

## 8. Učenje na temelju primjera (engl. instance-based learning)

### 8.1 Uvod

Metode koje smo spominjali do sada pokušavaju konstruirati općeniti, eksplicitni opis neke ciljne funkcije.

**Metode učenja na temelju primjera** pohranjuju primjere za učenje.

Postupak generalizacije odgođen je do trenutka potrebe za klasifikacijom novog uzorka. Tada se ispituje odnos novog uzorka prema pohranjenim primjerima za učenje da bi se odredila vrijednost ciljne funkcije na novom primjeru.

- **Metoda  $k$ -najbližih susjeda** (engl. *k-nearest neighbor*)
- **Metoda lokalne regresije s težinskim faktorima** (engl. *locally weighted regression*)
- **Zaključivanje na temelju slučajeva** (engl. *case based reasoning*)
- **Radijalne bazne funkcije** (engl. *radial basis functions*)

Prve dvije metode: primjeri su točke euklidskog prostora. Metoda zaključivanja na temelju slučajeva (engl. *case-based reasoning*) koristi složeniju simboličku reprezentaciju primjera za učenje.

Podjela metoda za učenje:

***lazy methods vs. eager methods***

odgađaju odluku o klasifikaciji do trenutka predočavanja novog primjera-upita (primjer takvih metoda su: metoda  $k$ -najbližih susjeda, metoda lokalne regresije s težinskim faktorima i zaključivanje na temelju slučajeva).

***Eager methods*** su sve do sada iznesene metode kao što su neuronske mreže, ID3 algoritam te od gornje navedenih, radijalne bazne funkcije.

Dvije važne razlike (prednosti) *lijenih* metoda, tj. metoda s odgodom, naspram ostalih metoda:

1. konstruiraju različitu aproksimaciju ciljne funkcije za svaki različiti novi upit-primjer koji treba biti klasificiran;
2. umjesto procjene ciljne funkcije, jednom za cijeli prostor, te metode procjenjuju ciljnu funkciju samo lokalno, u okolini novog primjera. Takva lokalna procjena ciljne funkcije je pogodna za vrlo kompleksne ciljne funkcije.

Nedostatak metoda učenja na temelju primjera:

1. visoka cijena klasificiranja novog primjera. (računanje se dešava prilikom predočavanja novog upita umjesto prethodno, na primjerima za učenja kao kod ostalih metoda)
2. razmatraju se svi atributi nekog primjera prilikom klasifikacije iako samo neki mogu imati utjecaj na ciljnu funkciju ( $k$ -najbližih susjeda).

## 8.2 Metoda k-najbližih susjeda

Primjeri su točke u n-dimenzionalnom prostoru  $R^n$ .  
Udaljenost - Euklidska metrika. Ciljna funkcija - diskretne ili realne vrijednosti.

Primjer  $x$  je opisan vektorom značajki

$$[a_1(x), a_2(x), \dots, a_n(x)],$$

gdje je  $a_k(x)$  označava k-ti atribut primjera  $x$ .

Euklidska udaljenost između dva vektora  $x_i$  i  $x_j$  je

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n [(a_r(x_i) - a_r(x_j))]^2}$$

Za ciljnu funkciju sa diskretnim vrijednostima:  
 $f: R^n \rightarrow V$ , gdje je  $V = \{v_1, v_2, \dots, v_s\}$

### Algoritam k najbližih susjeda (k-nn).

Algoritam za učenje

- Za svaki primjer za učenje  $(x, f(x))$  dodaj primjer na listu primjeri\_za\_učenje

Algoritam klasifikacije

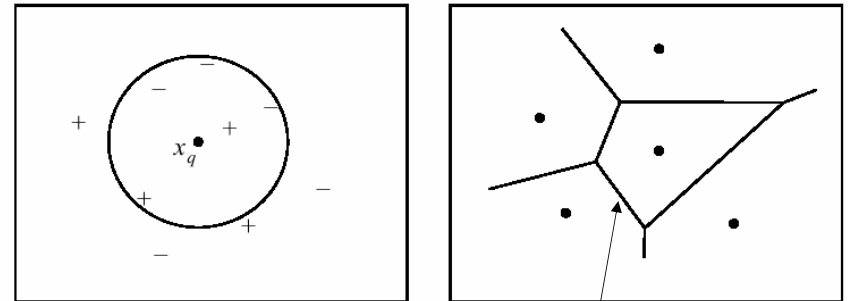
- Za dani primjer  $x_q$  s nepoznatom klasifikacijom
  - Neka  $x_1, x_2, \dots, x_k$  označavaju k primjera koji su najbliži  $x_q$ .
  - Vрати

$$\hat{f}(x_q) \leftarrow \arg \max_{v \in V} \sum_{i=1}^k \delta(v, f(x_i))$$

gdje je  $\delta(a, b) = 1$  ako  $a = b$ , 0 inače.

$\hat{f}(x_q)$  je najčešća vrijednost ciljne funkcije koja se pojavljuje među k primjera za učenje koji su najbliži upitu  $x_q$ .

Uoči razliku rezultata kod 1-nn i 5-nn algoritma.



k-nn algoritam nikad ne oblikuje eksplicitnu hipotezu za ciljnu funkciju  $f$ .

Za 1-nn možemo je predočiti Veronoi dijagramom. Decizijska površina je konveksni poliedar koji okružuje svaki primjer za učenje.

Prilagodba k-nn algoritma za ciljnu funkciju realnih vrijednosti  $f: R^n \rightarrow R$ , umjesto najčešće pojavljivane vrijednosti ciljne funkcije odgovor na upit je srednja vrijednost ciljnih funkcija k najbližih susjeda.

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k f(x_i)}{k}$$

### 8.2.1 Modifikacija k-nn algoritam uvođenjem težinskih faktora udaljenosti

Poboljšanje k-nn → uvođenje težinskih faktora  $w_i$  za svaki od k susjeda, koji ovisi o njegovoj udaljenosti od upita  $x_q$ .

$$\hat{f}(x_q) \leftarrow \arg \max_{v \in V} \sum_{i=1}^k w_i \delta(v, f(x_i)), \text{ gdje je}$$

$$w_i = \frac{1}{d(x_q, x_i)^2}$$

U slučaju  $x_i = x_q$  tada pridružujemo  $\hat{f}(x_q)$  vrijednost  $f(x_i)$ .

Modifikacija k-nn u slučaju kontinuirane ciljne funkcije.

$$\hat{f}(x_q) \leftarrow \frac{\sum_{i=1}^k w_i f(x_i)}{\sum_{i=1}^k w_i}$$

Sada nema razloga da uzimamo u obzir cijeli prostor primjera, ali je očito da će udaljeni primjeri imati vrlo mali utjecaj na klasifikaciju.

Ovakva globalna metoda naziva se **Shepard-ova** metoda.

### 8.2.2 Primjedbe na k-nn algoritam

- efikasna induktivna metoda
- robusna na šum u primjerima za učenje

Induktivna pristranost:

pretpostavka da je klasifikacija upita  $x_q$  slična klasifikaciji primjera u blizini.

Udaljenost se računa na temelju svih atributa (za razliku od ID3 ili učenja skupova pravila koji selektiraju podskupove atributa pri formiranju hipoteze).

«*Curse of dimensionality*» – osjetljivost k-nn na sve atribute bez obzira na dimenziju prostora (broj atributa) i njihov značaj za ciljnu funkciju.

Rješenje: rastezanje ili stiskanje osi Euklidskog prostora (množenje vrijednosti atributa s faktorima) da bi se smanjio utjecaj nevažnih atributa.

Praktična tema vezana za k-nn je efikasno indeksiranje memorije zbog brzog dohvata primjera kod novog upita.

### 8.2.3 Nazivlje

Metode s odgodom - područje statističkog raspoznavanja uzoraka.

Regresija - način aproksimacije ciljne funkcije s realnim vrijednostima.

Rezidual (ostatak) - pogreška  $\hat{f}(x) - f(x)$ .

Jezgrena funkcija (*engl. kernel function*) - funkcija udaljenosti koja se koristi za određivanje težinskih faktora primjera za učenje, tj. jezgrena funkcija **K** je takva da je

$$w_i = K(d(x_i, x_q)).$$

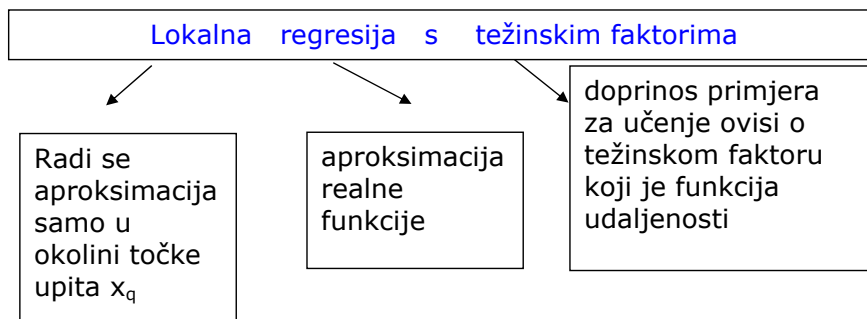
### 8.3 Lokalna regresija s težinskim faktorima (*engl. locally weighted regression*)

Algoritam k-nn se može interpretirati kao aproksimiranje ciljne funkcije u  $f(x)$  u točki  $x = x_q$ .

Regresija s težinskim faktorima generalizacija je te metode jer konstruira eksplicitnu aproksimaciju ciljne funkcije na cijelom lokalnom području oko  $x_q$ .

Aproksimacija ciljne funkcije, može biti:

- linearnom funkcijom
- kvadratnom
- višeslojnom neuronskom mrežom

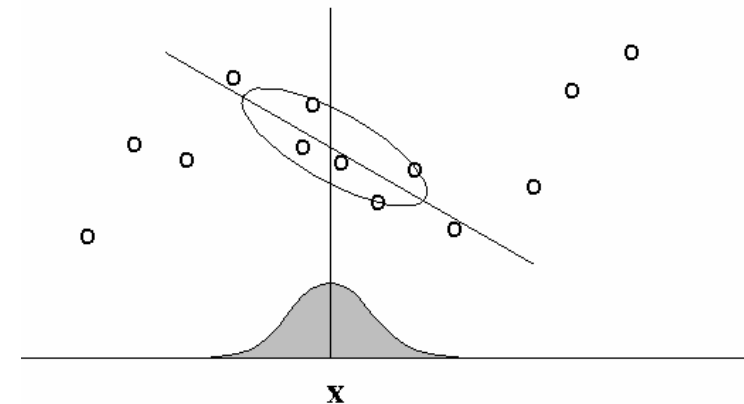


Neka je dan je upit  $x_q$

- konstruira se aproksimacije  $\hat{f}$  ciljne funkcije koja odgovara primjerima za učenje u okolini  $x_q$

- aproksimacija se koristi za izračun vrijednosti  $\hat{f}(x_q)$ .

### 8.3.1. Lokalna linearna regresija s težinskim faktorima



f aproksimiramo linearnom funkcijom

$$\hat{f}(x) = w_0 + w_1 a_1(x) + \dots + w_n a_n(x).$$

$a_i(x)$  označava vrijednost i-tog atributa primjera  $x$ .

Učenje neuronskih: mreža - metoda gradijentnog silaska za nalaženje koeficijenata  $w_0, \dots, w_n$  - tako da se minimizira suma kvadrata pogreške nad primjerima za učenje.

$$E = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x))^2$$

iz čega smo izveli:

$$\Delta W = \eta \sum_{x \in D} (f(x) - \hat{f}(x)) a_j(x)$$

(usporedi s  $\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id}$  iz poglavlja o neuronskim mrežama).

Ovo gore je metoda globalne aproksimacije. Kako je prilagoditi da izvedemo lokalnu aproksimaciju?

Tri moguća kriterija prilagodbe ove metode za lokalnu aproksimaciju:

1. Minimizacija kvadrata pogreške samo nad  $k$  najbližih susjeda.

$$E_1(x_q) = \frac{1}{2} \sum_{x \in k \text{ najbližih susjeda od } x_q} (f(x) - \hat{f}(x))^2$$

2. Minimizacija kvadrata pogreške nad ciljnim skupom  $D$  uz umnožak s težinskim faktorima

$$E_2(x_q) = \frac{1}{2} \sum_{x \in D} (f(x) - \hat{f}(x_q))^2 K(d(x_q, x))$$

3. Kombinacija 1. i 2.

$$E_3(x_q) = \frac{1}{2} \sum_{x \in k \text{ najbližih susjeda od } x_q} (f(x) - \hat{f}(x))^2 K(d(x_q, x))$$

(Uoči da je pogreška definirana kao funkcija upita  $x_q$ )

Model pod 2 je računski najzahtjevniji. ako usvojimo 3. model, pravilo učenja je

$$\Delta W = \eta \sum_{x \in k \text{ najbližih susjeda od } x_q} K(d(x_q, x)) (f(x) - \hat{f}(x_q)) a_j(x)$$

## 8.4 Radijalne bazne funkcije

Metoda aproksimacije funkcije (povezana sa nn i lokalnom regresijom). Hipoteza je funkcija oblika:

$$\hat{f}(x) = w_0 + \sum_{u=0}^k w_u K_u(d(x_u, x)) \quad (1)$$

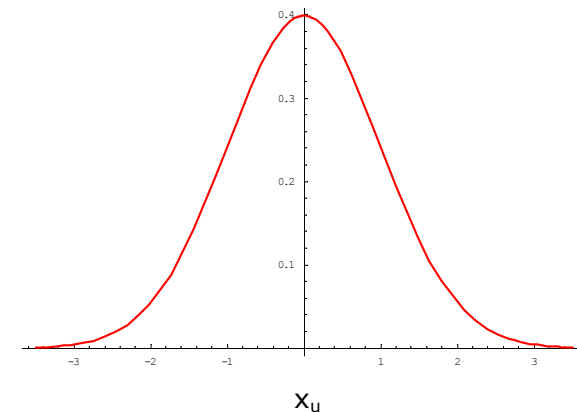
gdje su:

- $x_u$  primjeri iz  $x$ ,
- $K(d(x_u, x))$  jezgrena funkcija koja se smanjuje kada udaljenost raste.  $k$
- $k$  proizvoljan broj jezgrenih funkcija

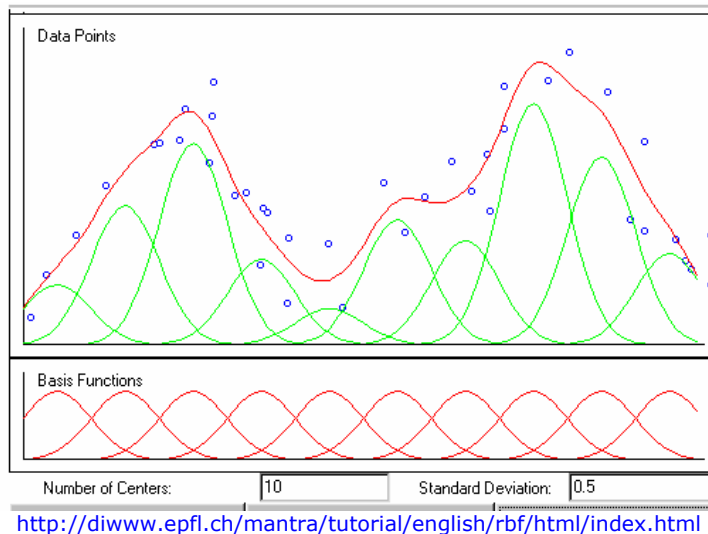
Iako je  $\hat{f}(x)$  globalna aproksimacija  $f(x)$ , doprinos svake  $K_u(d(x_u, x))$  je lokalna - samo u okolini  $x_u$ .

Uobičajen izbor za  $K_u(d(x_u, x))$  su Gaussove funkcije s centrom u  $x_u$  i varijancom  $\sigma^2$ .

$$K_u(d(x_u, x)) = e^{-\frac{1}{2\sigma_u^2} d^2(x_u, x)}$$

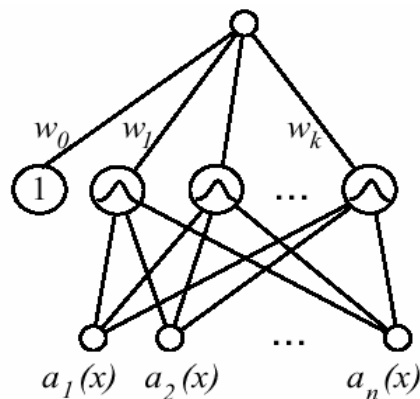


Prema (Hartman *et al.*, 1990) izraz (1) može aproksimirati bilo koju funkciju proizvoljno točno za dovoljno veliki broj Gaussovih jezgri uz uvjet da se varijance mogu nezavisno odrediti.



Funkcija (1) se može interpretirati kao dvoslojna neuronska mreža:

- prvi sloj računa  $K_u(d(x_u, x))$
- drugi sloj je linearna kombinacija vrijednosti prvog sloja



RBF mreže se treniraju u dva koraka:

1. određuje se broj skrivenih jedinica  $k$ , određuje se  $x_u$  i  $\sigma^2$  koji određuju jezgenu funkciju.
2. određuju se težinski faktori  $w_i$  tako da mreža odgovara podacima za učenje – na temelju minimizacije sume kvadrata pogreške. Za vrijeme te faze jezgrene funkcije se ne mijenjaju pa je učenje efikasno.

Nekoliko metoda za izbor broja  $k$ :

1. za svaki primjer za učenje  $(x_i, f(x_i))$  – jedna jezgrena funkcija s centrom u  $x_i$  i sa istim varijancama. na ovaj način RBFu potpunosti odgovara primjerima za učenje
2. broj jezgrenih funkcija < broj primjera za učenje. efikasniji način.
  - Centri RBFa mogu biti smješteni
    - uniformno po  $X$
    - neuniformno,
      - slučajnim izborom, izvlačeći primjere iz skupa za učenje u skladu s njihovom distribucijom
      - prototipovima grupa primjera za učenje (uz uporabu algoritma grupiranja)

Zaključak:

- RBF daju globalnu aproksimaciju ciljne funkcije kao linearnu kombinaciju više lokalnih jezgrenih funkcija.
- mogu biti trenirane efikasnije od unaprijednih neuronskih mreža s *backpropagation* algoritmom (algoritam radi u dva koraka).