

Strojové učení

Jindřich Matuška

Faculty of Informatics, Masaryk University

5. prosince 2024

Čas na odpovědníky

Obsah

Rozhodovací stromy, učení rozhodovacích stromů

Perceptron, lineární klasifikace

Základy lineární regrese

Strojové učení

Systemy, které se učí z poskytnutých dat

- Učení s učitelem – pro vstupy jsou známé očekávané výstupy
- Učení bez učitele – model se učí vzory v datech
- Zpětnovazebné učení – akce provedené modelem jsou ohodnoceny, model se učí na základě těchto odměn

Základní úlohy strojového učení

- Klasifikace – rozdělení nových dat do předem určených tříd
- Regrese – modelování spojitě proměnné
- Shlukování – rozdělení dat do několika skupin se vzájemně podobnými atributy

Obsah

Rozhodovací stromy, učení rozhodovacích stromů

Perceptron, lineární klasifikace

Základy lineární regrese

Rozhodovací strom

Model především pro klasifikaci

- Vnitřní větve obsahují testovaný atribut
- Listové uzly obsahují hodnoty (třídy či diskrétní hodnoty)
- Hrany odpovídají možným hodnotám testovaného atributu

Rozhodovací strom je konzistentní s datasetem právě tehdy, když správně klasifikuje všechny příklady z datasetu.

Příklad 10.1.1

Mějme dataset D zadaný následující tabulkou (řádky reprezentují jednotlivé příklady, sloupce atributy, závislá proměnná je *Tenis*)

<i>Vlhkost</i>	<i>Počasí</i>	<i>Větrno</i>	<i>Tenis</i>
vysoká	zataženo	ano	NE
střední	slunečno	ne	ANO
nízká	slunečno	ano	ANO
střední	děšť	ne	NE

- Vytvořte (ručně) rozhodovací strom, který je konzistentní s tímto datasetem. Snažte se o co nejmenší strom.
- Jak by (konzistentní) strom vypadal, pokud by v D byl navíc ještě příklad:

střední, slunečno, ne | NE?

Odbočka – chyby, modely, validace

Data obsahují chyby měření.

Hledáme funkci T , máme však jen nepřesná data $f(X)$

$$f(X) \in T(X) + e(X)$$

Vytvořit k datům konzistentní model je jednoduché, ale pravděpodobně overfittuje.

Učení rozhodovacích stromů

Problém nalezení nejmenšího konzistentního modelu je NP-úplný, budeme tedy používat heuristiky.

```
def generateDT(data):  
    node = Node()  
    node.attribute = getBestAttribute(data)  
    for value, subset in data.splitAttribute(node.attribute):  
        node.childs[value] = generateDT(subset)  
    return node
```

Entropie

Míra neuspořádanosti, nejistoty, informace.

Nechť D je dataset s příklady rozdělenými do n tříd, P_i je pravděpodobnost náhodného výběru prvku třídy i .

$$E(D) = \sum_{i=1}^n -P_i \cdot \log_2(P_i)$$

Informační zisk

Jak moc se liší entropie původního datasetu a rozděleného datasetu.

Nechť D je dataset s atributem A nabývajícím k hodnot. Atribut A rozděluje množinu D na podmnožiny D_1, \dots, D_k .

$$Gain(D, A) = E(D) - \sum_{i=1}^k \frac{||D_i||}{||D||} \cdot E(D_i)$$

Příklad 10.2.1, 10.2.2

Uvažujme dataset daný tabulkou. Nalezněte atribut, který by učící algoritmus zvolil jako rozhodovací atribut pro kořen stromu.

Následně pro tento dataset vybudujte kompletní rozhodovací strom.

	<i>Vlhkost</i>	<i>Počasí</i>	<i>Větrno</i>	<i>Teplota</i>	<i>Tenis</i>
1	vysoká	zataženo	ano	nižší	NE
2	střední	zataženo	ne	vyšší	ANO
3	nízká	slunečno	ano	nižší	ANO
4	střední	déšť	ne	nižší	NE
5	nízká	slunečno	ano	vyšší	ANO
6	vysoká	déšť	ne	vyšší	NE
7	střední	slunečno	ano	nižší	ANO

Obsah

Rozhodovací stromy, učení rozhodovacích stromů

Perceptron, lineární klasifikace

Základy lineární regrese

Perceptron (neuron)

Lineární model, na výslednou hodnotu aplikuje *aktivační funkci*.
Pokud používáme perceptron jako klasifikátor, označujeme aktivační funkci jako prahovou funkci.

Uvažme vstupní vektor $\vec{x} = \langle x_1, \dots, x_n \rangle$ a vektor vah $\vec{w} = \langle w_0, w_1, \dots, w_n \rangle$. Výstup perceptronové jednotky s váhami \vec{w} pro příklad \vec{x} je

$$C[\vec{w}](\vec{x}) = \begin{cases} 1 & \text{pokud } w_0 + \sum_{i=1}^n w_i \cdot x_i = \vec{w} \cdot \vec{x} \geq 0 \\ 0 & \text{jinak} \end{cases}$$

Perceptron často značíme pomocí následujícího schématu:

Příklad 10.3.1

Pro každou z následujících logických operací definujte perceptron, který ji implementuje. Jako množinu vstupů uvažujte pouze vektory nad $\{0, 1\}$.

- Binární NAND
- Binární implikace
- n -ární disjunkce

Geometrický význam perceptronu

$$C[\vec{w}](\vec{x}) = \begin{cases} 1 & \text{pokud } w_0 + \sum_{i=1}^n w_i \cdot x_i = \vec{w} \cdot \tilde{x} \geq 0 \\ 0 & \text{jinak} \end{cases}$$

Perceptron je lineární separátor.

Dělicí nadrovina je množina bodů $\{\vec{x} \mid \vec{w} \cdot \tilde{x} = 0\}$

Perceptron je konzistentní s datasetem, pokud klasifikuje všechny prvky datasetu správně.

Online perceptronový algoritmus

Iterativní proces nastavení správných vah perceptronu.

Pro tréninkový set $D = \{(\vec{x}_1, c_1), \dots, (\vec{x}_p, c_p)\}$, kde c_i je opravdová hodnota příkladu x_i je sekvence vektorů vah počítána následovně:

- $\vec{w}^{(0)}$ je inicializována náhodně s hodnotami okolo 0,
- $\vec{w}^{(t+1)} = \vec{w}^{(t)} - \alpha \cdot (C[\vec{w}^{(t)}](\vec{x}_k) - c_k) \cdot \vec{x}_k$,

kde $k = (t \bmod p) + 1$,

$0 < \alpha \leq 1$ je konstanta učení

Příklad 10.4.2

Mějme trénovací sadu

$$D = \{(\langle 3, -1 \rangle, 1), (\langle 2, 1 \rangle, 1), (\langle 0, 3 \rangle, 0)\}$$

Aplikujte perceptronový algoritmus, dokud nenalezne separující nadrovinu. Uvažujte $\vec{w}^{(0)} = \langle 0, -2, 1 \rangle$ a $\alpha = 1$. Na závěr načrtněte dělící přímku.

Obsah

Rozhodovací stromy, učení rozhodovacích stromů

Perceptron, lineární klasifikace

Základy lineární regrese

Lineární regrese

Klasifikace – rozřazení dat do tříd

Regrese – modelování neznámé funkce

Lineární regrese používá model podobný perceptronu:

$$\vec{R}[\vec{w}](\vec{x}) = \vec{w} \cdot \vec{x}$$

Data prokládáme co nejlepším lineárním modelem

Chybová funkce

Chybová funkce vyjadřuje jak moc se model blíží datům.
Menší chyba znamená lepší model.

Kvadratická chyba

Uvažme dataset $D = \{(\vec{x}_1, f_1), \dots, (\vec{x}_p, f_p)\}$, kde f_i je opravdová hodnota příkladu \vec{x}_i . Kvadratická chyba pro model s váhami \vec{w} je

$$E(\vec{w}) = \frac{1}{2} \cdot \sum_{k=1}^p (R[\vec{w}](\vec{x}_k) - f_k)^2 = \frac{1}{2} \cdot \sum_{k=1}^p (\vec{w} \cdot \vec{x}_k - f_k)^2$$