(S)$, drugič

Na začetku $Cond = True$, $S = \{x_1, \ldots x_5, x_{10}\}$

L	S_{Cond}	p_{da}	p_{ne}	$Impurity(S, Cond)$
$dolzina = 3$	$\{x_1, x_3\}$	1	0	0
$dolzina = 4$	$\{x_2, x_{10}\}$	0.5	0.5	0.5
$dolzina = 5$	$\{x_4, x_5\}$	1	0	0
$kljun = da$	$\{x_1, \ldots x_5, x_{10}\}$	0.83	0.17	0.17
$kljun = ne$	$\emptyset$	NaN	NaN	NaN
$zob = veliko$	$\{x_1, x_2, x_4\}$	1	0	0
$zob = malo$	$\{x_3, x_5, x_{10}\}$	0.67	0.33	0.33

Druga iteracija algoritma *Covering*(S)

Izbrano pravilo $IF\ zob = veliko THEN\ delfin = da$

- Pokriva tri primere
- Ostanejo trije primeri $S = \{x_3, x_5, x_{10}\}$

Trenutna (urejena) množica pravil

$$\begin{array}{lll} IF\ skrge = da & THEN\ delfin = ne & ELSE \\ IF\ zob = veliko & THEN\ delfin = da & \end{array}$$

Prva iteracija algoritma *LearnRule*(S), tretjič

Na začetku $Cond = True$, $S = \{x_3, x_5, x_{10}\}$

L	S_{Cond}	p_{da}	p_{ne}	$Impurity(S, Cond)$
$dolzina = 3$	$\{x_3\}$	1	0	0
$dolzina = 4$	$\{x_{10}\}$	0	1	0
$dolzina = 5$	$\{x_5\}$	1	0	0
$kljun = da$	$\{x_3, x_5, x_{10}\}$	0.67	0.33	0.33
$kljun = ne$	$\emptyset$	NaN	NaN	NaN

Tretja iteracija algoritma *Covering*(S)

Izbrano pravilo $IF\ dolzina = 4\ THEN\ delfin = ne$

- Pokriva en primer
- Ostaneta dva pozitivna primera $S = \{x_3, x_5\}$, čista množica

Končna (urejena) množica pravil

$$\begin{array}{lll} IF\ skrge = da & THEN\ delfin = ne & ELSE \\ IF\ zob = veliko & THEN\ delfin = da & ELSE \\ IF\ dolzina = 4 & THEN\ delfin = ne & ELSE \\ delfin = da & & \end{array}$$

Kontrola pristranskosti in variance

Velikost množice pravil

- število pravil
- število testov v pravilih

Kontrola velikosti

1. Sprotno rezanje: predčasni izhod iz iteracije *LearnRule*
2. Naknadno rezanje: odstranjevanje pravil ali posameznih testov

Znani algoritmi in implementacije

CN2 (Clark and Nibblet 1987)

Na voljo knjižnica za R.

Ripper (Cohen 1995)

Na voljo v zbirki Weka, za uporabo v R na voljo skozi knjižnico RWeka.