FAKULTA APLIKOVANÝCH VĚD ZÁPADOČESKÉ UNIVERZITY

Vylepšení algoritmu „Nejbližšího souseda“ pomocí jednoduché adaptivní míry vzdálenosti

Vladimír Láznička

18. 10. 2015

Obsah

Úvod

Algoritmus *Nearest Neighbor* (dále *NN*) je jedním z nejjednodušších a nejstarších (byl navržen v roce 1951) algoritmů **pro klasifikaci příznaků** – určování jejich třídy. Jeho použití spočívá v určování třídy klasifikovaného vzoru na základě třídy nejbližšího (nejpodobnějšího) sousedního vzoru (patřícího do množiny takzvaných **trénovacích vzorů**, u kterých známe jejich třídu) ve vstupním prostoru R^d o dimenzi d. K určení míry vzdálenosti (podobnosti) se pak typicky používá *Euklidovská* nebo *Manhattanská* vzdálenostní funkce. Jakousi nástavbou tohoto algoritmu je pak *K Nearest Neighbors* (dále *KNN*) algoritmus (navržen v roce 1977). Ten určuje třídu klasifikovaného vzoru ne pouze jedním nejbližším známým vzorem, ale celkem K nejbližšími vzory obsaženými v prostoru R. Třída je pak zvolena ta, která byla **zastoupena v množině K nejbližších sousedních vzorů nejčastěji**. Tato úprava typicky eliminuje část možných nepřesností způsobených nahodile rozmístěnými trénovacími vzory, pokud se v prostoru **R** mezi sebou prolínají. Přístup takového algoritmu je snadno pochopitelný a poměrně intuitivní, proto je *KNN* oblíbeným a často používaným algoritmem pro klasifikaci příznaků, např. pro vyhodnocování příznakových vektorů získaných z různých signálů.

V závislosti na konkrétní aplikaci algoritmu, může *KNN* dosahovat velmi dobré přesnosti, nicméně nemusí být příliš efektivní právě v situacích, **kdy se trénovací vzory s různými třídami navzájem prolínají** nebo se ve shlucích vzorů jedné třídy nachází jeden či několik vzorů jiných tříd, což může snížit *KNN* klasifikaci neznámého vzoru. Jednou z možností, jak tyto nevýhody omezit, je určení metriky (nebo váhy) jednotlivým příznakům (dá se také říci dimenzím v prostoru vzorů) v klasifikovaném vzoru, kde ty více relevantní příznaky mají vyšší vliv na určení podobnosti než ty méně relevantní. Výsledkem je pak potenciálně přesnější klasifikace, ovšem za cenu citelně vyšší výpočetní náročnosti algoritmu. Další možností je využití **adaptivní míry vzdálenosti**, která je rozebírána v tomto textu.

Popis algoritmu KNN

V této kapitole bude vysvětlen princip činnosti a vlastnosti standardního algoritmu *KNN* společně s uvedením jednoduchého pseudokódu. To by mělo pomoci čtenáři se zorientováním se v kapitole následující, kde bude popsáno vylepšení algoritmu o adaptivní míru vzdálenosti.

Princip činnosti algoritmu

Jak bylo zmíněno v úvodu, algoritmus obecně funguje na principu přiřazování třídy jednotlivým klasifikovaným vzorům na základě podobnosti s K nejbližšími trénovacími vzory nacházející se ve stejném prostoru R^d s dimenzí d. Za vzory zde považujeme uspořádané dvojice ve tvaru: $D = (\vec{X}, Y)$. X představuje takzvaný příznakový vektor (reprezentovaný např. jako jednorozměrné pole reálných čísel), jejichž **hodnoty na jednotlivých pozicích**

představují právě jednotlivé příznaky. `Y` pak značí třídu vzoru, ta může být vyjádřena číslem, znakem… záleží na dohodnutém značení. Pro funkci algoritmu je samozřejmě zapotřebí, aby všechny vektory spadaly do stejného lineárního (pod)prostoru, tedy aby měly **stejnou dimenzi**.

Podobnost mezi vzory je pak určena pomocí zvolené vzdáleností funkce, která vypočítá vzdálenost mezi příslušnými vektory – **čím menší vzdálenost v prostoru R^d, tím podobnější jsou si vzory**. Mezi nejčastěji používané vzdálenostní funkce patří *Euklidovská* a *Manhattanská* funkce.

Vzorec pro Euklidovskou funkci: $d(A,B) = d(B,A) = \sqrt{\sum_{i=1}^{d}(B_i - A_i)^2}$

- `A` a `B` představují vektory, `d` v sumě je pak jejich dimenze (kolik mají prvků).
- Z geometrického hlediska si lze (v R^2 a R^3) představit výsledek jako přímou vzdálenost mezi dvěma vektory.

Vzorec pro Manhattanskou funkci: $d(A,B) = d(B,A) = \sqrt{\sum_{i=1}^{d}|B_i - A_i|}$

- `A` a `B` představují vektory, `d` v sumě je pak jejich dimenze (kolik mají prvků).
- Výsledek si lze představit jako nejkratší vzdálenost, jakou by cestovatel musel urazit v blokové zástavbě z jednoho bodu do druhého.

Algoritmus tedy **iterativně vypočítá vzdálenost mezi klasifikovaným vektorem příznaků a celou množinou, řekněme o velikosti `N`, trénovacích vektorů**. Z této množiny pak vybere `K` vektorů, které mají nejmenší vzdálenost od toho klasifikovaného, a z příslušných vzorů přečte jejich třídu. Klasifikovanému vzoru je pak přidělena třída, která byla v rámci vybraných trénovacích vzorů **zastoupena nejvíce**.

Pseudokód

```
Vstup:      trenovaciMnozina = {(x1, y1), ..., (xN, yN)}
            klasifikovanyVzor = (x, y)
            cisloK = k

BEGIN
      FOR EACH instance IN trenovaciMnozina
      DO
            vypocitejVzdalenost(instance, klasifikovanyVzor)
      LOOP

      seradPodleVzdalenosti(trenovaciMnozina)
      nejblizsiSousede := vyberKNejblizsich(k, trenovaciMnozina)
      urcenaTrida := vyberTridu(nejblizsiSousede)
      klasifikovanyVzor.y := urcenaTrida
END
```

Dodatečné vlastnosti algoritmu

KNN patří do kategorie tzv. **líných (lazy) klasifikačních algoritmů** – v zásadě nedochází k žádnému tréninku, pouze se vloží trénovací vzory, se kterými se pak dále zachází. To znamená, že je okamžitě použitelný pro příchozí klasifikované vzory, ale zejména díky nutnosti **výpočtu vzdálenosti mezi vektory na celé trénovací množině při každém novém vzoru** může být algoritmus v závislosti na velikosti zmíněné množiny pomalejší.

Přesnost algoritmu je také silně ovlivněna výběrem vhodného množství (`K`) sousedních vzorů ke klasifikaci. Existují mnohé případové studie, které se výběrem vhodného čísla zabývají, nicméně to není předmětem tohoto textu.

Adaptivní míra vzdálenosti v algoritmu KNN

Princip úpravy algoritmu

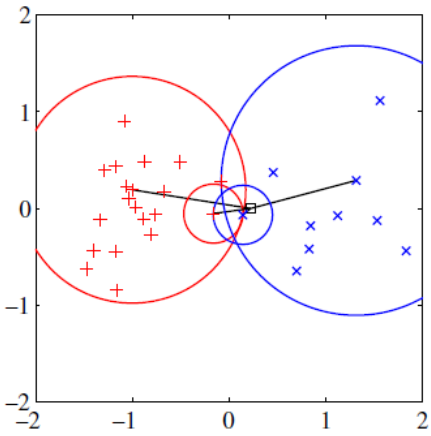
Princip spočívá v ohodnocování jednotlivých trénovacích vzorů pomocí nějaké hodnoty, která nám určuje jakousi „výhodnost“ nebo „spolehlivost“ daného trénovací vzoru při klasifikaci vzoru s neurčenou třídou. **Za „spolehlivé“ vzory považujeme takové, které se v prostoru $\mathbb{R}^d$ nacházejí mezi ostatními vzory stejné třídy** (v nějakém shluku). Pokud se tedy např. nějaký vzor třídy T_1 vyskytne uvnitř nebo poblíž shluku vzorů třídy T_2, lze jej považovat za jakousi výjimku a jeho „spolehlivost“ v určování třídy klasifikovaného vzoru bude významně nižší.

Z geometrického hlediska si lze představit **sféru okolo vektoru daného vzoru, která dosahuje k nejbližšímu dalšímu vektoru s jinou třídou**. Čím má tato sféra větší poloměr, tím je daný vzor „výhodnější“. Právě poloměr takové sféry bude představovat zmíněnou hodnotu, kterou přiřadíme jednotlivým vzorům. Tento poloměr si vyjádříme následujícím vzorcem:

$$r_i = \min_{l:Y_l \neq Y_i} d\left(\vec{X}_i, \vec{X}_l\right) - \varepsilon; \quad l, i = 1, \dots, N$$

- Jako vzdálenostní funkci opět využijeme Euklidovskou/Manhattanskou. Vybereme takovou, aby korespondovala se vzdálenostní funkcí použitou při samotné klasifikaci.
- Y značí třídu vzoru, X pak jeho vektor.
- ε je pak nespecifikovaná proměnná kladné (typicky zanedbatelné) hodnoty. V praktickém použití ji ignorujeme.

Tento poloměr je tedy určován pro každý trénovací vzor a udává vzdálenost mezi vektorem X_i náležícím vzoru, pro který poloměr r_i počítáme, a nejbližším vektorem X_l s jinou třídou Y_l.



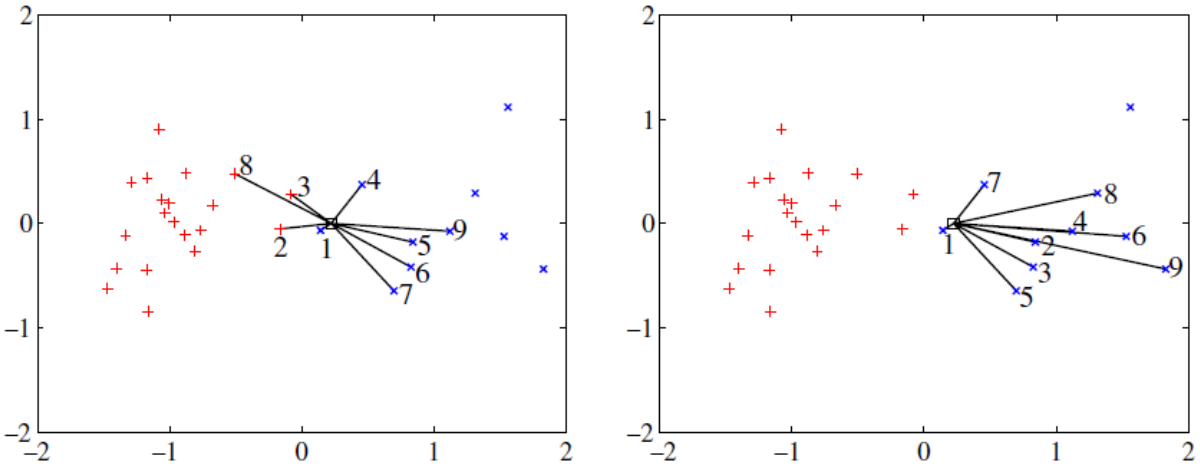
Obrázek 1 - sféra okolo vektorů, které se nacházejí uvnitř shluků je větší než okolo vektorů na jejich okrajích

Zmiňovaná **adaptivnost vzdálenosti** se pak uplatňuje při samotné klasifikaci, resp. při určování vzdáleností mezi vektorem klasifikovaného vzoru a vektory trénovacích vzorů. Platí tedy, že čím větší je určený *poloměr imaginární sféry* daného vzoru, tím se její vektor považuje za bližší vektoru klasifikovaného vzoru. Vzorec pro výpočet vzdálenosti mezi vektorem klasifikovaného vzoru a všemi trénovacími vektory bude tedy následující:

$$d_{new}(\vec{X}, \vec{X}_i) = \frac{d(\vec{X}, \vec{X}_i)}{r_i}$$

- X značí vektor klasifikovaného vzoru, X_i je pak vektor trénovacího vzoru.
- r_i je poloměr imaginární sféry trénovacího vzoru.

Vzorec nám v zásadě říká, že příznakové vektory, jejichž vzorům byl přiřazen vyšší *poloměr*, se v konečném výpočtu jeví jako bližší, resp. jejich vzory se jeví jako více podobné tomu klasifikovanému.



Obrázek 2 - Srovnání výběru 9 nejbližších sousedních vektorů při klasifikaci. Vlevo - KNN, vpravo - KNN s adaptivní mírou vzdálenosti.

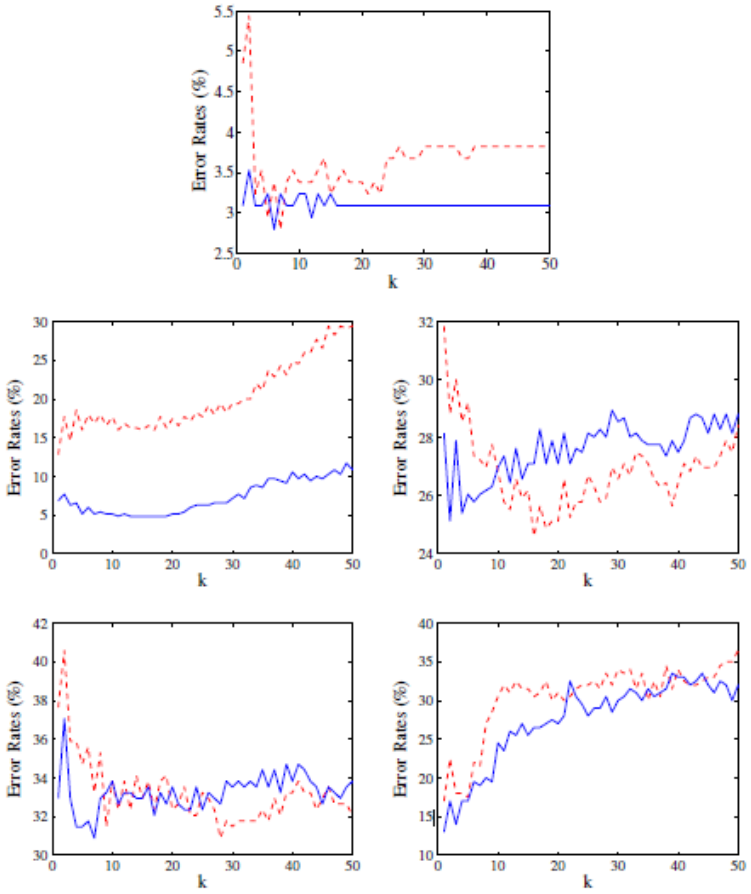
Shrnutí úpravy

Adaptivní míra vzdálenosti by měla výrazně omezit vliv „méně spolehlivých“ trénovacích vzorů na výsledek klasifikace třídy, přičemž nemá žádný citelný vliv na výkon algoritmu. Důvodem toho je, že určení *poloměru imaginární sféry* jednotlivých trénovacích vzorů proběhne právě jednou ještě před samotným procesem klasifikace (za předpokladu, že neměníme v průběhu používání algoritmu množinu trénovacích vzorů), lze ji tedy považovat za určitou formu trénovaní jinak líného (lazy) klasifikačního algoritmu.

Výsledky algoritmu

Testy algoritmu probíhaly na **klasifikaci dat získaných z pěti různých reálných lékařských měření**. Porovnání probíhalo vůči klasickému *KNN* algoritmu bez adaptivní míry vzdálenosti a to jak s použitím *Euklidovské*, tak *Manhattanské* vzdálenostní funkce. Měřila se **chybovost** (jaké % klasifikovaných vzorů bylo klasifikováno špatně) při použití různé hodnoty `K`. Grafy jsou převzaty z článku (1).

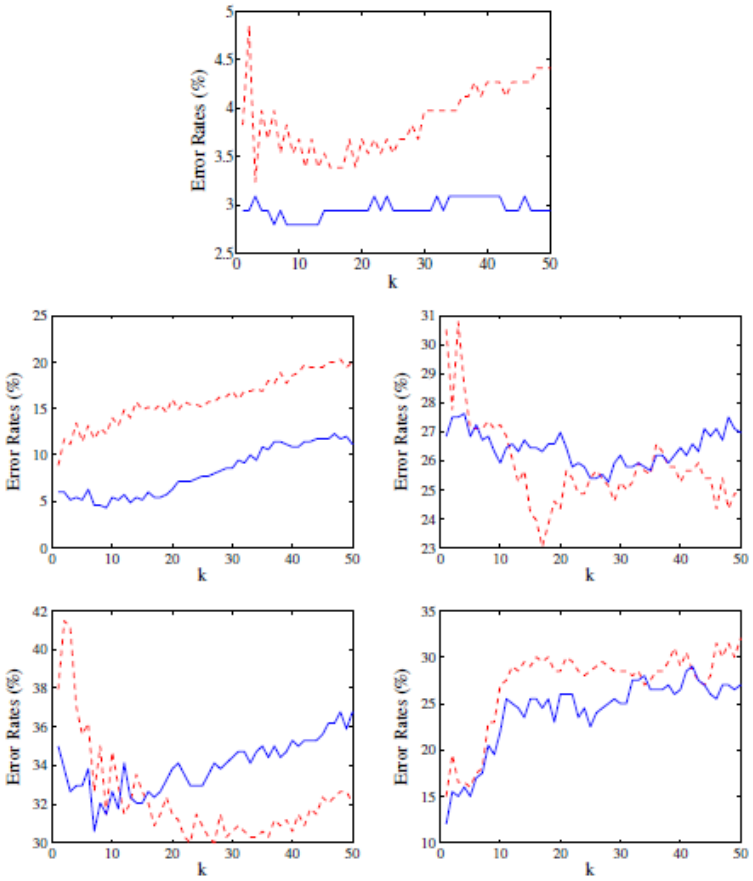
- Porovnání chybovosti algoritmu *KNN* a jeho adaptivní verze v závislosti na parametru `K` - *Euklidovská* vzdálenostní funkce:



Obrázek 3 - Porovnání chybovosti klasického KNN algoritmu (červený graf) a adaptivního KNN (modrý graf) algoritmu při použití Euklidovské vzdálenostní funkce. Osa Y = chybovost [%], Osa X = hodnota K.

Jak si lze všimnout, *KNN* s adaptivní mírou vzdálenosti poskytuje nižší chybovost především **pro nižší množství nejbližších sousedů** (`K < 10`), u vyšších hodnot se výsledky srovnávají, případně otočí. Nicméně v případě některých datasetů bylo dosaženo výrazně nižší chybovosti i při vysokém počtu nejbližších sousedů (`K = 50`).

- Porovnání chybovosti algoritmu *KNN* a jeho adaptivní verze v závislosti na parametru `K` – *Manhattanská* vzdálenostní funkce:



Obrázek 4 - Porovnání chybovosti klasického KNN algoritmu (červený graf) a adaptivního KNN (modrý graf) algoritmu při použití Manhattanské vzdálenostní funkce. Osa Y = chybovost [%], Osa X = hodnota K.

I v případě použití *Manhattanské* vzdálenostní funkce lze vysledovat podobný trend – **výsledky při použití adaptivní míry vzdálenosti jsou lepší při nižší hodnotě `K`**. U některých datasetů jsou pak lepší výsledky patrné i pro vysoké hodnoty `K`.

Závěr

Z výsledků je patrné, že ačkoliv zkoumaná úprava algoritmu je velice jednoduchá, nezanedbatelně zvyšuje jeho úspěšnost při klasifikaci určitých vzorů. Z principu vyplývá, že **největší úspěch (ve srovnání s klasickým *KNN* algoritmem) bude mít v případě dat, kde bude docházet k určité míře prolínání vzorů jiných tříd**. Z výsledků také plyne, že obecně užitečnější se zdá být **při použití nižšího počtu sousedních vzorů ke klasifikaci**, neboť při větším počtu už zisk přesnosti nebyl příliš znatelný a v některých případech dokonce žádný. Výhodou použití adaptivní míry vzdálenosti je však také v podstatě **nulové navýšení výpočetní náročnosti klasifikace**. Výpočet hodnoty *poloměru imaginární sféry* trénovacích vzorů proběhne před samotnou klasifikací, tedy ve fázi trénování klasifikátoru, což zpravidla nepředstavuje problém.

Bibliografie

1. Jigang Wang, Predrag Neskovic and Leon N. Cooper / ELSEVIER Pattern Recognition Letters, Volume 28, Issue 2, 15 January 2007, Pages 207–213 (http://www.sciencedirect.com/science/article/pii/S0167865506001917)
2. K-Nearest Neighbors Algorithm, Wikipedia (https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm)