Adaptivní míra vzdálenosti

VYLEPŠENÍ KLASIFIKAČNÍHO ALGORITMU
K-NEAREST NEIGHBORS

Úloha klasifikace/rozpoznávání příznaků

- Klasifikace objektů/vzorků do tříd
- Vzorek klasifikace
  - $D = (\vec{X}, Y)$.
- Vektor příznaků
  - X = [x1, x2, …, xn]
  - Typicky Integer nebo Real
  - Dimenze velikosti n
- Klasifikační třída
  - Integer, Real, Char, String…

Obecný proces klasifikace příznaků

- Trénovací objekty/vzorky
  - Předem známá třída
- Testovací objekty/vzorky
  - Snažíme se klasifikovat jejich třídu
- Algoritmy pro klasifikaci
  - Výběr příznaků
  - Fáze trénování
  - Fáze klasifikace
  - Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Linear Discrimination Analysis (LDA), Neurální sítě, Perceptrony…

Využití klasifikace/rozpoznávání příznaků

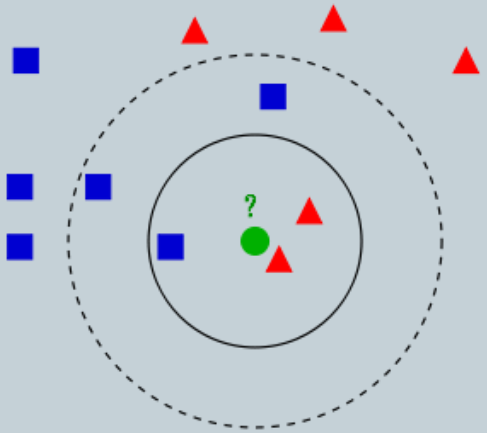
- Využití v medicíně (proces diagnózy)
  - Klasifikace nádorových onemocnění
  - Klasifikace naměřeného EEG signálu
- Identifikace a autentikace
  - Otisky prstů, sítnice oka
  - Rozeznání obličeje
- Navigace, rozpoznávání cílů
  - Autonomní systémy
  - Rozeznávání tvarů
- Rozpoznání řeči

Algoritmus K-Nearest Neighbors

- Jeden z nejjednodušších algoritmů ve strojovém učení
- Jedná se o tzv. „lazy“ algoritmus (trénovaní zde prakticky neprobíhá, vše se řeší až při samotné klasifikaci)
- Založený na klasifikaci třídy pomocí *k* nejbližších sousedů ve vstupním prostoru
- Nearest Neighbor (NN) algoritmus – původní verze algoritmu – je použit pouze jeden nejbližší soused

Znázornění činnosti KNN

- Klasifikovanému (testovacímu) vzorku je určena třída podle *k* nejbližších sousedních vzorků



- Pro určení nejbližších sousedních vzorků se používá vzdálenostní funkce pro výpočet vzdálenosti mezi jejich vektory (typicky Euklidovská nebo Manhattanská)

Pseudokód KNN algoritmu

```
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
```

Některé vlastnosti algoritmu

- Je poměrně výpočetně náročný
  - N*(*výpočet vzdálenosti*) + N*log(N) + k
- Je deterministický (při shodné hodnotě *k* vyjde klasifikace vždy stejně)
- Výsledek klasifikace značně závisí na výběru vstupních dat (trénovacích vzorků i použitých příznaků) a na zvolení vhodné hodnoty *k*
- **Algoritmus může mít problémy s přesností, pokud jsou jednotlivé vzorky odlišných tříd ve vstupním prostoru „promíchány“**

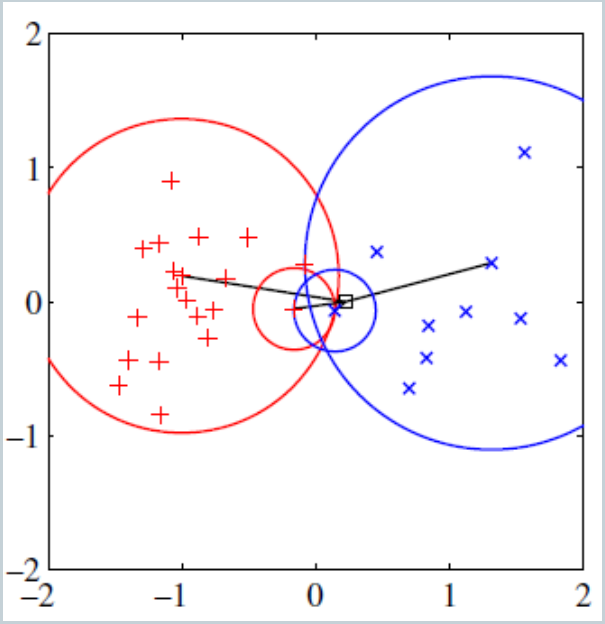
Adaptive distance measure

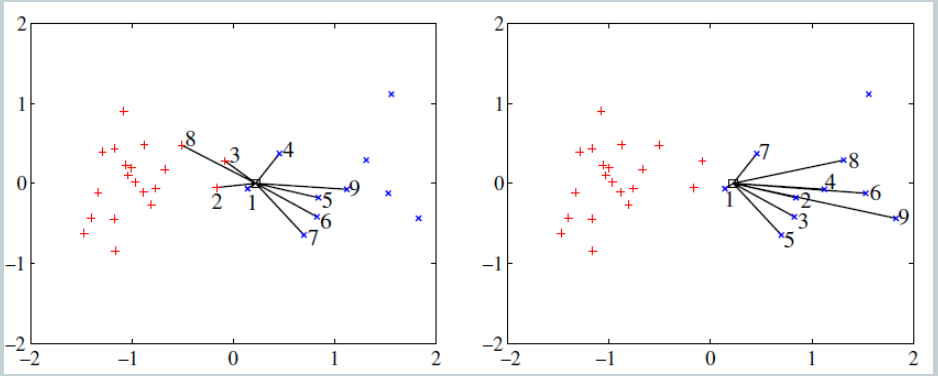
- Úprava pro zlepšení přesnosti klasifikace
- Snaží se omezovat vliv trénovacích vzorků jedné třídy, které jsou „zamíchané“ mezi vzorky jiné třídy
- Přiřazuje jednotlivým trénovacím vzorkům „míru vzdálenosti“, která určuje, jak velký vliv na klasifikaci budou mít
- Probíhá v trénovací fázi algoritmu – nemá negativní vliv na rychlost samotné klasifikace

Princip činnosti úpravy I

- Určení míry vzdálenosti jako poloměru „sféry vlivu“ jednotlivých trénovacích vzorků
- Poloměr – vzdálenost od nejbližšího vzorku s jinou třídou
  - Čím dále je vzorek s jinou třídou – tím větší je „sféra vlivu“ trénovacího vzorku
- Hodnota je pak použita při klasifikaci – dělí výsledek výpočtu vzdálenosti mezi klasifikovaným vzorkem a trénovacími vzorky
  - Čím vyšší hodnota „sféry vlivu“, tím blíže se trénovací prvek jeví

Princip činnosti úpravy II





Pseudokód výpočtu adaptivní míry vzdálenosti

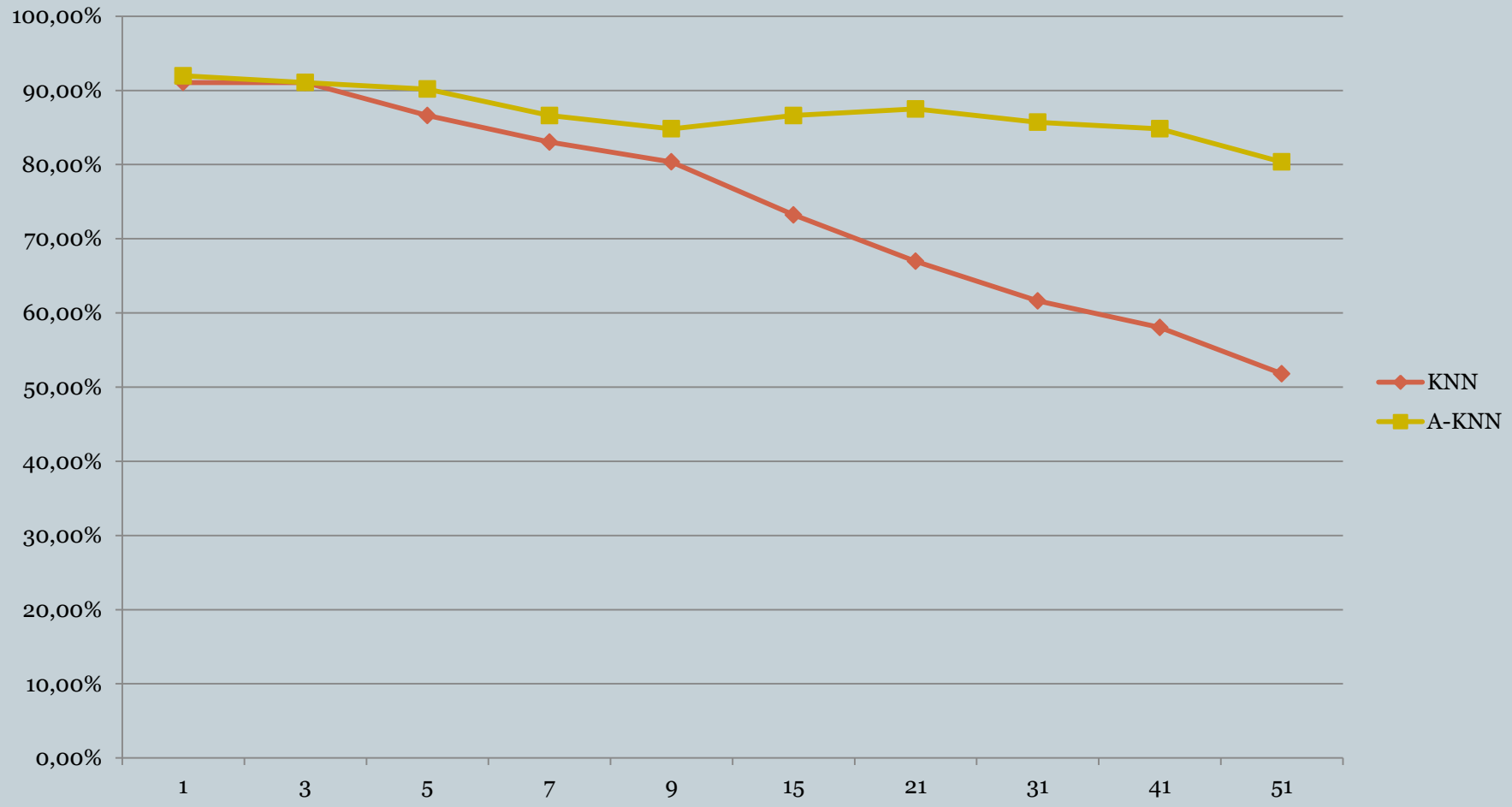
```
BEGIN
        FOR EACH instance1 IN trenovaciMnozina
        DO
                polomer := INFINITE;
                FOR EACH instance2 IN trenovaciMnozina
                DO
                        IF(instance1.trida != instance2.trida) THEN
                            vzdalenost := vypoctiVzdalenost(instance1,
instance 2);
                            IF (vzdalenost < polomer) THEN polomer :=
vzdalenost;
                LOOP

                instance1.polomer := polomer;
        LOOP
END
```

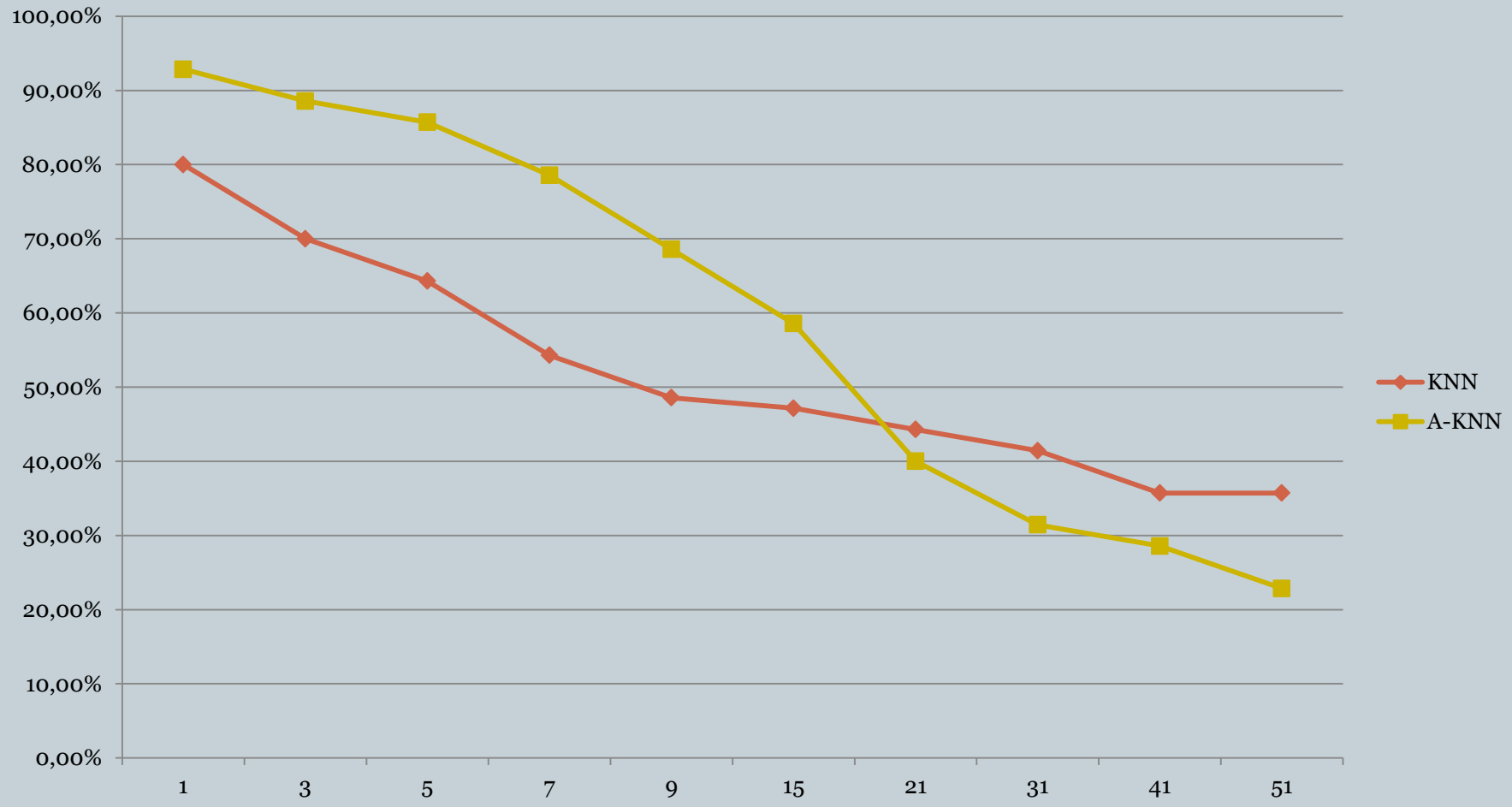
Měření přínosu úpravy

- Vytvoření jednoduchého programu pro klasifikaci příznakových vektorů
- Vstup:
  - Soubor s množinou trénovacích vzorků (vektor + třída)
  - Soubor s množinou testovacích vzorků (vektor)
  - Nastavení parametrů – zvolení vzdálenostní funkce (Euklidovská/Manhattanská), nastavení hodnoty *k* a vypnutí/zapnutí adaptivní míry vzdálenosti
- Výstup:
  - Soubor s množinou testovacích vzorků s klasifikovanou třídou a vyjádřením přesnosti klasifikace v %

Výsledky měření – ukázka #1



Výsledky měření – ukázka #2



Shrnutí

- Přínos v řadě nastaveních nezanedbatelný (někde až přes 25%)
- Teoretický předpoklad – **výraznější přínos pro nižší hodnoty parametru *k*** – potvrzen jen z části (pro jeden ze tří datasetů docházelo k opačnému jevu)
- Úprava nesnižuje rychlost výpočtu klasifikace (pouze dochází k mírně většímu využití paměti – je třeba mít uloženy hodnoty „sféry vlivu“ pro každý trénovací vzorek)

Odkazy

- Zdrojový článek - http://www.sciencedirect.com/science/article/pii/S0167865506001917
- KNN (Wikipedia) - https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- Projekt na GitHub - https://github.com/Vlada47/PRO_semestralka