Neparametrické metody

Neparametrické metody

- obsah
  - odhad funkce hustoty
    - metoda Parzenova okénka
    - pravděpodobnostní neuronové sítě
    - metoda k_n-nejbližších sousedů
  - odhad aposteriorní pravděpodobnosti

  - pravidlo k-nejbližších sousedů
    - vlastnosti
    - zlepšení výpočetní složitosti
    - metriky

Odhady hustoty

Neparametrické metody

- hustota rozdělení
  - 1. známá
  - 2. neznámá, ale známý tvar funkce
    - parametry lze odhadnout
    - pro „běžné“ funkce hustoty (např. normální rozdělení, ...)
  - 3. neznámá → nejčastěji
    - problém: hustota reálných dat málokdy „pasuje“ na některou z běžných hustot

→ neparametrické metody

- použití: pro libovolné rozdělení (+)
  - není třeba předpoklad o tvaru hustoty
- potřeba více trénovacích dat (–)

Neparametrické metody

- rozdělení metod
  - odhady funkce hustoty $p(\boldsymbol{x}|\omega_j)$
    - odhad hustoty na základě trénovacích dat
    - uspokojivé odhady → použity jako skutečné hodnoty do klasifikátoru
    - proces nutné provést pro každou třídu ω_j

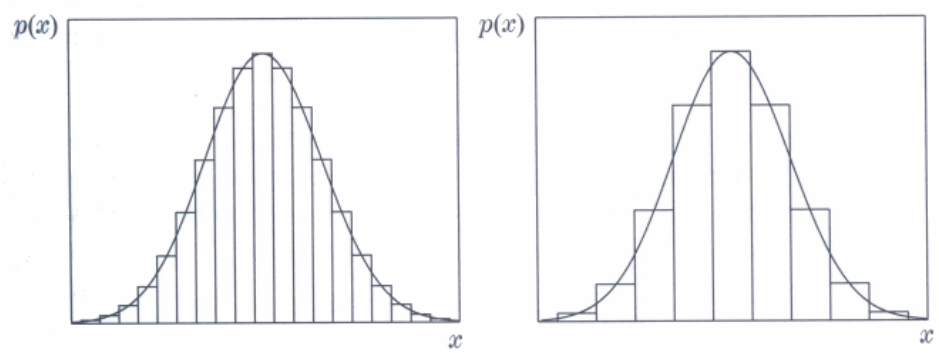
    → metoda histogramu, Parzenovo okénko

  - odhady apost. pravděpodobnosti $P(\omega_j|\boldsymbol{x})$
    - přímý odhad rozhodovacího pravidla

    → metody nejbližšího souseda

Odhady hustoty – myšlenka (1)

- princip metody
  - aproximace hustoty pomocí histogramu



- *P* ... pravděpodobnost, že vzor padne do přihrádky o rozměru *R*

$$P = \int_R p(\mathbf{x}')d\mathbf{x}' \qquad (1)$$

neznáme

- obráceně
  - když se odhadne *P* → lze odhadnout i *p(**x**)*

Odhady hustoty – myšlenka (2)

- trénovací vzory $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ vybrané nezávisle podle $p(\boldsymbol{x})$
  - k .... počet vzorů padnoucí do oblasti R
  - n .... počet vzorů celkem

- P_k ... pravděpodnost, že k z n vzorů spadne do oblasti R

$$P_k = \binom{n}{k} P^k (1-P)^{n-k}$$

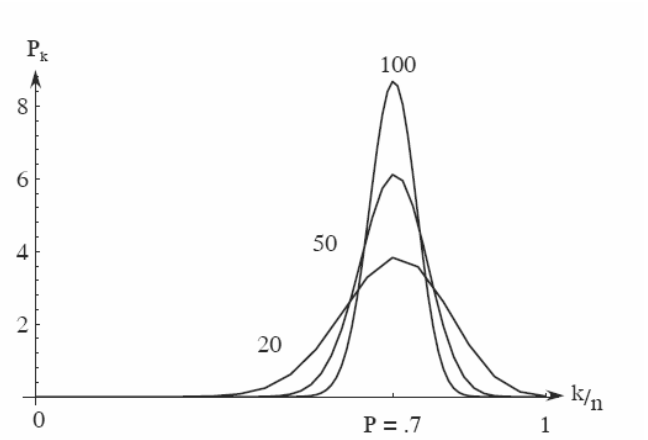
→ P_k má binomické rozdělení a střední hodnota pro k dat je

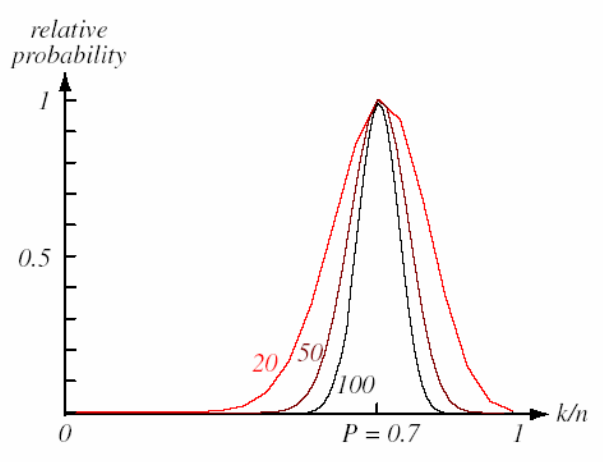
$$E(k) = n \cdot P$$

Odhady hustoty – ilustrace

- závislost P_k na hodnotě k/n
  - skutečná hodnota hustoty rozdělení, z něhož se v bodě ***x*** vybíralo

    P = 0.7
  - každá křivka je binomická





  - rostoucí *n* ... křivka s ostřejším vrcholem
  - $n \to \infty$ ... Diracova delta funkce

Odhady hustoty – myšlenka (3)

- binomické rozdělení P_k
  - velmi špičaté okolo své střední hodnoty

    $\rightarrow P \approx k/n$ (2)
  - dobrý odhad zejména pro velké n

- kombinace vztahů (1) a (2)
  - pravděpodobnost, že $\boldsymbol{x}$ padne do oblasti R

$$P = \int_R p(\mathbf{x}')d\mathbf{x}' \approx \frac{k}{n} \qquad (3)$$

$$P = \int_R p(\mathbf{x}')d\mathbf{x}'$$

$$P \approx k/n$$

Odhady hustoty – myšlenka (4)

- když oblast *R* malá a *p* se v oblasti *R* příliš nemění
  → oblast lze aproximovat obdélníkem

  – *V* ... „šířka“oblasti *R*

$$V = \int_R 1 d\mathbf{x}'$$

  – odhad pravděpodobnosti *P*

$$\mathrm{P} = \int_R \mathrm{p}(\mathbf{x}')\mathrm{d}\mathbf{x}' \approx \mathrm{p}(\mathbf{x}) \cdot \mathrm{V} \qquad (4)$$

kde ***x*** je bod v *R*

  – odhad hustoty *p(**x**)* v bodě ***x***

$$\mathrm{p}(\mathbf{x}) \approx \frac{\mathrm{k/n}}{\mathrm{V}}$$

(4) $\mathrm{P} = \int_R \mathrm{p}(\mathbf{x}')\mathrm{d}\mathbf{x}' \approx \mathrm{p}(\mathbf{x}) \cdot \mathrm{V}$

(3) $\mathrm{P} = \int_R \mathrm{p}(\mathbf{x}')\mathrm{d}\mathbf{x}' \approx \frac{\mathrm{k}}{\mathrm{n}}$

Odhady hustoty – praktický problém

- *n* fixní a fixní velikost *V*
  - *k/n* konverguje k p(x)·V

prostorově zprůměrovaný odhad $p(\mathbf{x})$

  - pro získání p(x) je třeba V → 0

$$p(\mathbf{x}) \approx \frac{k/n}{V}$$

- *n* fixní a *V → 0*

nepoužitelné odhady

  → velikost oblasti se zmenšuje „do nekonečna“
  - do oblasti nepadne žádný vzor → odhad $p(x) \approx 0$
  - do oblasti náhodou padne 1 nebo více vzorů → odhad $p(x) \approx \infty$

→ odhad vždy vyhlazen

Odhady hustoty – jak obejít problém

- teoreticky
  - nekonečně vzorů pro dané rozdělení

- prakticky
  - pousloupnosti oblastí R_1, R_2 ... různé velikosti kolem $\boldsymbol{x}$
    - každá R_i obsahuje vzor $\boldsymbol{x}$
    - R_1 pracuje s 1 vzorem
    - R_2 pracuje se 2 vzory
    - .....
  - V_n ... „šířka“ oblasti R_n
  - n-tý odhad hustoty $p(\boldsymbol{x})$ v bodě $\boldsymbol{x}$ ... $p_n(\mathbf{x}) \cong \frac{k_n / n}{V_n}$
    - k_n ... počet vzorů, které padnou do R_n

Odhady hustoty – konvergence

- konvergence $p_n(\mathbf{x}) \rightarrow p(\mathbf{x})$

$\lim_{n\to\infty} V_n = 0$

$\lim_{n\to\infty} k_n = \infty , \quad p(x) \neq 0$

$\lim_{n\to\infty} \frac{k_n}{n} = 0$

oblasti se zmenšují

z nekonečně vzorků jich nekonečně mnoho padne do oblasti R_n

ačkoliv velké množství vzorů padne do R_n → z celkového počtu vzorů n je to zanedbatelné množství

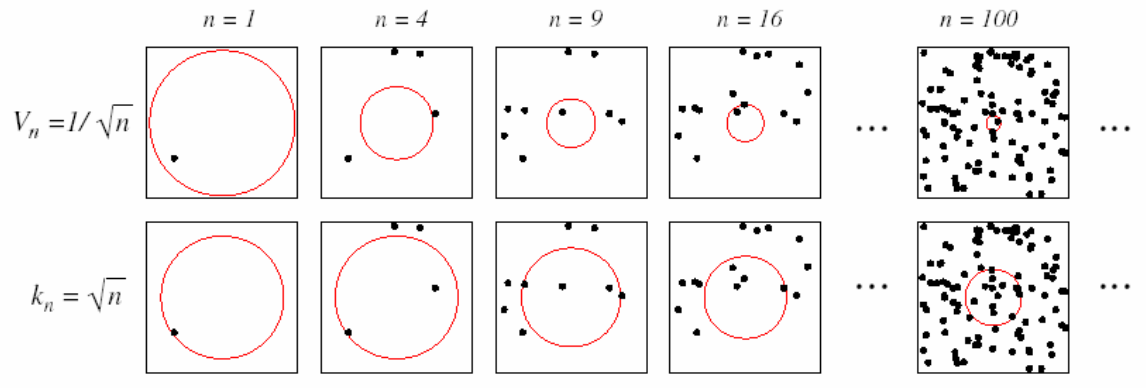
Vytvoření posloupností oblastí

- „šířka“ oblasti V_n funkcí n
  - oblast se zmenšuje určením V_n jako funkci n
    - např. $V_n = 1 / \sqrt{n}$

  → metoda: Parzenovo okénko

- počet vzorků k_n funkcí n
  - oblast je tak velká, aby obsahovala k_n sousedů vzoru $\boldsymbol{x}$
    - např. $k_n = \sqrt{n}$

  → metoda: k_n-nejbližších sousedů



Metoda Parzenova okénka

- oblast R_n ... d-dimenzionální hyperkostka
  - střed hyperkostky v bodě $\boldsymbol{x}$
  - objem hyperkostky se stranou h_n

    $V_n = h_n^d$

  → počet vzorů, které padnou do hyperkostky, se definuje pomocí okénka

- okénko ... jednotková hyperkostka se středem v počátku

$$\varphi(\mathbf{u}) = \begin{cases} 1 & |u_j| \le 1/2 \quad \text{pro } j = 1,\ldots,d \\ 0 & \text{jinak} \end{cases}$$

vzor ***u*** padne do okénka

vzor ***u*** nepadne do okénka

- okénko ... hyperkostka o objemu V_n vycentrovaná v $\boldsymbol{x}$

$$\varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right) = \begin{cases} 1 & \text{když vzor } \mathbf{x}_i \text{ padne do hyperkostky} \\ & \text{o objemu } V_n \text{ vycentrované v } \mathbf{x} \\ 0 & \text{jinak} \end{cases}$$

Metoda Parzenova okénka

- počet vzorů k_n v hyperkostce

$$k_n = \sum_{i=1}^{n} \varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

$$p_n(x) \cong \frac{k_n/n}{V_n}$$

- *n*-tý odhad hustoty $p_n(\boldsymbol{x})$

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{V_n}\varphi\left(\frac{\mathbf{x} - \mathbf{x}_i}{h_n}\right)$$

  - spojitá funkce $p(\boldsymbol{x})$ → aproximace nespojitou skokovou funkcí $\varphi(\boldsymbol{u})$
    → odhad zatížen chybou

- Parzen: odhad pomocí hladké funkce $\varphi(\boldsymbol{u})$
  - okénko $\varphi(\boldsymbol{u})$

    $\varphi(\mathbf{u}) \geq 0$ a $\int \varphi(\mathbf{u})d\mathbf{u} = 1$
  - pro takové okénko $\varphi(\boldsymbol{u})$ → odhad $p_n(\boldsymbol{x})$ je funkcí hustoty

Parzenovo okénko – vliv h_n

- vliv šířky okénka h_n na odhad $p_n(\boldsymbol{x})$
  - funkce $\delta_n(\boldsymbol{x})$ .... Diracův puls

$$\delta_n(\mathbf{x}) = \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}}{h_n}\right)$$

  - $p_n(\boldsymbol{x})$ pomocí $\delta_n(\boldsymbol{x})$

$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\delta_n(\mathbf{x}-\mathbf{x}_i)$$

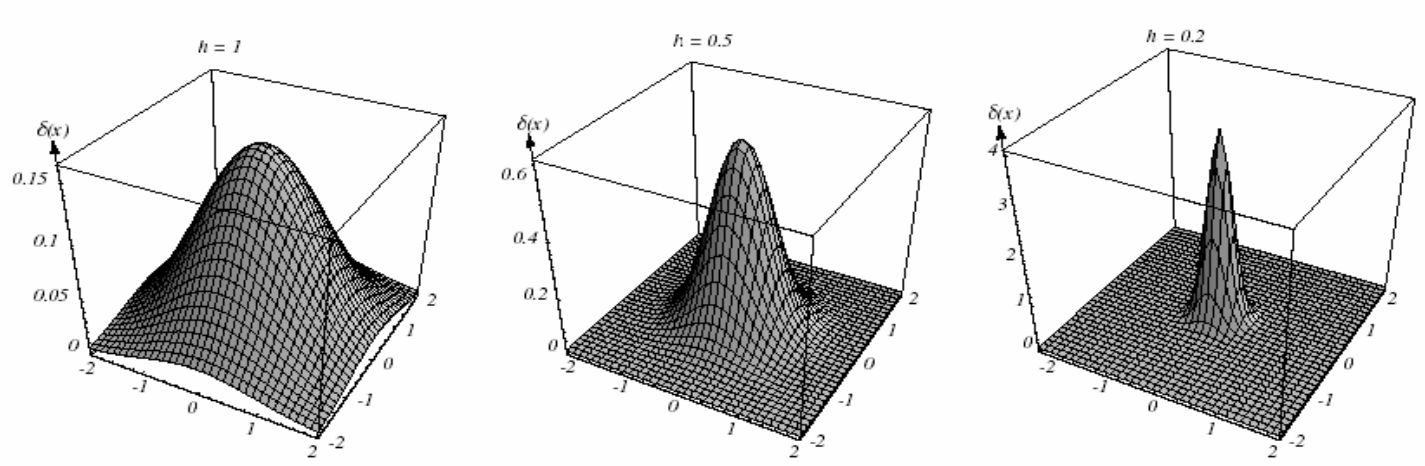
$$p_n(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)$$

- Diracův puls δ_n
  - δ_n má konstatní objem
  - $V_n = h_n^d$ → h_n má vliv na amplitudu (výšku) i šířku funkce $\delta_n(\mathbf{x})$

$$\int \delta_n(\mathbf{x})d\mathbf{x} = \int \frac{1}{V_n}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h_n}\right)d\mathbf{x} = \int \varphi(\mathbf{u})d\mathbf{u} = 1$$

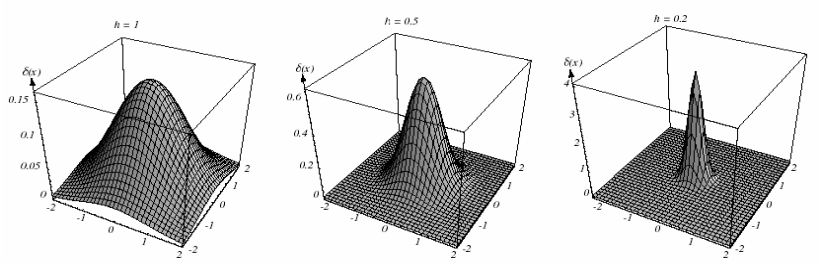
Parzenovo okénko – příklady

- příklad (*dim=2*)
  - Parzenovo okénko ... 2D-rotační symetrická normální funkce
  - δ_n .... $h_n=1$ a $h_n=0{,}5$ a $h_n=0{,}2$



$\delta_n(\boldsymbol{x})$ normalizována → na svislé ose různá škálování

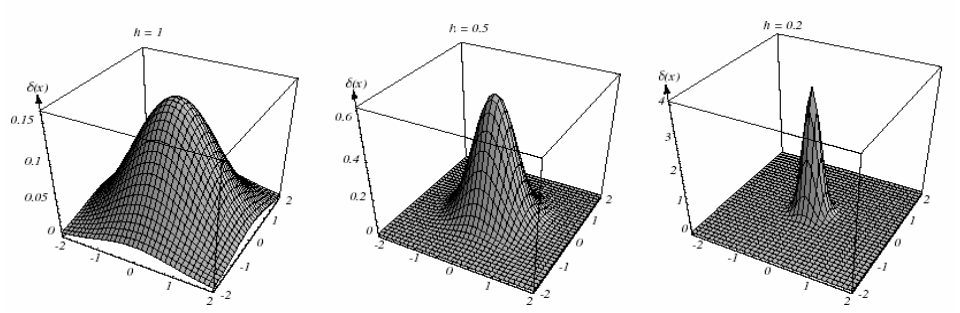
Vlastnosti Parzenova okénka



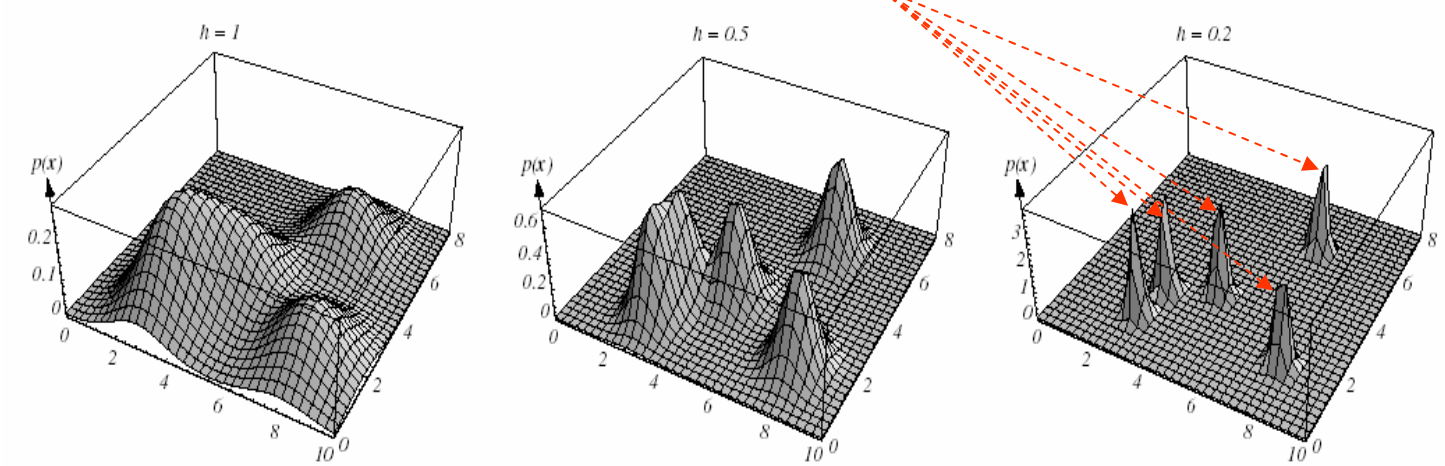
- volba šířky h_n
  - h_n velmi velké
    - malá amplituda δ_n (nízký vrchol)
    - $\boldsymbol{x}$ daleko od $\boldsymbol{x_i}$ aby se změnilo $\delta_n(\boldsymbol{x}-\boldsymbol{x_i})$ od $\delta_n(\boldsymbol{0})$
  - h_n velmi malé
    - vysoký vrchol $\delta_n(\boldsymbol{x}-\boldsymbol{x_i})$
    - vrchol blízko $\boldsymbol{x}=\boldsymbol{x_i}$
  - h_n dosáhne 0
    - $\delta_n(\boldsymbol{x}-\boldsymbol{x_i})$ Diracova delta funkce vycentrovaná v $\boldsymbol{x_i}$

Parzenovo okénko – příklady

- Parzenova okénka – tvar δ_n



- odhad $p_n(\boldsymbol{x})$ pomocí 5 trénovacích vzorů



na svislé ose různá škálování

Ilustrační příklad 1 (dim=1)

- normální rozdělení $p(\boldsymbol{x}) \sim N(0,1)$
  - okénko ... Gaussova funkce

$$\varphi(u) = \frac{1}{\sqrt{2\pi}} e^{\frac{-u^2}{2}}$$

  - „šířka“ kostky h_n

$$h_n = \frac{h_1}{\sqrt{n}}$$

h_1 ... parametr ovlivňující šířku okénka (uživatelsky definovaný)

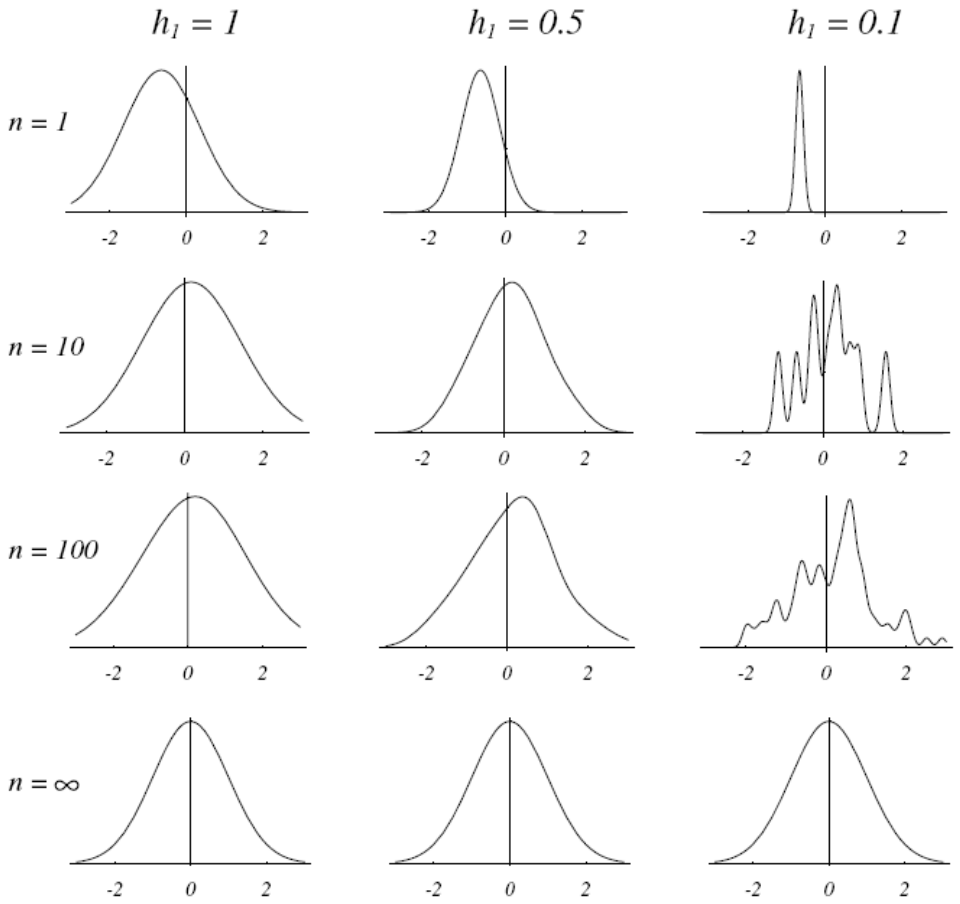
→ $p_n(\boldsymbol{x})$

$$p_n(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{h_n} \varphi\left(\frac{x - x_i}{h_n} \right)$$

$d=1 \rightarrow V_n = h_n$

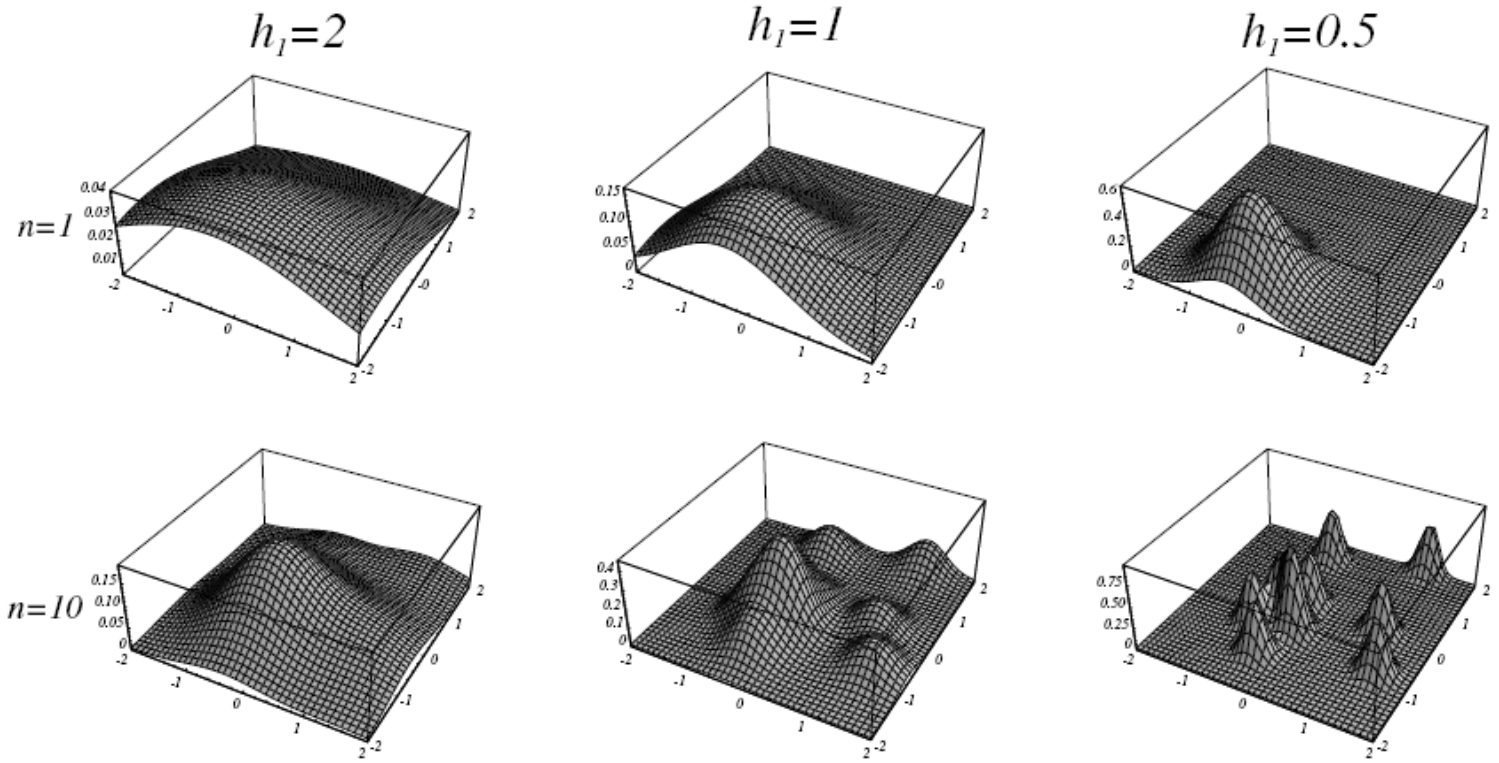
Ilustrační příklad 1 – výsledky

- odhad neznámé $p(\boldsymbol{x}) \sim N(0,1)$
  - iniciální šířka okénka h_1 a počet vzorů n



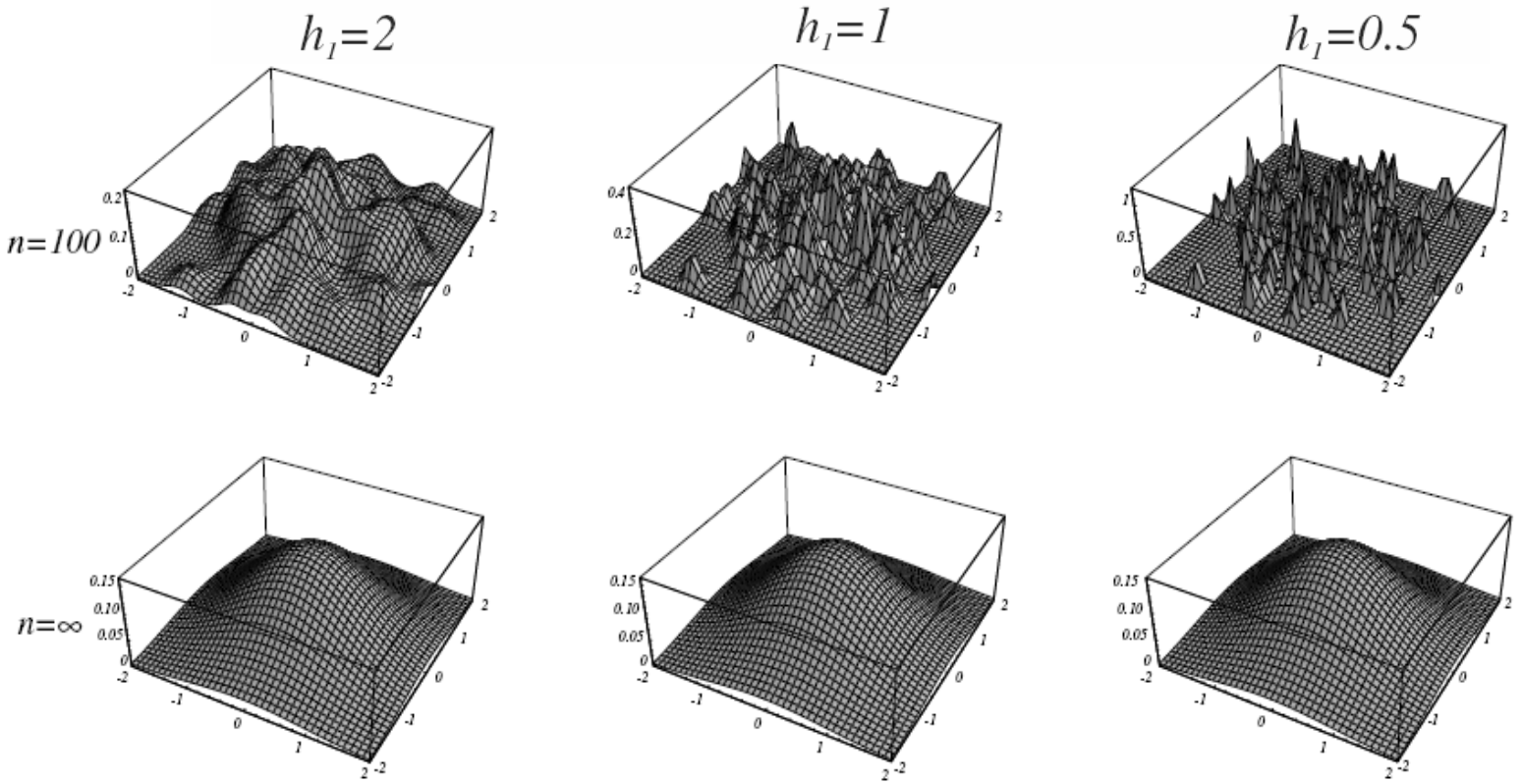
Ilustrační příklad 2 (dim=2)

- odhad *p(**x**)* ... dvourozměrné normální rozdělení
  - iniciální šířky okénka h_1 a počet vzorů *n* (*n=1 a n=10*)



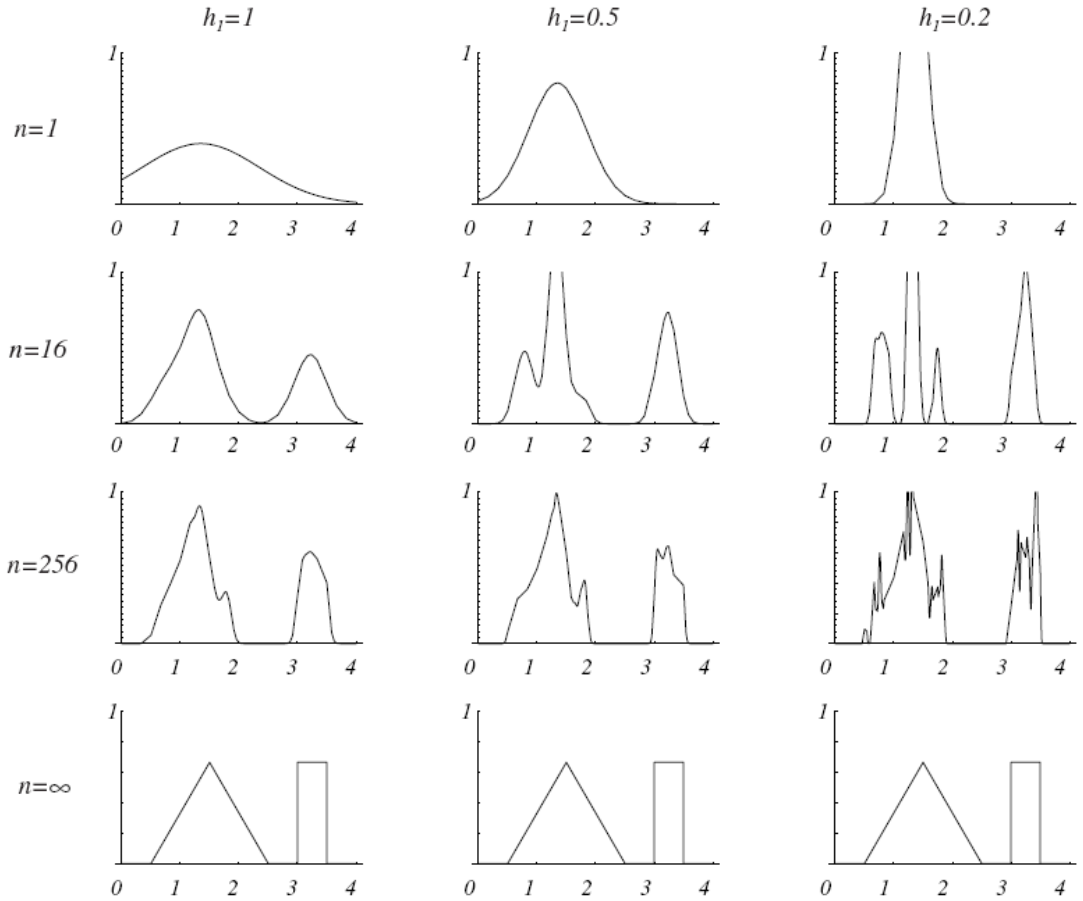
Ilustrační příklad 2 (dim=2)

- odhad *p(**x**)* ... dvourozměrné normální rozdělení
  - iniciální šířky okénka h_1 a počet vzorů *n* (*n=100 a n=∞*)



Ilustrační příklad 3 (dim=1)

- neznámá hustota: směs uniformní a trojúhelníkové hustoty



Klasifikace – Parzenovo okénko

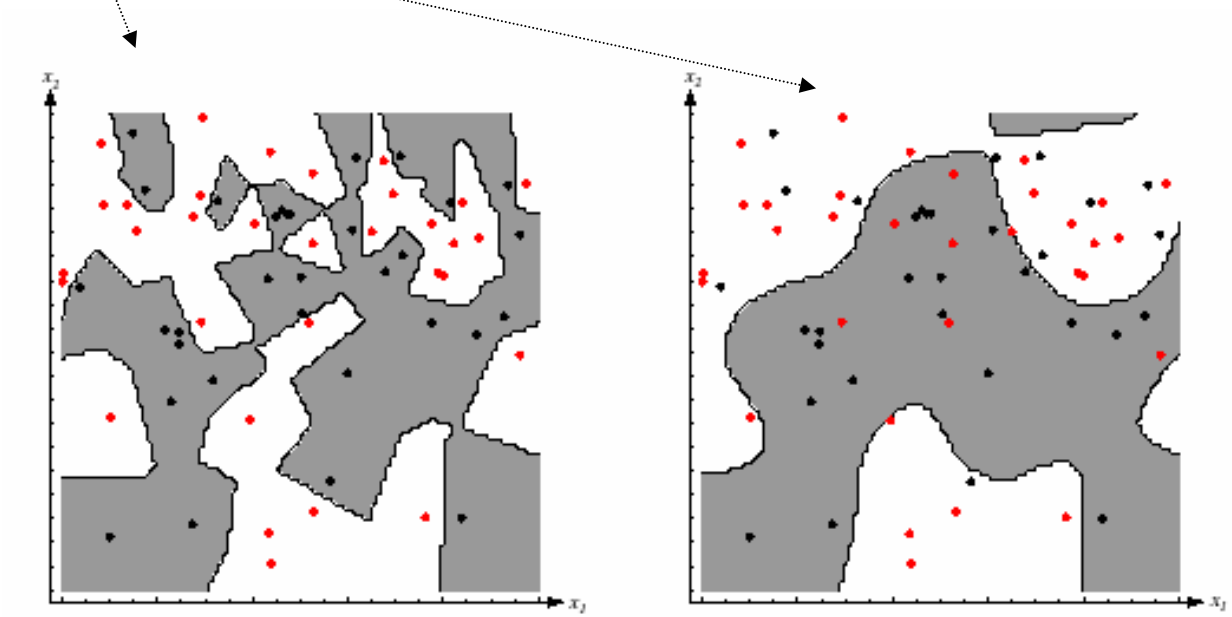
- klasifikátor založený na Parzenově okénku
  - 1. předložení vzoru ***x***
  - 2. vypočten odhad hustoty v bodě ***x*** pro každou třídu
  - 3. klasifikace vzoru ***x*** podle minimální ztráty

- klasifikátor pro 2 třídy
  - ***x*** do ω_1 (resp. ω_2) když

$$\frac{\frac{1}{N_1}\sum_{i=1}^{N_1}\frac{1}{h}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)}{\frac{1}{N_2}\sum_{i=1}^{N_2}\frac{1}{h}\varphi\left(\frac{\mathbf{x}-\mathbf{x}_i}{h}\right)} > (<)\frac{P(\omega_2)}{P(\omega_1)}\frac{\lambda_{21}-\lambda_{22}}{\lambda_{12}-\lambda_{11}}$$

Klasifikace – rozhodovací hranice

- rozhodovací hranice .... vliv okénkové funkce φ a šířky h
  - h malé → komplikované oblasti
  - h velké → jednodušší oblasti



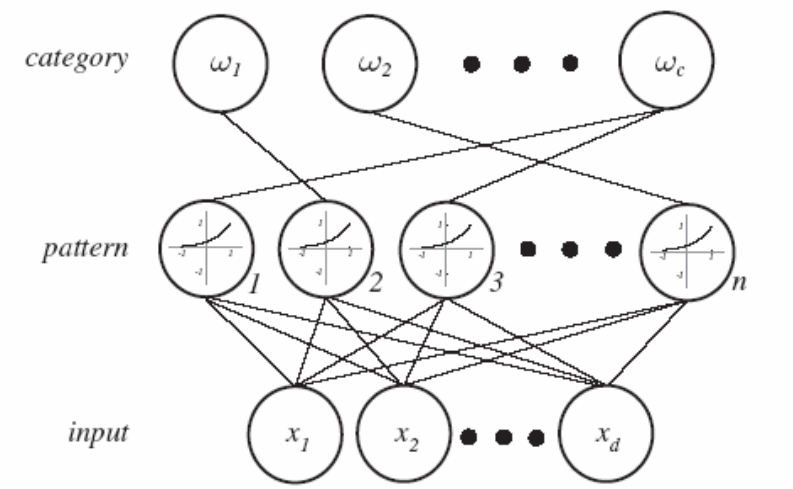
Parzenovo okénko – závěr

- klasifikátor
  - okénko úzké → malá chyba na trénovacích vzorech → problém přeučení

- metoda Parzenova okénka
  - obecné (+)
  - velké množství trénovacích dat (–)
  - časová i prostorová náročnost (–)
    - počet trénovacích dat .... roste exponenciálně s dimenzí dat
      → „ prokletí dimenze“
    - mnohodimenzionální funkce typicky mnohonásobně složitější než nízko-dimenzionální funkce

Pravděpodobnostní neuronové sítě (PNS)

- paralelní implementace Parzenova okénka
  - neuronové sítě (NS)
- vstup
  - *n* *d*-dimenzionálních vzorů v *c* třídách

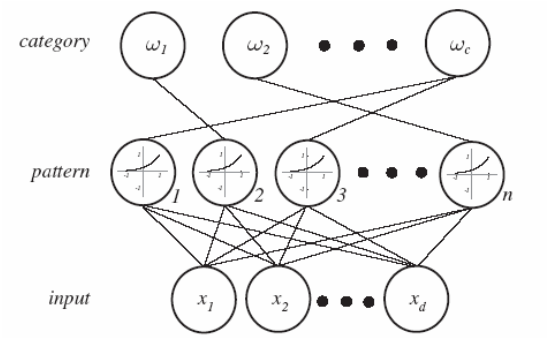


- *d* vstupních neuronů
- *n* skrytých neuronů
- *c* výstupních neuronů

- vstupní neuron → spojen s každým skrytým neuronem
- skrytý neuron → spojen s právě jediným výstupním neuronem

Pravděpodobnostní neuronové sítě (PNS)

- vstupní vrstva ↔ skrytá vrstva
  - váhy, které je nutné naučit



- skrytý neuron
  - skalární součin váhového vektoru ***w*** a normalizovaného předloženého vzoru ***x***

$$\text{net} = \mathbf{w}^T\mathbf{x}$$

  - skalární součin → aplikována funkce

$$\exp\left[(\text{net}-1)/\sigma^2\right]$$

    - σ ... uživatelsky definovaný parametr

- výstupní neuron
  - součet výstupů ze skrytých neuronů, se kterými je spojen

PNS – učení

- 1. normalizace trénovacích vzorů

$$\sum_{i=1}^{d} x_i^2 = 1$$

- 2. předložení trénovacího vzoru $\boldsymbol{x}_k$
  - I) aktualizace vah: vstupní neurony ↔ *k*-tý skrytý neuron

    $\boldsymbol{w}_k = \boldsymbol{x}_k$
  - II) vytvoření vazby: *k*-tý skrytý neuron ↔ výstupní neuron *i* pro třídu ω_i vzoru $\boldsymbol{x}_k$

    $a_{ki} = 1$

- 3. proces zopakován pro všechny trénovací vzory

PNS – algoritmus učení

- značení
  - $\boldsymbol{x}_j = (x_{j1}, \ldots, x_{jn})$ $j=1,\ldots,n$
  - $\boldsymbol{w}_j = (w_{j1}, \ldots, w_{jn})$ $j=1,\ldots,n$

- algoritmus

```
1. begin
2.    initialize j ← 0, n, a_ij ← 0      j=1,...,n  i=1,...,c
3.    do j ← j+1
4.       x_jk ← x_jk / (Σx_ji^2)^1/2      // normalizace
5.       w_jk ← x_jk                      // učení
6.       if (x in ω_c) then a_jc ← 1
7.    until j=n
8. end
```

PNS – klasifikace

- PNN klasifikace
  - 1. normalizace testovaného vzoru ***x***
  - 2. předložení vzoru ***x*** vstupní vrstvě

  - 3a. každý skrytý neuron *k* vypočte skalární součin net_k

    $$net_k = \boldsymbol{w}_k^T \cdot \boldsymbol{x}$$

  - 3b. na net_k aplikována aktivační funkce

    $$\exp\left(\left(\text{net}_k - 1\right)/\sigma^2\right)$$

    σ ... uživatelsky definovaný parametr → šířka Gaussova okénka

  - 4. každý výstupní neuron sečte příspěvky od skrytých neuronů, se kterými je spojen

PNS – klasifikace

- aktivační funkce → exponenciála
  - odvození

Parzenovo okénko
= příslušný Gaussián

$$\varphi\left(\frac{\mathbf{x}-\mathbf{w}_k}{h_n}\right) \propto e^{\frac{-(\mathbf{x}-\mathbf{w}_k)^T(\mathbf{x}-\mathbf{w}_k)}{2\sigma^2}} = e^{\frac{-(\mathbf{x}^T\mathbf{x}+\mathbf{w}_k{}^T\mathbf{w}_k-2\mathbf{x}^T\mathbf{w}_k)}{2\sigma^2}} = e^{\frac{\mathrm{net}_k-1}{\sigma^2}}$$

$\boldsymbol{w}_k{}^T \cdot \boldsymbol{w}_k = \boldsymbol{x}^T \cdot \boldsymbol{x} = 1$, $h_n(\sigma)$ konstanta

- výstup skrytého neuronu
  - pravděpodobnost, že vzor $\boldsymbol{x}$ vygenerován Gaussovkou umístěnou ve středu trénovacího vzoru

- výstupní neuron
  - součet lokálních odhadů skrytých neuronů → diskriminační funkce $g_i(\boldsymbol{x})$
  - $max\{g_i(\boldsymbol{x})\}$ → hledaná třída pro vzor $\boldsymbol{x}$

PNS – algoritmus klasifikace

- algoritmus

```
1. begin
2.    initialize k ← 0, x ← test_pattern
3.    do k ← k+1
4.       net_k ← w_k^T · x
5.       if (a_ki=1) then g_i ← g_i+exp((net_k-1)/σ^2)
6.    until k=n
7.    return class ← argmax_i(g_i(x))
8. end
```

- vlastnosti PNS
  - rychlost učení
    - jeden průchod trénovací množinou
  - prostorová složitost
    - úměrná počtu vazeb $O((n+1)d)$
  - lze použít online učení
    - nový vzor lze snadno zabudovat do již naučeného modelu

Metoda k_n-nejbližších sousedů

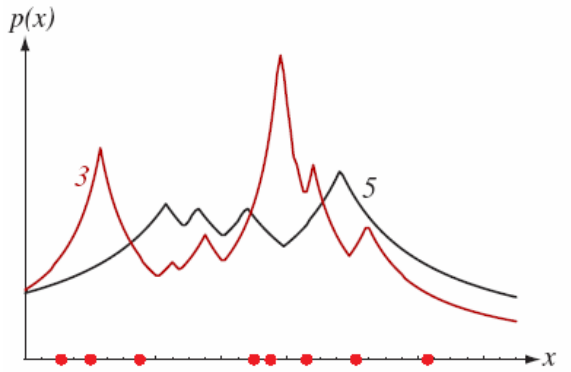
- odhad hustoty $p_n(\boldsymbol{x})$ z n trénovacích vzorů
  - okénko okolo $\boldsymbol{x}$
  - okénko se zvětšuje, dokud v něm není k_n vzorů
    → k_n-nejbližších sousedů k bodu $\boldsymbol{x}$

- velikost okénka → závislá na trénovacích datech
  - velká hustota okolo $\boldsymbol{x}$ → malé okénko → dobré rozlišení
  - malá hustota okolo $\boldsymbol{x}$ → široké okénko (zastaví se, až narazí na nějakou oblast s velkou hustotou)

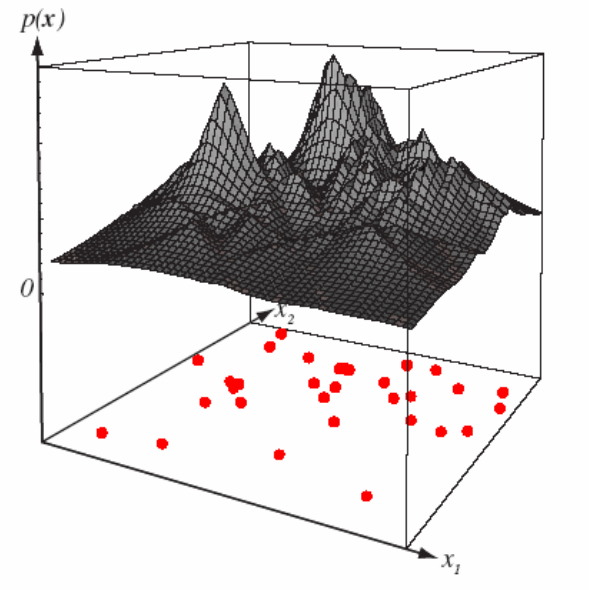
- odhad hustoty $p_n(\mathbf{x}) \cong \frac{k_n / n}{V_n}$

k_n-nejbližších sousedů – postřehy

- postřehy
  - $p_n(\boldsymbol{x})$ spojitá → gradient není spojitý
  - body nespojitosti gradientu → nemusí být na pozici trénovacích vzorů



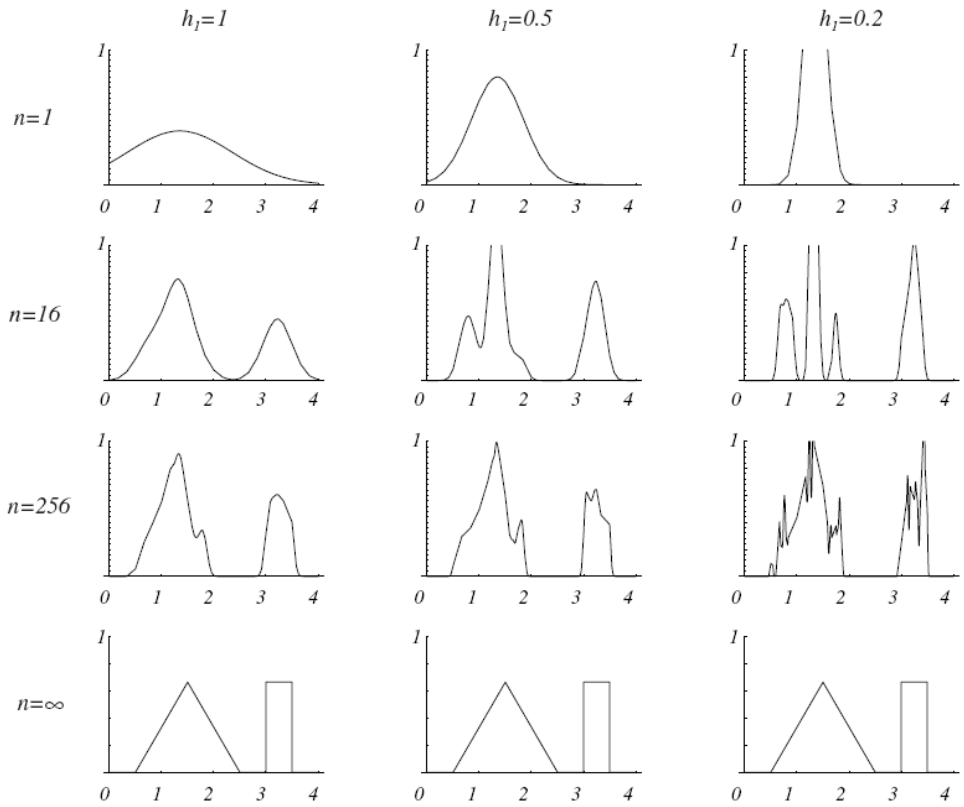
odhad hustoty pomocí 3- a 5- nejbližších sousedů
(dim=1)



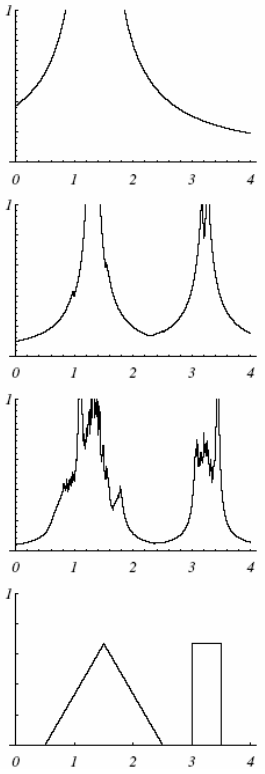
odhad hustoty pomocí 5-nejbližších sousedů
(dim=2)

k_n-nejbližších sousedů a Parzenovo okénko

- porovnání metod odhadu
  - uniformní a trojúhelníková hustota pro *n = 1, 16, 256, ∞*



Parzenovo okénko



k_n–nejbližších sousedů

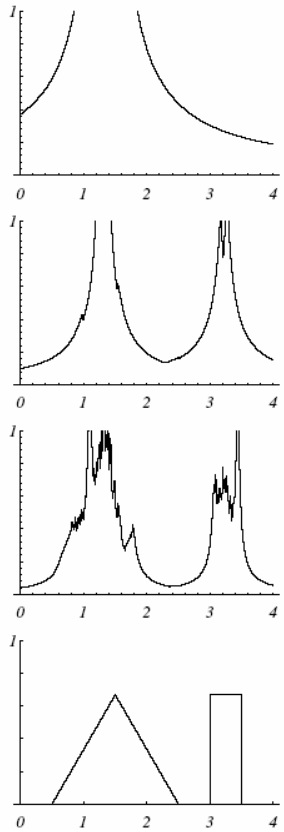
$k_n=\sqrt{n}$

k_n-nejbližších sousedů – postřehy

- *n=1 a* $k_n=\sqrt{n}=1$
  - odhad hustoty $p_n(x) = 1/(2|x-x_1|)$
    → špatný odhad

- rostoucí *n*
  - odhad se zlepšuje ale $\int p_n(x) = \infty$ (ale má být 1)

  - výhoda: $p_n(x)$ se nikdy nepřiblíží k 0
  - použití: mnohorozměrné prostory



k_n-nejbližších sousedů – klasifikace

- klasifikace do 2 tříd ω_1 a ω_2
  - 1. předložení vzoru $\boldsymbol{x}$
  - 2. N_1 a N_2 ... počet trénovacích vzorů pro ω_1 a. ω_2 $(N = N_1 + N_2)$
  - 3. nalezení poloměrů (hyper)koulí
    - r_1 ... poloměr (hyper)koule se středem v $\boldsymbol{x}$, která obsahuje k_n trénovacích bodů z ω_1
    - r_2 ... poloměr (hyper)koule se středem v $\boldsymbol{x}$, která obsahuje k_n trénovacích bodů z ω_2
  - 4. spočtení objemů V_1 a V_2 (hyper)koulí
  - 5. klasifikace vzoru $\boldsymbol{x}$ do ω_1 (ω_2)

$$\frac{k_n}{k_n}\frac{N_2}{N_1}\frac{V_2}{V_1} > (<) \frac{P(\omega_2)}{P(\omega_1)}\frac{\lambda_{21}-\lambda_{22}}{\lambda_{12}-\lambda_{11}}$$

$$\frac{V_2}{V_1} > (<) \frac{N_1}{N_2}\frac{P(\omega_2)}{P(\omega_1)}\frac{\lambda_{21}-\lambda_{22}}{\lambda_{12}-\lambda_{11}}$$

- poznámky
  - hodnota k_n ... obecně různá pro jednotlivé třídy
  - hledání nejbližších sousedů
    - Eukleidovská vzdálenost → (hyper)koule
    - Mahalanobisova vzdálenost → (hyper)elipsoidy

Odhad apost. pravděpodobnosti

- odhad aposteriorní pravděpodobnosti $P(\omega_i|\boldsymbol{x})$
  - vstup
    - n trénovacích vzorů
  - umístění okénka
    - okénko o velikosti V okolo $\boldsymbol{x}$, aby mělo k trénovacích vzorů
  - odhad $P_n(\omega_i|\boldsymbol{x})$
    - k_i z k trénovacích vzorů patří do ω_i → odhad $P_n(\omega_i|\boldsymbol{x}) = k_i / k$
  - klasifikace
    - třída, která je nejvíce reprezentována v daném okénku

→ dostatek trénovacích vzorů + dostatečně malé okénko → dobré výsledky

Metody odhadu hustoty – shrnutí

- metody na odhad hustoty

  - Parzenovo okénko
    - definuje se, jak se má zvětšovat objem V_n okénka
    - V_n je funkcí n

  např. $V_n=1/\sqrt{n}$

  - k_n-nejbližších sousedů
    - definuje se, kolik vzorů k_n má okénko obsahovat → V_n se zvětšuje, dokud nemá k_n vzorů
    - k_n je funkcí n

  např. $k_n=\sqrt{n}$

Pravidlo *k*-nejbližších sousedů

Pravidlo nejbližšího souseda

- pravidlo nejbližšího souseda
  - vstup
    - $D^n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ ... „označkované prototypy“
    - $\boldsymbol{x}$ ... neznámý vzor → klasifikován
  - výstup
    - $\boldsymbol{x}'$ ... prototyp z D^n nejblížší k $\boldsymbol{x}$ → klasifikace $\boldsymbol{x}$ do třídy, kam patří $\boldsymbol{x}'$

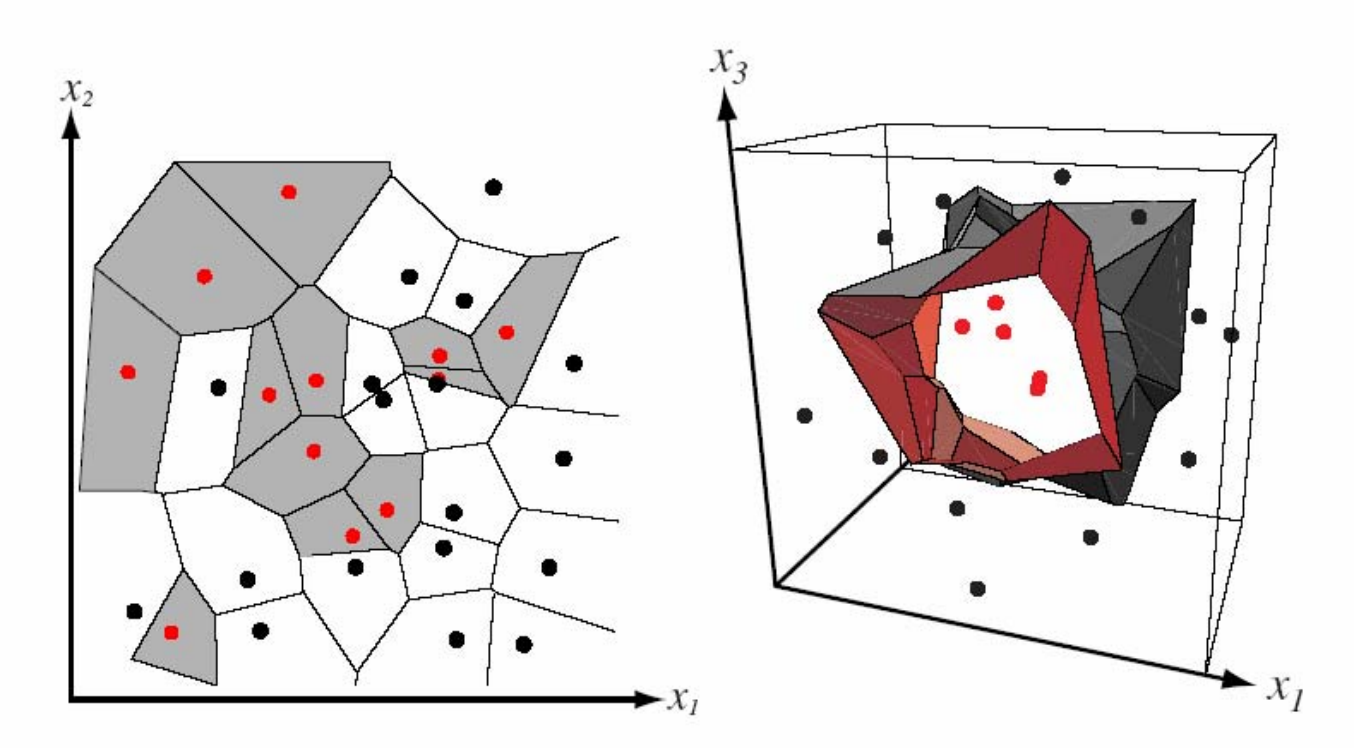
- vlastnosti
  - suboptimální
  - typicky
    - větší chyba pravděpodobnosti než minimální možná chyba
  - lze ukázat
    - pro $n \to \infty$: pravděpodobnost_chyby $\leq$ 2 * Bayesovská_pravděpodobnost_chyby

Pravidlo nejbližšího souseda – proč funguje?

- pravidlo nejbližšího souseda
  - θ' ... značka třídy, kam patří prototyp $\boldsymbol{x'}$
  - pravděpodobnost $\theta'=\omega_i$
    → aposteriorní pravděpodobnost $P(\omega_i|\boldsymbol{x'})$
  - velký počet vzorů
    - lze předpokládat, že $\boldsymbol{x'}$ dostatečně blízko k $\boldsymbol{x}$
      → $P(\omega_i|\boldsymbol{x'}) \approx P(\omega_i|\boldsymbol{x})$

- rozdělení příznakového prostoru do oblastí
  - oblast $R_i = \{ \boldsymbol{x} ; d(\boldsymbol{x},\boldsymbol{x}_i) \leq d(\boldsymbol{x},\boldsymbol{x}_k), i \neq k \}$
  - body v oblasti označkovány třídou „jejich“ trénovacího vzoru
    → Voronoiova mozaika

Pravidlo nejbližšího souseda – Voronoiova mozaika



n-úhelníkové oblasti

3D-krystaly

Pravidlo nejbližšího souseda – vlastnosti 1

- vlastnosti pravidla
  - ω_i ... nejpravděpodobnější třída $\boldsymbol{x}$

$$P(\omega_i \mid \mathbf{x}) = \max_k P(\omega_k \mid \mathbf{x})$$

  - *$P(\omega_i|\boldsymbol{x}) \sim 1$*
    - pravidlo nejbližšího souseda ≈ Bayesovský výběr
    - minimum chybné pravděpodobnosti malé → pravděpodobnost chyby při metodě nejbližšího souseda také malá
  - *$P(\omega_i|\boldsymbol{x}) \sim 1/c$*
    - pravidlo nejbližšího souseda ≠ Bayesovský výběr
    - pravděpodobnosti chyby přibližně stejná u obou metod

Pravidlo nejbližšího souseda – vlastnosti 2

- $n \to \infty$
  - značení

    $$P = \lim_{n \to \infty} P_n$$

    průměrná pravděpodobnost chyby na *n* trénovacích vzorech
  - lze dokázat

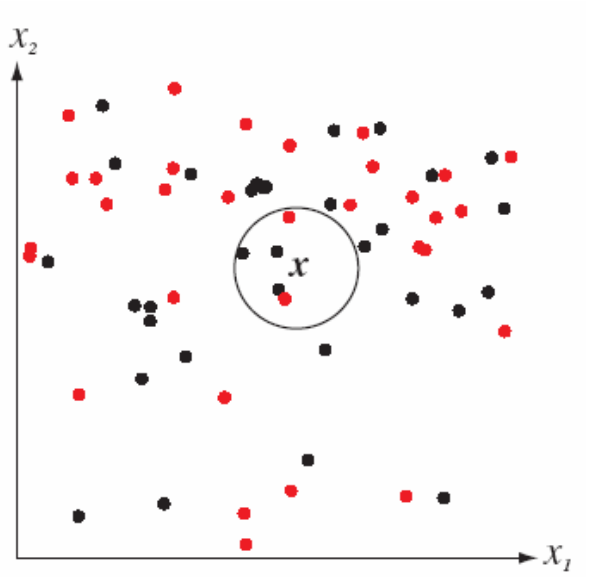
    $$P^* \le P \le P^*\left(2 - \frac{c}{c-1}P^*\right) \le 2P^*$$

    Bayesovská chyba

    počet tříd
- $n < \infty$
  - obecný případ → není znám žádný vztah
    - konvergence může být libovolně pomalá
    - chyba P_n se nemusí monotónně snižovat
  - obecně nelze analyticky spočítat
    - nutné další předpoklady o pravděpodobnostní struktuře problému

Pravidlo *k*-nejbližších sousedů

- zobecnění metody
  - vzor ***x*** klasifikován do třídy nejvíce zastoupené mezi *k* nejbližšími trénovacími vzory
    - nalezení *k* nejbližších sousedů (trénovacích vzorů)
    - zjištění tříd nejbližších sousedů
    - vítězství třídy s „největším počtem hlasů“



k=5

Zlepšení výpočetní složitosti nejbližšího souseda

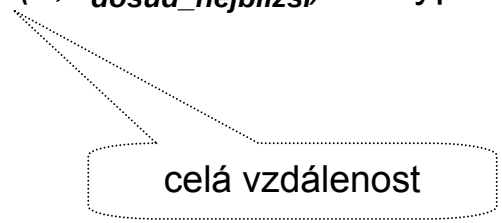
- literatura
  - mnoho analýz ohledně výpočetní složitosti (pro *dim=1* a *dim=2*)

- 2 základní techniky
  - 1. metoda částečné vzdálenosti
  - 2. eliminace zbytečných prototypů

Metoda částečné vzdálenosti – 1

- vstup
  - *n* trénovacích vzorů (prototypů) dimenze *d*
  - konstanta *r (r<d)*

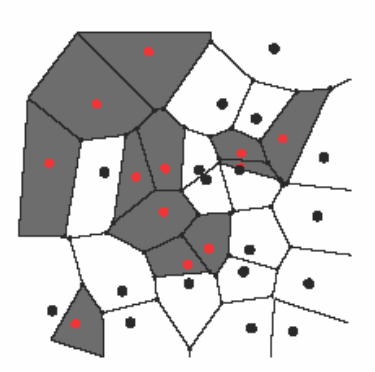
- výpočet
  - částečná vzdálenost $d_r(\boldsymbol{x},\boldsymbol{x'})$ .... prvních *r* příznaků
  - $d_r(\boldsymbol{x},\boldsymbol{x'}) > d(\boldsymbol{x},\boldsymbol{x}_{dosud\_nejbližší})$ → výpočet končí



Eliminace zbytečných prototypů – 2

- metoda
  - eliminace prototypů, v jejichž okolí jsou jen prototypy ze stejné třídy

  → rozhodovací hranice i celková chyba se nezmění



- algoritmus

```
1. begin
2.   inicializace j ← 0, D ← trénovací_data, n ← počet_prototypů
3.   vytvoření úplného Voronoiova diagramu z D
4.   do j ← j+1
5.     nalezení Voronoiovy sousedy pro prototyp xj'
6.     if (některý soused z jiné třídy než xj') then označení xj'
7.   until j=n
8.   odstranění všech neoznačených prototypů
9.   vytvoření Voronoiova diagramu ze zbylých (označených) prototypů
10.end
```

Eliminace zbytečných prototypů – 2

- prototyp zůstává
  - přispívá-li k rozhodovací hranici → aspoň jeden z jeho sousedů patří k jiné třídě

- vlastnosti
  - negarantuje minimální množinu prototypů (–)
  - sníží výpočetní složitost (+)
    - bez změny přesnosti výpočtu
  - nelze dodatečně přidávat prototypy do „vyčištěného“ modelu (–)
    - k vyčištení je potřeba znalost všech trénovacích dat

Metriky pro metody nejbližších sousedů – 1

- metrika
  - nezápornost: $D(\mathbf{a},\mathbf{b}) \geq 0$
  - reflexivita: $D(\mathbf{a},\mathbf{b}) = 0$ jen když $\mathbf{a} = \mathbf{b}$
  - symetrie: $D(\mathbf{a},\mathbf{b}) = D(\mathbf{b},\mathbf{a})$
  - trojúhelníková nerovnost: $D(\mathbf{a},\mathbf{b}) + D(\mathbf{b},\mathbf{c}) \geq D(\mathbf{a},\mathbf{c})$

- nejběžnější metriky
  - Eukleidovská metrika

$$D(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} (a_k - b_k)^2 \right)^{1/2}$$

  - Minkowského metrika

$$L_k(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} |a_k - b_k|^k \right)^{1/k}$$

Metriky pro metody nejbližších sousedů – 2

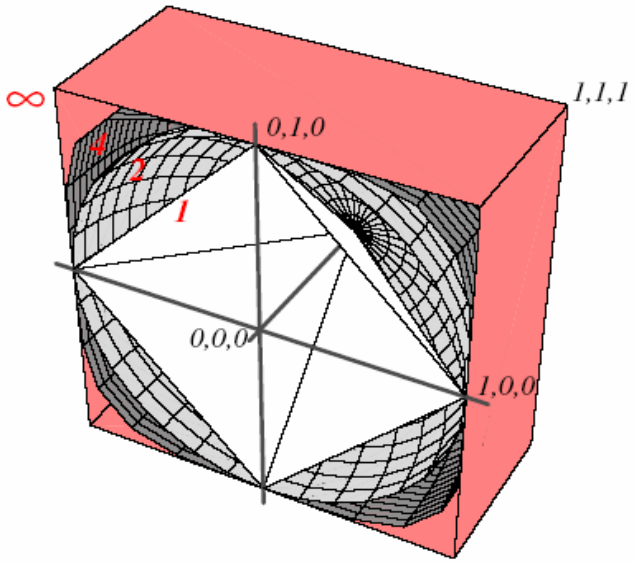
- Minkowského metrika $L_k(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} |a_k - b_k|^k \right)^{1/k}$
  - *k=1* ... Manhatannovská vzdálenost
  - *k=2* ... Eukleidovská vzdálenost
  - *k*=∞ ... maximum z projekcí bodů na jednotlivé souřadnicové osy

  - jak vypadá množina bodů ve vzdálenosti 1 od počátku pro *k=1, k=2, k*=∞ ?

Metriky pro metody nejbližších sousedů – 2

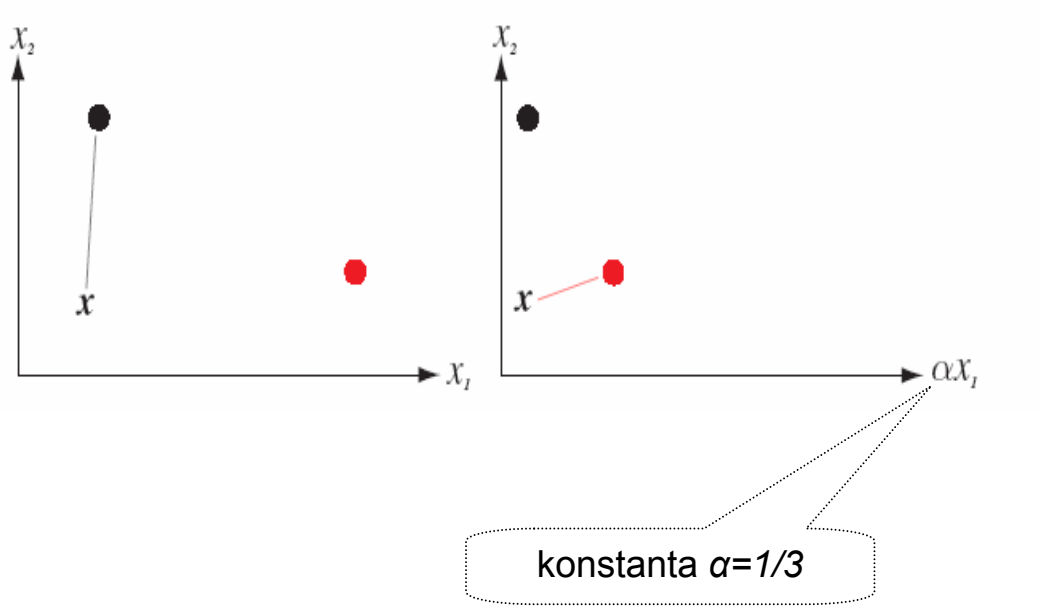
- Minkowského metrika $L_k(\mathbf{a},\mathbf{b}) = \left(\sum_{k=1}^{d} |a_k - b_k|^k \right)^{1/k}$
  - *k=1* ... Manhatannovská vzdálenost
  - *k=2* ... Eukleidovská vzdálenost
  - *k*=∞ ... maximum z projekcí bodů na jednotlivé souřadnicové osy

  - jak vypadá množina bodů ve vzdálenosti 1 od počátku pro *k=1, k=2, k*=∞ ?



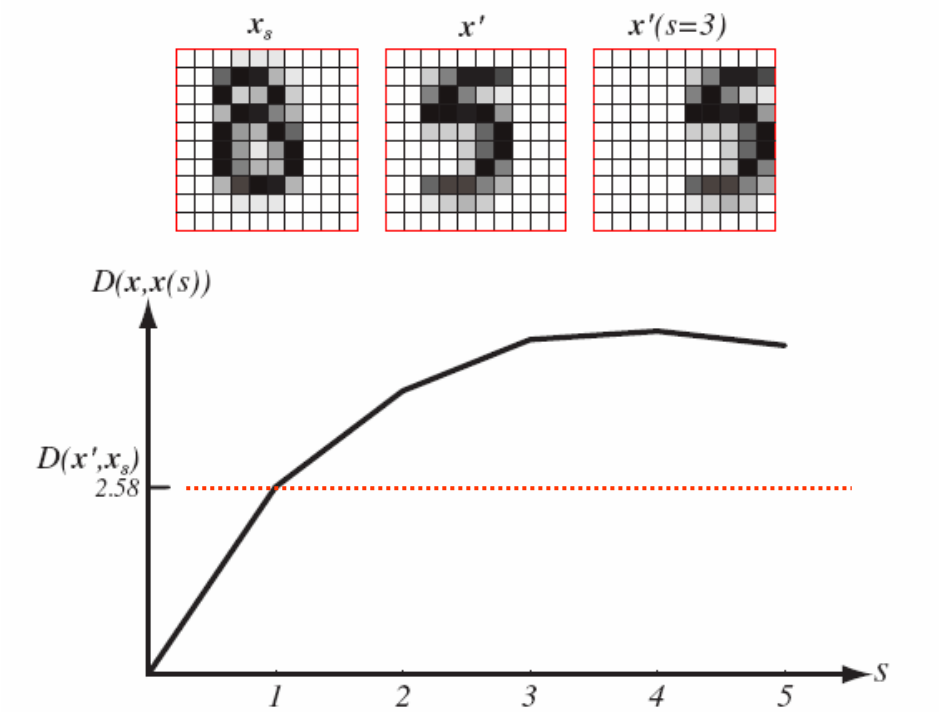
Metriky pro metody nejbližších sousedů – 3

- ne vždy se hodí Eukleidovská metrika
  - např. vynásobení každé souřadnicové osy konstantou
    → odlišné Eukleidovské vzdálenosti v transformovaném prostoru
    → vliv na nalezení nejbližšího souseda



Metriky pro metody nejbližších sousedů – 4

- Eukleidovská metrika – invariance vůči posunutí



s > 1 → $L_2(\mathbf{x'},\mathbf{x'}(s)) > L_2(\mathbf{x'},\mathbf{x}_8)$ → chybná klasifikace

Tangentová metrika – konstrukce klasifikátoru

- tangentová metrika
  - výpočetně náročná (–)
  - invariantní vůči základním transformacím (+)
  - *r* transformací
    - např. horizontální a verikální posun, rotace, škálování, čárové ztenčení, ...

- konstrukce klasifikátoru
  - pro každý prototyp ***x'***
    1. aplikace jednotlivých transformací $F_i(\mathbf{x}', \alpha_i)$
    2. vytvoření tangentového vektoru $\mathbf{TV}_i$ pro každou transformaci i

       $$\mathbf{TV}_i = F_i(\mathbf{x}', \alpha_i) - \mathbf{x}'$$
    3. uspořádání tangentových vektorů $\mathbf{TV}_i$ do matice $\mathbf{T}(\mathbf{x}')$

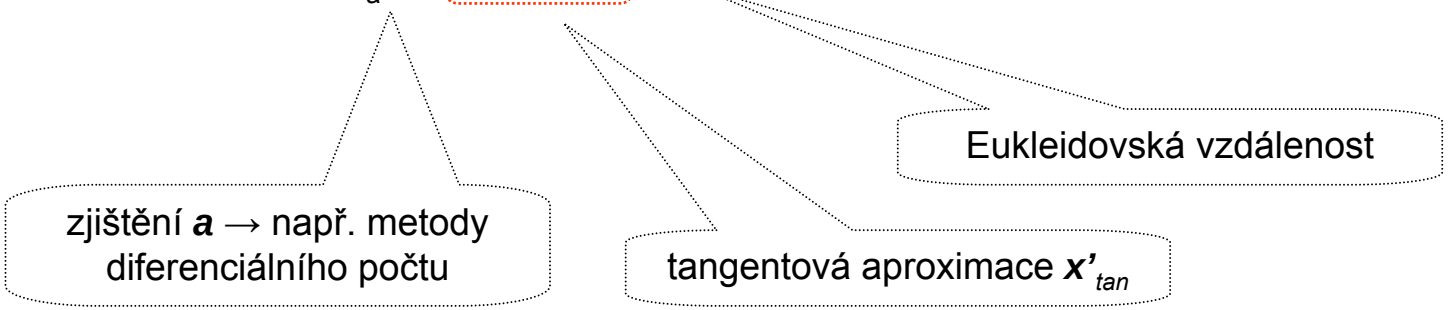
$$\mathrm{T}(\mathbf{x}') = \begin{pmatrix} \mathbf{TV}_1 \\ \vdots \\ \mathbf{TV}_r \end{pmatrix}$$

tangentové vektory

Tangentová metrika – klasifikace

- klasifikace neznámého vektoru ***x***
  - 1. spočtení tangentové vzdálenosti pro každý prototyp ***x'***

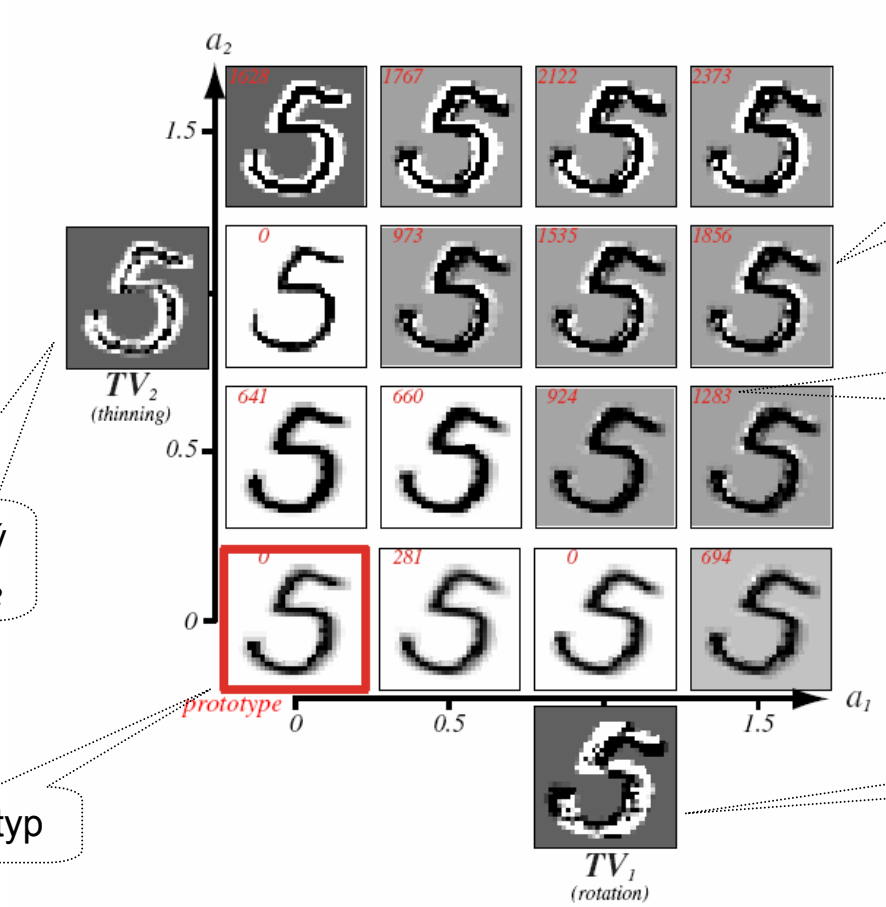
$$D_{tan}(\mathbf{x}',\mathbf{x}) = \min_{a}\left\{\left\|\left(\mathbf{x}'+T(\mathbf{x}')\mathbf{a}\right)-\mathbf{x}\right\|\right\}$$



  - 2. nalezení prototypu s minimální tangentovou vzdáleností od ***x***

Tangentová metrika – příklad

- 2 transformace
  - rotace a čárové ztenčení



lineární kombinace tangentových vektorů s koeficienty a_1, a_2

$\boldsymbol{x'}_{tan} = \boldsymbol{x}+a_1\boldsymbol{TV}_1+a_2\boldsymbol{TV}_2$

Eukleidovská vzdálenost prototypu a tangentové aproximaci $\boldsymbol{x'}_{tan}$

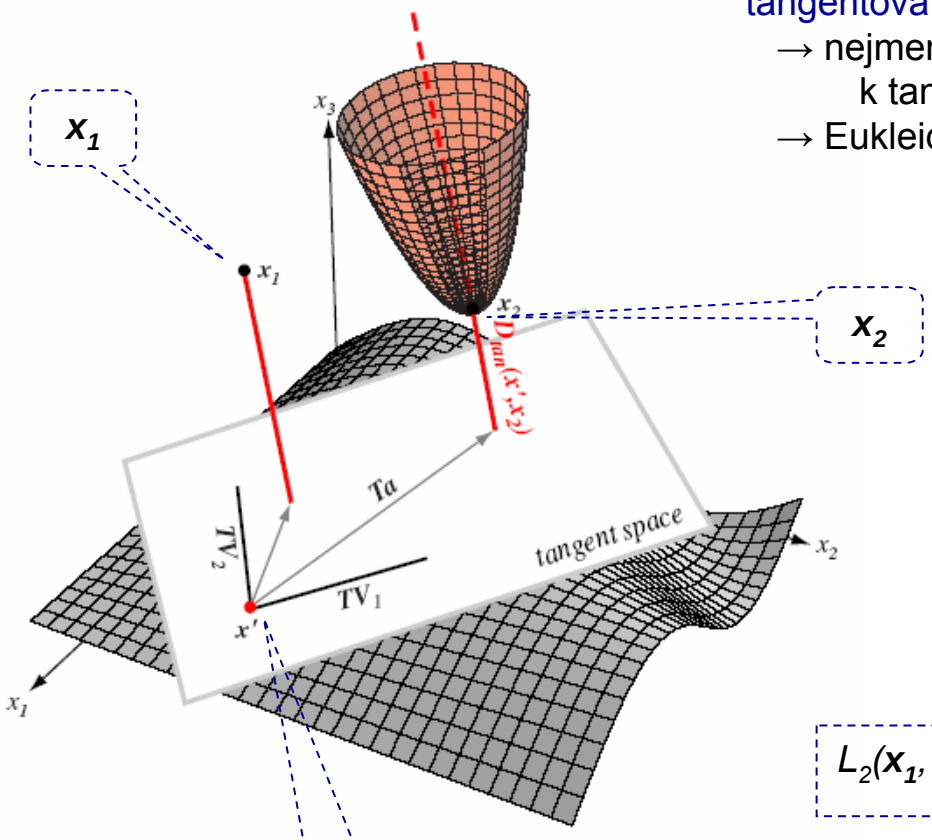
tangentový vektor $\boldsymbol{TV}_2$

tangentový vektor $\boldsymbol{TV}_1$

prototyp

Tangentová metrika – příklad

předloženy vzory x_1 a x_2



tangentová vzdálenost $D_{tan}(x_2,x') = min_a \{|| (x'+T(x')a) - x_2 ||\}$

→ nejmenší Eukleidovská vzdálenost bodu x_2 k tangentovému prostoru bodu x'

→ Eukleid. vzdálenost x_2 k tangentovému prostoru x' → kvadratická funkce proměnné a

$L_2(x_1, x') < L_2(x_2, x')$ ale $D_{tan}(x_1, x') > D_{tan}(x_2, x')$

prototyp x'