Klasifikace podle nejbližších sousedů
Nearest Neighbour Classification [k-NN]

Michal Houdek, Tomáš Svoboda, Tomáš Procházka

6. června 2001

Obsah

1 Úvod

Klasifikace podle nejbližších sousedů spadá mezi neparametrické metody klasifikace. Tyto metody jsou založeny na podstatně slabších předpokladech než metody parametrické. Nepředpokládáme znalost tvaru pravděpodobnostních charakteristik tříd.

Metoda nejbližšího souseda je založena na hledání přímo aposteriorní[1] pravděpodobnosti. Je myšlenkovým rozšířením metody klasifikace podle nejbližší vzdálenosti od etalonu.

Dalším význačným zástupcem neparametrických metod klasifikace je metoda *Parzenových okének*, která aproximuje z dat hustotu pravděpodobnosti. Touto metodou se ale nebudeme dále zabývat.

2 Definice a postup klasifikace

Známe trénovací množinu $\{(x_i, \omega_i)\}_{i=1,..,K}$. Kde x_i je vzorek, kterému je přiřazena třída ω_i a K je velikost trénovací množiny. Pro neznámý prvek x hledáme x'_k takové, že $\|x'_k - x\| = min\|x'_i - x\|_{i=1,..,K}$. Prvek zařadíme do stejné třídy, do jaké náleží x'_k.

Nejčastější je klasifikace podle jednoho souseda (1-NN), ale existují i klasifikace pro obecně k sousedů.

3 Příklady použití

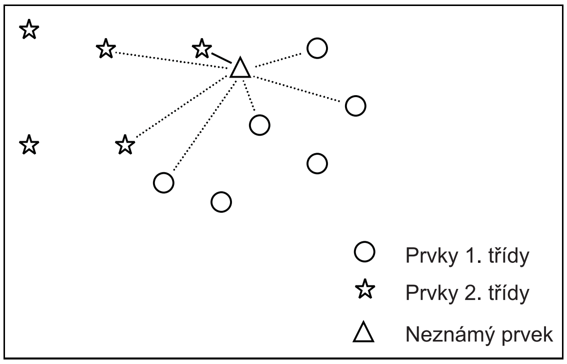
3.1 Příklad použití 1-NN

Zjistíme vzdálenosti všech prvků trénovací množiny od neznámého prvku. Vybereme ten prvek trénovací množiny, který je nejblíže a neznámý prvek klasifikujeme do stejné třídy. V našem případě je nejblíže prvek 2. třídy $\Rightarrow$ neznámý prvek klasifikujeme jako prvek 2. třídy (viz obr. 1).

3.2 Příklad použití 3-NN

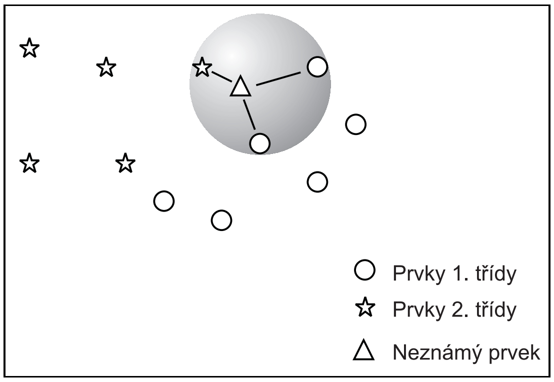
Kolem neznámého prvku vytvoříme hyperkouli, která obsahuje právě tři nejbližší prvky trénovací množiny. Neznámý prvek klasifikujeme do té třídy,

[1] mající zkušenostní povahu



Obrázek 1: Popis klasifikace 1-NN

která je v hyperkouli zastoupena největším počtem prvků. V našem případě klasifikujeme neznámý prvek do 1. třídy (viz obr. 2). Při použití metod k-NN pro k>1 je velmi důležitá volba k. Pro dvě třídy volíme k vždy liché (kvůli jednoznačnosti rozhodování) pro více tříd mohou nastat situace, kdy nelze jednoznačně rozhodnout.



Obrázek 2: Popis klasifikace 3-NN

4 Vlastnosti k-NN klasifikátoru

4.1 Klady k-NN klasifikátoru

- triviální návrh a implementace ⇒ slouží jako dobrá referenční metoda
- v aplikacích často vychází $P^{NN}(\varepsilon) \approx P^{neur.}(\varepsilon)$ - chyby metod k-NN a Neuronové sítě jsou srovnatelné

4.2 Zápory k-NN klasifikátoru

- pro 1-NN platí $P^{NN}(\varepsilon) > P(\varepsilon)$ a to i při velké trénovací množině, $P(\varepsilon)$ je chyba Bayesovského klasifikátoru,
  ale $lim_{K \to \infty} P^{NN}(\varepsilon) < 2.P(\varepsilon)$ - asymptotickou chybu[2] můžeme dále zmenšit použitím "editace" trénovací množiny (viz dále)
- pomalé rozhodování,
  ale existují i rychlé vyhledávací algoritmy (viz níže)
- vysoká paměťová náročnost,
  ale možnost kondenzace trénovací množiny (viz níže)
- závislý na metrice $\|.\|$ - záleží na měřítku $\Rightarrow$ potřeba normalizovat
- žádná generalizace
  - Vapnik - Červoněnkisova dimenze je ∞
  - chyba na trénovací množině je 0, tj. přestože na trénovací množině získáme nulovou chybu klasifikace, nelze vyjadřit jaká chyba nastane na testovacích datech - může být jakákoliv

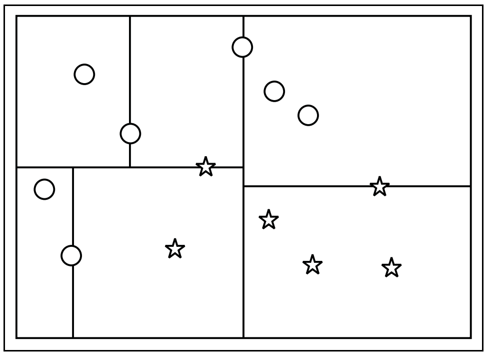
5 Urychlení k-NN klasifikace

5.1 Sofistikovaný algoritmus vyhledání nejbližších k-sousedů

- klasický problém výpočetní geometrie (computation geometry) postupy rychlejší než $O(logN)$, pro d[3]=2,..,k
- k-D stromy - data trénovací množiny setřídíme podle jedné souřadnice, nalezneme medián a podle něj rozdělíme data na dvě množiny. Vzniklé podmnožiny opět dělíme podle další souřadnice. Takto postupujeme dokud nezískáme buňky obsahující právě jeden bod. Při půlení množin střídáme cyklicky souřadnice podle kterých množiny půlíme. Při klasifikaci nových bodů postupujeme následovně:
  1. Nalezneme buňku do které patří nový bod.

[2]chyba při velikosti trénovací množiny jdoucí k ∞
[3]d je dimenze prostoru vzorků

2. Změříme vzdálenost nového bodu od bodu ležícího v dané buňce a vzdálenosti od všech hranic buňky. Pokud bod trénovací množiny leží blíže než všechny hranice, nalezli jsme nejbližšího souseda a algoritmus končí.
3. V případě, že něktreré hranice jsou blíž, než bod trénovací množiny musíme prohledat i buňky za těmito hranicemi.



Obrázek 3: k-D strom

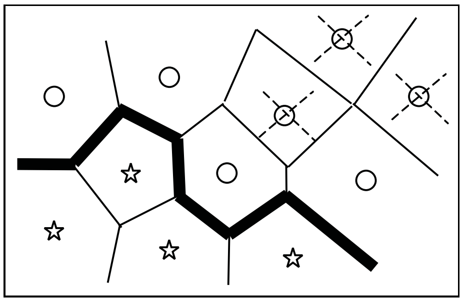
5.2 Vyloučení vzorků, které neovlivňují klasifikaci (G. Toussaint)

5.2.1 Vyloučení pomocí Voroniova diagramu

Voroniův diagram vytváříme následovně. Nejprve nalezneme hrany Voronoiova diagramu. Mezi dvěma prvky množiny existuje Voronoiova hrana právě tehdy když existuje bod, který je od těchto dvou prvků stejně daleko a všechny ostatní prvky množiny leží od tohoto bodu dál. Mezi takovými dvěma prvky vedeme stěnu tak, že oba prvky jsou podle ní osově souměrné. Spojnice těchto prvků je na stěnu kolmá a stěna jí protíná v polovině délky. Spojením stěn vzniknou Voronoiovy buňky kolem každého bodu, viz Obrázek 4.

Lze dokázat, že vyloučením buněk, jejichž Voronoiova buňka sousedí pouze s buňkami vzorku stejné třídy se nezmění rozhodovací nadplocha. Tím dosáhneme zrychlení klasifikace.

Nevýhodou Voronoiova diagramu je jednak složitost jeho výpočtu, která je rovna $O(NlogN + N^{d/2})$ a také fakt, že pro dimenze výrazně větší než 1 leží téměř všechny body poblíž rozhodovací nadplochy. Vyloučených bodů je tedy málo a tím se výhoda Voronoiova diagramu ztrácí.



Obrázek 4: Voroniův diagram

5.2.2 Aproximační metody

Například lze použít místo Voroniova grafu Gabrielův graf. Jeho složitost je $O(N^2)$

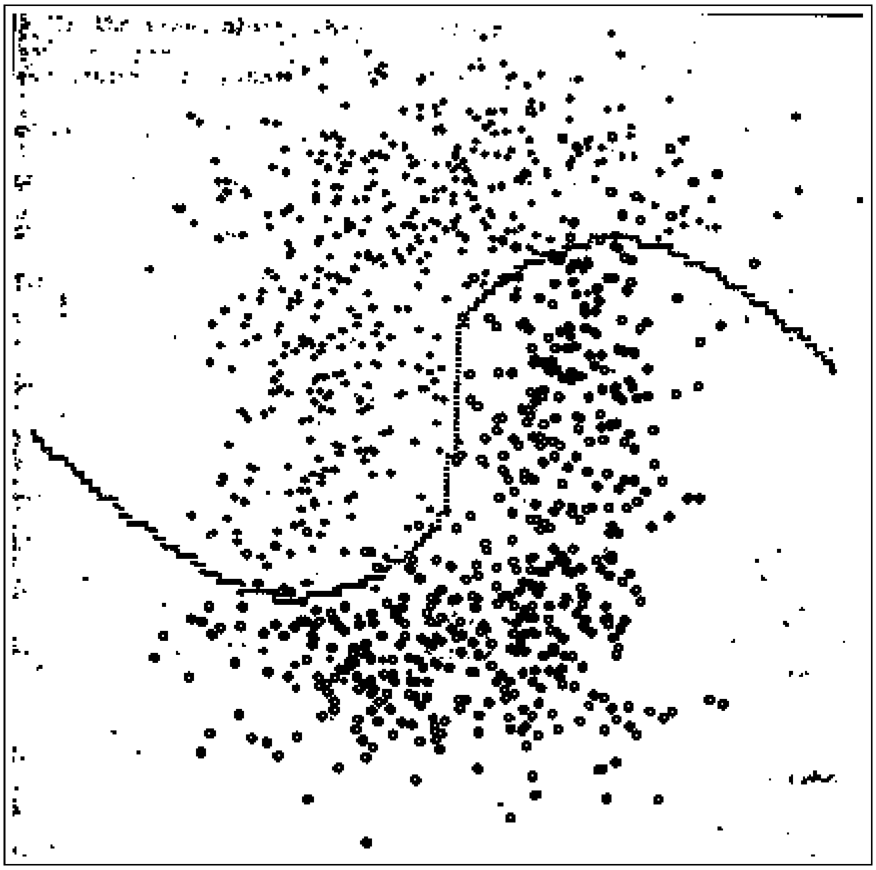
5.3 Kondenzace (condensing) - Devijer, Kittler

Metody kondenzace jsou *iterativní* a *aproximační*. Aproximační metody nejsou optimální, ale nezvětšují chybu klasifikátoru. Jsou jednodušší na výpočet.

5.3.1 Kondenzace trénovací množiny pro 1-NN

- vlož do seznamu A náhodně vybraný vzorek, ostaní vzorky z trénovací množiny vlož do seznamu B.
- a) klasifikuj vzorky z B metodou 1-NN s trénovací množinou A
  b) je-li x_i klasifikován nesprávně, přesuň x_i z B do A
- došlo-li v předchozím kroku při průchodu celým seznamem B k přesunu, zopakuj předchozí krok.

Výstupem algoritmu je množina A což je kondenzovaná trénovací množina pro 1-NN. Tato metoda nedává při týchž datech vždy stejnou rozhodovací nadplochu, závisí na inicializaci. Každá z možných rozhodovacích nadploch je ale stejně vhodná a platná. Na obrázcích 5 a 6 je příklad kondenzace trénovací množiny.



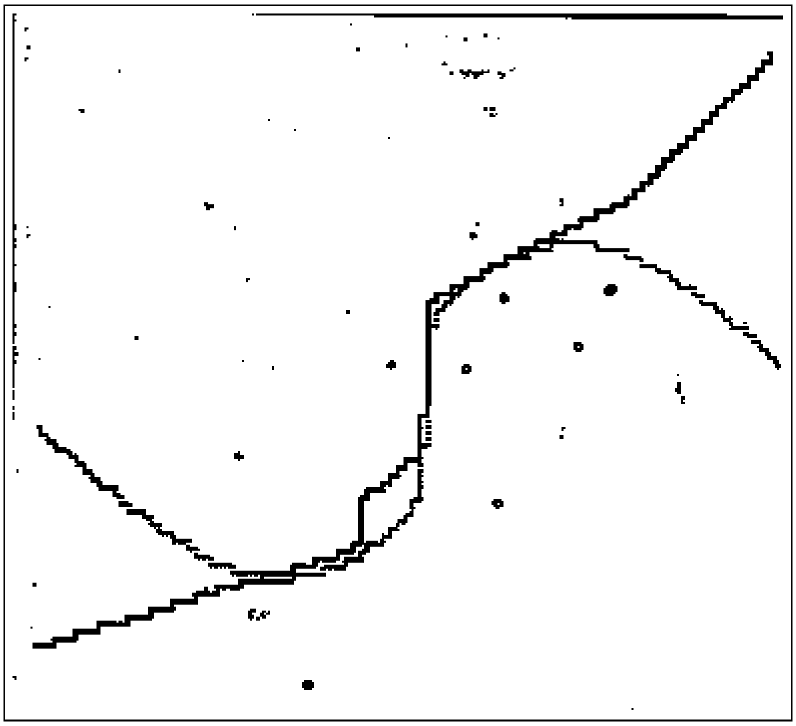
Obrázek 5: Příklad kondenzace - počáteční dvourozměrná data. Tečkovaná čára znázorňuje rozhodovací nadplochu Bayesovského klasifikátoru.

6 Editace trénovací množiny

Cílem editace je *snížení chyby* $P^{NN}(\varepsilon)$, ne zrychlení postupu klasifikace. Jedná se o vyloučení prvků z trénovací množiny, které podle hustoty baysovské pravděpodobnosti vnášejí chybu do klasifikace (samostatné prvky jedné třídy, obklopené prvky z jiné třídy). Tyto prvky jsou většinou chyby měření nebo odchylky od typických zástupců.

6.1 Alogoritmus editace trénovací množiny

- náhodně rozdělíme trénovací množinu S_n na dvě poloviny S_{n_1} a S_{n_2}, $S_n = S_{n_1} + S_{n_2}$
- klasifikujeme vzorky z S_{n_1} metodou k-NN. S_{n_2} používáme jako tréno-



Obrázek 6: Příklad kondenzace - zkondenzovaná trénovací množina. Plnou čarou je znázorněna rozhodovací nadplocha NN klasifikátoru. Je patrné, že v místech, kde se rozhodovací nadplochy rozcházejí, je malá pravděpodobnost výskytu dat.

vací množinu.

- "vyeditujeme" (vyloučíme) z S_{n_1} vzorky, které nebyly správně klasifikovány v předchozím kroku
- vzniklou množinu S'_{n_1} použijeme ke klasifikaci metodou k-NN

6.2 Asymptotická analýza

$$P^{edit}(\varepsilon) = P(\varepsilon)\frac{1-P(\varepsilon)}{1-P^{kNN}(\varepsilon)}$$

Je-li $P^{kNN}(\varepsilon) << 1$ (např. 0,05), pak je editovaný 1-NN *quasi-Bayesovský*, tj. téměř nerozlišitelný od Bayesovského klasifikátoru. Je třeba vzít v úvahu, že platí $P^{kNN}(\varepsilon) > P(\varepsilon)$ a proto je zlomek vždy "o kousek" větší než 1.

7 Asymptotická chyba k-NN klasifikátoru

Chyba klasifikace nejbližšího souseda při K vzorcích:

$$P^{NN}(\varepsilon) = \lim_{K\to\infty} P_K^{NN}(\varepsilon)$$

Střední hodnota průměru asymptotické chyby:

$$P^{NN}(\varepsilon) = \int_x P^{NN}(\varepsilon|x)p(x)dx$$

$$P_K^{NN}(\varepsilon|x) = \int \overbrace{P_K^{NN}(\varepsilon|x, x'_k)}^{1.} \overbrace{p(x'_k|x)}^{2.} dx'_k$$

Je-li počet vzorků velmi vysoký je pravděpodobnost, že x'_k je nejbližším sousedem s x, funkcí jejich vzdálenosti:

$$K >> 1 \to p(x'_k|x) = \delta(x'_k - x)$$

Pravděpodobnost, že jak x, tak x'_k jsou ze stejné třídy jako ω_r:

$$P(\omega_r, \omega_r|x, x'_k) = P(\omega_r|x)P(\omega_r|x'_k)$$

Chybná klasifikace pak nutně nastává v případech pro než x a x'_k nejsou ze stejné třídy:

$$P_K^{NN}(\varepsilon|x, x'_k) = 1 - \sum_{r=1}^{R} P(\omega_r|x)P(\omega_r|x'_k)$$

Po zpětném dosazení do předchozích vztahů získáme:

$$P^{NN}(\varepsilon|x) = \lim_{K\to\infty} P_K^{NN}(\varepsilon|x) = \int [1 - \sum_{r=1}^{R} P(\omega_r|x)P(\omega_r|x'_k)]\delta(x'_k - x)dx'_k =$$
$$1 - \sum_{r=1}^{R} P^2(\omega_r|x)$$

Asymptotická chyba je pak vyjádřena:

$$\boldsymbol{P^{NN}(\varepsilon) = \int [1 - \sum_{r=1}^{R} P^2(\omega_r|x)]p(x)dx}$$

7.1 Maximum chyby P^{NN}

Lze dokázat (metodou Lagrangeových multiplikátorů), že $P^{NN}(\varepsilon)$ nabývá maxima, když

$$P(\omega_s|x) = \tfrac{1-P(\omega_r|x)}{R-1} = \tfrac{P(\varepsilon|x)}{R-1} \qquad s \neq r$$

$$P(\omega_s|x) = 1 - P(\varepsilon|x) \qquad s = r$$

tj. když je pro všechny třídy stejná chyba klasifikace.

Potom

$$\sum_{r=1}^{R} P^2(\omega_r|x) = [1 - P(\varepsilon|x)]^2 + (R-1)\left(\tfrac{P(\varepsilon|x)}{R-1}\right)^2$$

po úpravě pravé strany rovnice

$$\sum_{r=1}^{R} P^2(\omega_r|x) = 1 - 2P(\varepsilon|x) + P^2(\varepsilon|x) + \tfrac{P^2(\varepsilon|x)}{R-1}$$

což vede k hledanému tvaru

$$P^{NN}(\varepsilon|x) = 1 - \sum_{r=1}^{R} P^2(\omega_r|x) = 2P(\varepsilon|x) - \tfrac{R}{R-1}P^2(\varepsilon|x)$$

který udává chybu odhadu pro konkrétní x. Abychom získali $P^{NN}(\varepsilon)$ musíme integrovat přes všechna x. Zde si pomůžeme vyjádřením rozptylu, z něhož vyplyne převod jinak problematického členu $P^2(\varepsilon|x)$.

$$var\{P(\varepsilon|x)\} = \int_x [P(\varepsilon|x) - P(\varepsilon)]^2 p(x)dx = \int P^2(\varepsilon|x)p(x)dx - P^2(\varepsilon) > 0$$

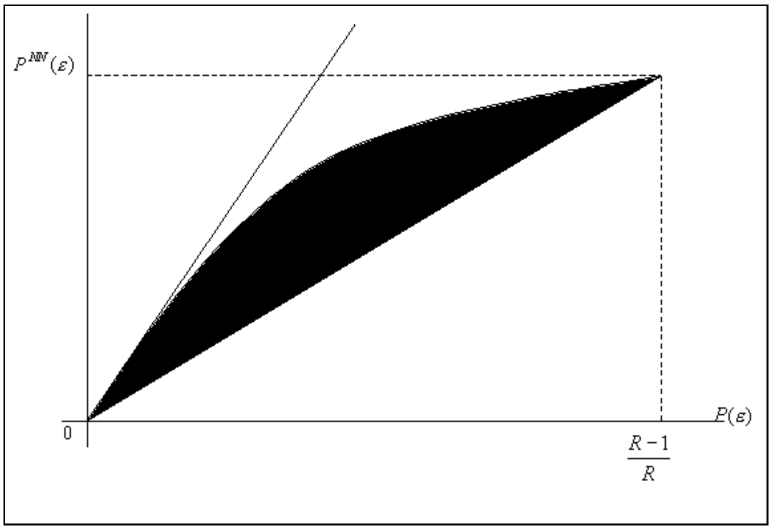
Vidíme tedy, že $P^2(\varepsilon) < \int P^2(\varepsilon|x)p(x)dx$ a můžeme tedy naší rovnici přepsat nerovnicí

$$P^{NN}(\varepsilon) \leq 2P(\varepsilon) - \tfrac{R}{R-1}P^2(\varepsilon)$$

Závěrem pak můžeme zopakovat již dříve uvedenou a nyní dokázanou vlastnost

$$\boldsymbol{P(\varepsilon) \leq P^{NN}(\varepsilon) \leq 2P(\varepsilon)}$$

Vztah mezi chybou Bayesovského klasifikátoru a asymptotickou chybou metody nejbližšího souseda přehledně znázorňuje Obrázek 7.



Obrázek 7: Vztah mezi chybou Bayesovského klasifikátoru a asymptotickou chybou metody nejbližšího souseda

7.2 Chyba k-NN klasifikátoru

Platí, že chyba metody nejbližších sousedů klesá s "počtem" nejbližších sousedů a dále ji lze snížit editací trénovací množiny, jak bylo uvedeno výše.

$$P^{NN}(\varepsilon) \leq P^{kNN}(\varepsilon) \leq P^{3NN}(\varepsilon) \leq P^{1NN}(\varepsilon) \leq 2P(\varepsilon)$$

$$\boldsymbol{P^{kNN}(\varepsilon) \leq P(\varepsilon) + \frac{P^{1NN}}{\sqrt{k\pi}}}$$

8 Příklad - Porovnání Bayesovského a 1-NN klasifikátoru

$P(\omega_1|x) = 0,6$
$P(\omega_2|x) = 0,2$
$P(\omega_3|x) = 0,2$
Bayesovská chyba:

$$P(\varepsilon|x) = 1 - \max_{\omega_i}(P(\omega_i|x)) = 1 - 0,6 = 0,4$$

Asymptotická chyba:

$$P^{NN}(\varepsilon|x) = 1 - \sum_{\omega_i} P^2(\omega_i|x) = 1 - (0,36 + 0,04 + 0,04) = 0,56$$