Neparametarsko učenje — najbliži susjedi

Poglavlje 18.8

Prema slajdovima Stuarta Russela, Krunoslava Puljića, Tomislava Šmuca, Hantao Zhanga (hvala)!

Sadržaj

- ♢ Neparametarsko učenje pod nadzorom — pretraga
- ♢ Najbliži susjedi
- ♢ Ubrzanja

Štreber

Neparametarski model — nema sažeti zapis preko parametara
- svaka hipoteza sadrži svojstva svih primjera,
- stvarni broj parametara je neograničen, raste s brojem primjera.

Najlakša metoda = učenje svih primjera "napamet".

- Štreber (engl. Rote learner).
- Pohrani u memoriju sve što trebaš naučiti
- Kad te netko pita za klasu novog objekta, pogledaj u memoriju
  - Ako objekt postoji u memoriji, vrati njegovu klasu (hipotezu)
  - Ako ne, reci da ne znaš.
- Ne možemo reći da je štrebanje učenje (no, katkad je nužno).

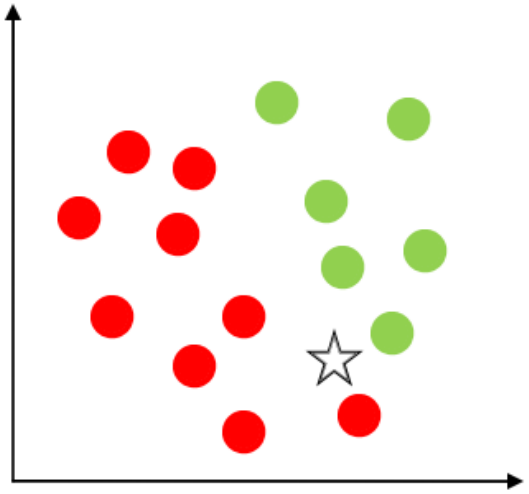
Štreber = pretraživanje cijele tablice (bez reda) — nema generalizacije!

K najbližih susjeda

K najbližih susjeda (engl. K–nearest neighbours), oznaka: k–NN.

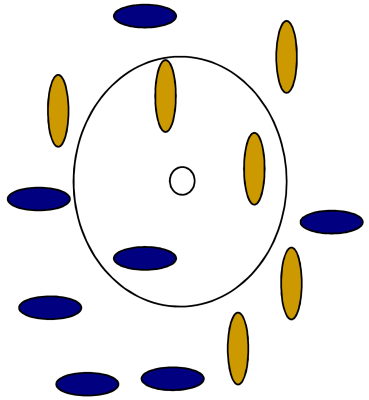
- Slično štreberu, s bitnom razlikom — koristi "blizinu" ili "sličnost".
- Za dani upit = novi primjer, odgovor je
  – ona klasa objekta koja je najzastupljenija među k najbližih susjeda u prostoru poznatih primjera.

Broj k susjeda je zadan (može se dinamički mijenjati).

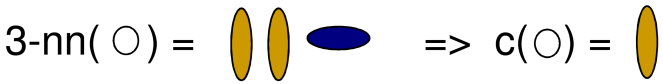


Algoritam za k–NN

- Izračuna se udaljenost između novog primjera x i svih primjera x_i iz skupa za učenje T.
- Odredi se k najbližih susjeda od x iz T.
- Pridjeli x onu klasu koja je najčešća među njegovih najbližih k susjeda iz T.



Primjer za $k = 3$.



Algoritam za k–NN (nastavak)

- Udaljenost između primjera x i x_i iz T:

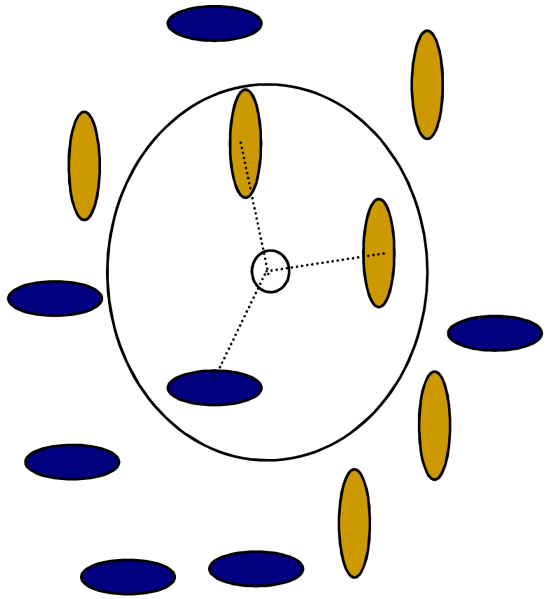
$$D(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{i,j})^2}.$$

- Ako su $i = 1, 2, \ldots, k$ indeksi k najbližih susjeda objektu x, onda je pripadna klasa:

$$\hat{f}(x) = \arg\max_{c} \sqrt{\sum_{i=1}^{k} \delta(c, f(x_i))},$$

gdje je

$$\delta(x, y) = \begin{cases} 1, & \text{za } x = y, \\ 0, & \text{za } x \neq y. \end{cases}$$



Algoritam za k–NN — problem udaljenosti

Problem izbora udaljenosti između primjera — standardna bi bila

$$D(x, x_i) = \sqrt{\sum_{j=1}^{n} (x_j - x_{i,j})^2}.$$

X_1	X_2	X_3	X_4	X_5	X_6	X_7	y
Godine	**Spol**	**Brak**	**Obrazovanje**	**Broj djece**	**Regija**	**Primanja (HRK/god)**	**Klasa G(1) / N(0)**
26	m	Da	sš	1	I	88000	0
34	ž	Ne	vss	2	S	65000	1
56	ž	Da	ss	4	J	135000	0
68	m	Ne	vss	1	Z	45000	1
19	m	Ne	ss	0	C	33000	1
.....	.....		.....	.....	.....		.....

Algoritam za k–NN — problem udaljenosti

Problem udaljenosti između primjera P = Pero, M = Matilda:

$$D(\mathrm{P},\mathrm{M}) = \sqrt{(x_1(\mathrm{P}) - x_1(\mathrm{M}))^2 + (x_2(\mathrm{P}) - x_2(\mathrm{M}))^2 + \cdots}$$

$$D(\mathrm{P},\mathrm{M}) = \sqrt{(26-34)^2 + (\mathrm{m} - \check{\mathrm{z}})^2 + (\mathrm{Da} - \mathrm{Ne})^2 + (\mathrm{S\check{S}} - \mathrm{VSS})^2 + \cdots}$$

???

X_1	X_2	X_3	X_4	X_5	X_6	X_7	y
Godine	**Spol**	**Brak**	**Obrazovanje**	**Broj djece**	**Regija**	**Primanja (HRK/god)**	**Klasa G(1) / N(0)**
26	m	Da	sš	1	I	88000	0
34	ž	Ne	vss	2	S	65000	1
19	m	Ne	ss	0	C	33000	1
.....	.....		.....	.....	.....		.....

Algoritam za k–NN — problem udaljenosti

Numerički atributi mogu imati vrlo različite intervale vrijednosti (na pr., godine, djeca, primanja — što nema puno smisla).

Zato treba napraviti normalizaciju numeričkih atributa X_j na jedinični interval $[0, 1]$ — na primjer, afinom transformacijom

$$x'_{i,j} = (x_{i,j} - \min_i x_{i,j}) \,/\, (\max_i x_{i,j} - \min_i x_{i,j}).$$

Udaljenost za kategoričke atribute = diskretna (trivijalna metrika):

$$x_j = x_{i,j} \Rightarrow x_j - x_{i,j} = 0, \quad x_j \neq x_{i,j} \Rightarrow x_j - x_{i,j} = 1.$$

K-nn algoritam možemo isto tako koristiti i za regresijske probleme. Kako izgleda $f(x)$ u tom slučaju?

Poboljšanje algoritma k–NN

Težinsko određivanje klase — uzima u obzir i udaljenost svakog od najbližih k susjeda do novog primjera

$$\hat{f}(x) = \arg\max_c \sqrt{\sum_{i=1}^{k} w_i \, \delta(c, f(x_i))},$$

gdje je težina w_i obratno proporcionalna udaljenosti

$$w_i \equiv \frac{1}{(d(x, x_i))^2}.$$

Ideja = dalji susjed ima manju težinu.

Metoda k–NN — problemi

- Koji k je optimalan?
- Veliki broj atributa — puno nevažnih, a u određivanju udaljenosti svi imaju istu težinu!
  Rješenje:
  – Težinski faktori za svaki atribut/varijablu.
  – Problem: Svaki atribut dobiva vlastitu težinu — kako ju odrediti?
- Veliki broj primjera u skupu za učenje — složenost:
  – Za svaki novi primjer koji želimo klasificirati, moramo odrediti udaljenost do svih primjera u memoriji i naći najbliže susjede!
  Rješenja:
  – Indeksiranje primjera (k–d-trees metoda).
  – Selektivno spremanje primjera za učenje (redukcija podataka).

Metoda k–NN — dobre i loše strane

- k–NN — dobre strane:
  - Vrlo jednostavan.
  - Relativno robustan na šum u podacima (osim za 1–NN)!
  - Prilično dobri rezultati se postižu ako imamo dovoljno velik skup primjera za učenje.

- k–NN — loše strane:
  - Vremenski zahtjevan za velik broj primjera u skupu za učenje!

Složenost je linearna u broju primjera (sekvencijalno traženje).

Napomena: Standardno, u algoritmima koji se koriste u UI,
  - linearna složenost je dobra, prema NP–teškim problemima.

Ovdje ne — jer je riječ o "običnom" pretraživanju!

Ubrzanja pretrage — k–d stabla

Ideja: dobrom organizacijom dobiti logaritamsku složenost traženja.

Organizacija = k–dimenzionalno binarno stablo
- svaka razina = ravnomjerna podjela vrijednosti u nekoj dimenziji,
- izbarana dimenzija varira po razinama.

Ako imamo k dimenzija (atributa), ovo se isplati kada je broj podataka reda veličine barem 2^k.