FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Základy strojového učení a rozpoznávání

Semestrální práce

Barbora Hlubučková

15. 3. 2023

1 Zadání

V mailu jsou potřebná data k řešení semestrální práce. Jedná se o množinu dvojdimenzionálních vektorů. Vaším úkolem je

1) automaticky určit počet tříd
   a) metodou **shlukové hladiny** (automaticky nalezněte hladinu *h*)
   b) metodou **řetězové mapy** (zkuste několik různých počátků)
   c) metodou **MAXIMIN**

   a metody vzájemně porovnat

2) metodou k-means rozdělit všechna data do zjištěného počtu tříd
   - porovnat nerovnoměrné binární dělení s přímým dělením do cílového počtu tříd

3) na výsledné rozdělení dat do jednotlivých tříd z bodu 2 vyzkoušet iterativní optimalizaci

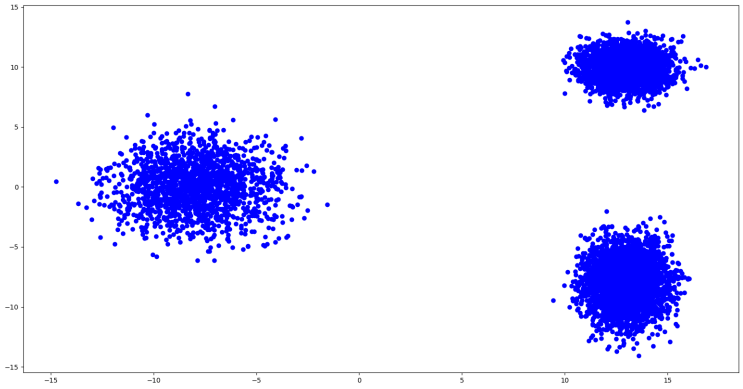
4) na základě informací od učitele (informace o zařazení do jednotlivých tříd ω_i z bodu 2 popřípadě informace z bodu 3) natrénovat
   a) **Bayesův klasifikátor** - tady nepředpokládám explicitně řešení té hranice (kuželosečky) stačí odhadnout parametry jednosložkového normálního rozložení a nějakým dostatečně jemným rastrem ohodnotit body v prostoru (kde se vyskytují trénovací data), kam který bod má největší pravděpodobnost.
   b) **vektorovou kvantizaci** - kde velikost kódové knihy bude rovna počtu zjištěných tříd. Podobně jako v předchozím bodě zakreslete pomocí rastru body v prostoru (trénovacích dat) odpovídající jednotlivým vzorům
   c) **klasifikátor podle nejbližšího souseda** - vyzkoušejte klasifikaci podle jednoho a podle dvou nejbližších sousedů a podobně jako v předchozím bodě zakreslete pomocí rastru body v prostoru (trénovacích dat), které klasifikujeme do jednotlivých tříd
   d) **klasifikátor s lineárními diskriminačními funkcemi** - porovnejte potřebný počet iterací při použití Rosenblattova alg., metody konstantních přírůstků a upravené metody konstantních přírůstků pro několik zvolených konstant učení. Podobně jako v předchozím bodě zakreslete pomocí rastru body v prostoru (trénovacích dat), které klasifikujeme do jednotlivých tříd a které případně klasifikovat nemůžeme (diskutujte proč)

Úlohu můžete řešit v libovolném jazyce. Ale pokud zvolíte nestandardní řešení (mimo C++, MATLAB, Python), tak mě musíte před odevzdáním informovat, abych se na vás připravil.

Výsledky a postup prezentujte krátkou zprávou - záměrně jsem zvolil dimenzi 2, aby se řešení úloh dalo pěkně zobrazit (očekávám spoustu obrázků).

2 Data

Zadaný dataset obsahuje 8044 dvoudimenzionálních vektorů, které jsou po zobrazení v grafu rozdělené do třech shluků, jak je vidět na obrázku 1.



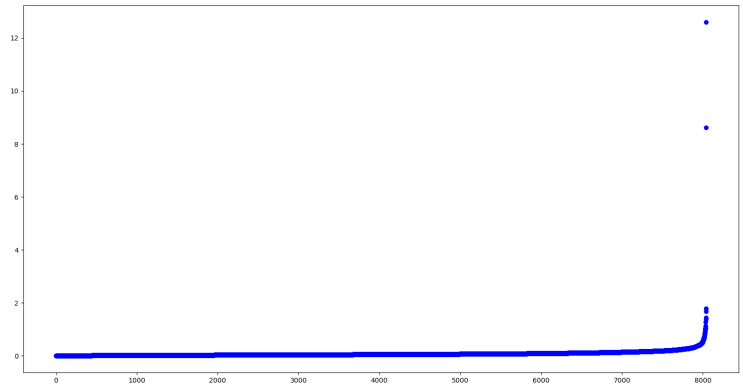
Obrázek 1: Zobrazení zadaného datasetu

3 Automatické určení počtu tříd

3.1 Metoda shlukové hladiny

Metoda shlukové hladiny je aglomerativní algoritmus, který přiřazuje celé množině obrazů T postupně posloupnost rozkladů B_0 až B_{N-1} a ke každému vzniklému shluku T_i přiřazuje shlukovací hladinu h.

V algoritmu metody nejprve rozdělíme všechna trénovací data zvlášť do shluků a následně v každém kroku hledáme dvojici shluků, které jsou k sobě nejblíže. Ty pak sjednotíme. Na konci algoritmu jsou pak všechna data v jediném shluku.

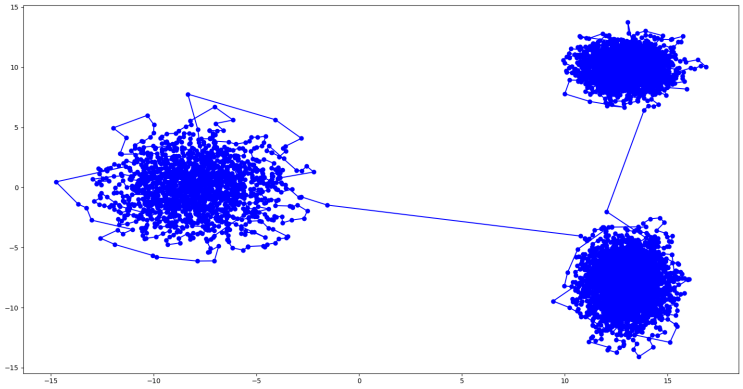


Obrázek 2: Metoda shlukové hladiny

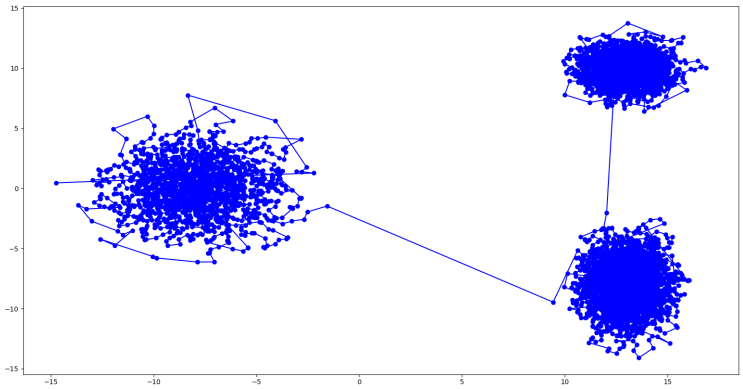
Na obrázku 2 jsme na základě průběhu shlukovací hladiny odhadli počet shluků v datech.

3.2 Metoda řetězové mapy

Metoda řetězové mapy je divnizní algoritmus. Nejprve jsme z množiny vybrali libovolná startovací obraz a k němu jsme pak hledali nejbližšího souseda. Další položka bude pak nejbližší soused k předchozímu, tu vybíráme z námi přeuspořádané trénovací množiny, kterou si náš algoritmus musí pamatovat. Tento proces pokračuje dokud se neuspořádají všechny obrazy trénovací množiny.



Obrázek 3: Řetězová mapa $\begin{bmatrix} -6.440836 & 1.586298 \end{bmatrix}$



Obrázek 4: Řetězová mapa $\begin{bmatrix} 11.60267 & 9.405461 \end{bmatrix}$

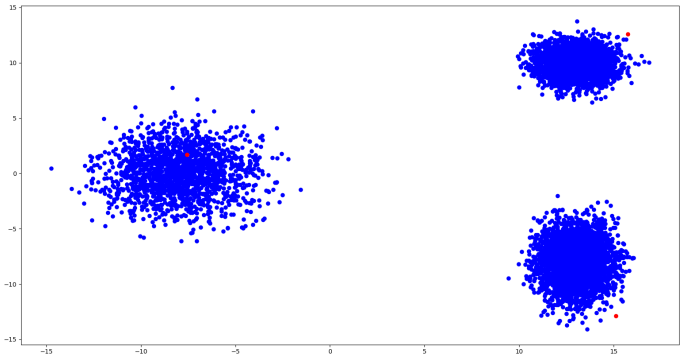
Na obrázku 3 a na obrázku 4 jsou řetězové mapy, s počátečními body $\begin{bmatrix} -6.440836 & 1.586298 \end{bmatrix}$ a $\begin{bmatrix} 11.60267 & 9.405461 \end{bmatrix}$. Dlouhé vzdálenosti mezi shluky označují změnu třídy. Vyzkoušeli jsme si tedy řešení pro dva různé počátky se stejným výsledkem, tedy třemi shluky.

3.3 Metoda MAXIMIN

Metoda MAXIMIN je jednoduchý heuristický algoritmus, který jsme využili pro určení počtu shluků.

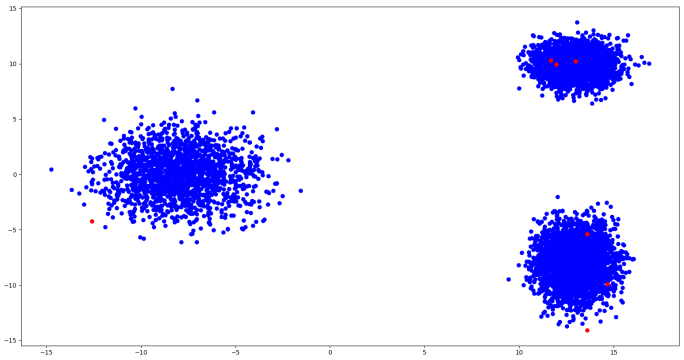
Nejprve jsme z trénovací množiny vybrali náhodně počáteční bod, který jsme označili za střed prvního shluku. K tomuto bodu jsme následně nalezli nejvzdálenější bod a ten jsme označili za střed druhého shluku. Dále jsme pro každý obraz trénovací množiny vyčíslili vzdálenost k předchozím středům a uchovali jsme tu minimální. Z minimálních vzdáleností jsme následně vybrali maximální a porovnali jsme ji s velikostí vzdáleností $q-$násobku vzdáleností mezi středy. Pokud je maximální vzdálenosti větší, zavedeme nový shluk s novým středem. Tak pokračujeme, dokud se nová maximální vzdálenost nedostane do sporu s podmínkou pro vytvoření nového shluku.

Metodu MAXIMIN jsme vyzkoušeli pro různé volby konstanty q. Nejprve jsme zvolili $q = 1$ jak lze pozorovat na obrázku 5, počáteční bod střed v $\begin{bmatrix} -7.572466 & 1.702891 \end{bmatrix}$ a zbylé dva středy se nacházeli v $\begin{bmatrix} 15.11274 & -12.89537 \end{bmatrix}$ a $\begin{bmatrix} 15.7576 & 12.58255 \end{bmatrix}$. Středy jsou na obrázku znázorněny červenou barvou.

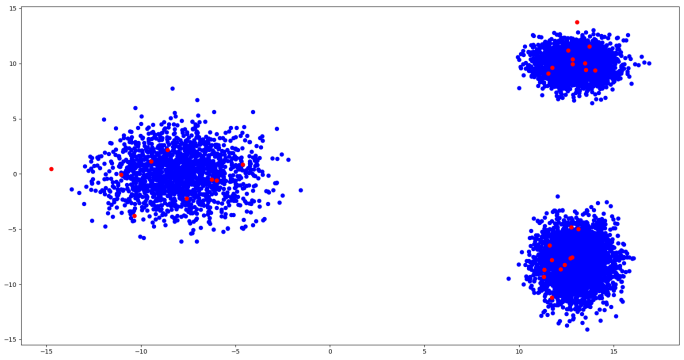


Obrázek 5: MAXIMIN pro $q = 1$

Dále jsme vyzkoušeli změnit konstantu $q = 0.5$ viz obrázek 6. Počáteční střed se nacházel v bodě $\begin{bmatrix} 11.67687 & 10.31263 \end{bmatrix}$. V tomto případě se nalezlo sedm středů. Zmenšení konstanty q tedy vede na nepřesnější výsledek, což jsme ověřili na obrázku 7, kdy jsme q ještě více zmenšili na hodnotu 0.09 a nalezli jsme třicet středů. Opět jsme znázornili na středy červenou barvou.

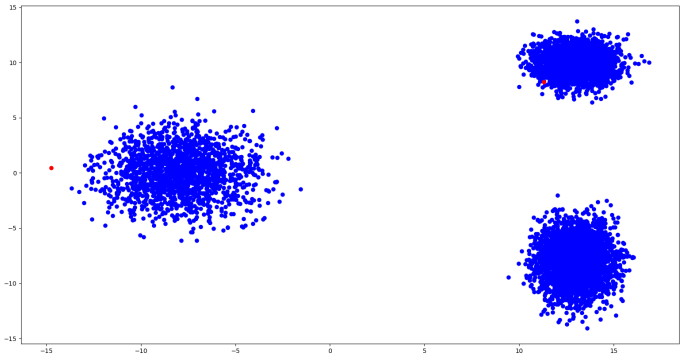


Obrázek 6: MAXIMIN pro $q = 0.5$



Obrázek 7: MAXIMIN pro $q = 0.09$

Jako poslední jsme zvolili konstantu $q = 1.9$. Výsledek je na obrázku 8. V tomto případě jsme kvůli zvětšení q nalezli pouze dva středy.



Obrázek 8: MAXIMIN pro $q = 1.9$

3.4 Porovnání metod

Metoda řetězové mapy se mi zdála nejlepší, ale je výrazně výpočetně pomalejší. V případě metody MAXIMIN by bylo možné vyzkoušet velké množství pokusů za stejný čas, ale je závislá na volbě konstanty q. Metoda shlukové hladiny je také pomalejší, ale její výsledky jsou stejně kvalitní, jako v případě řetězové mapy a není nutné se spoléhat na dobrou volbu počátečního prvku.

4 Metoda k-means

V předchozím bodu jsme zjistili cílový počet tříd $R = 3$. Tuto znalost využijeme při řešení $k-$means algoritmu, kdy data rozdělíme do cílového počtu tříd. Minimalizujeme tedy celkový ukazatel kvality J

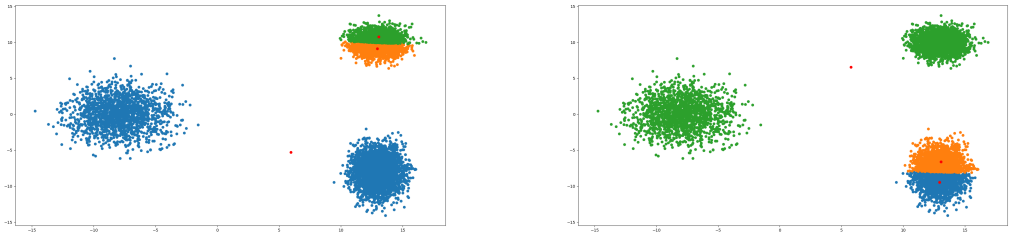
$$J = \sum_{i=1}^{R} J_i = \sum_{i=1}^{R} \sum_{x_i} d^2(\mathbf{x}, \boldsymbol{\mu}_i) = \sum_{i=1}^{R} \sum_{x_i} ||\mathbf{x} - \boldsymbol{\mu}_i||^2$$

kde J_i je hodnota kritéria $i-$tého shluku v $k-$tém kroce

4.1 Přímé dělení do cílového počtu tříd

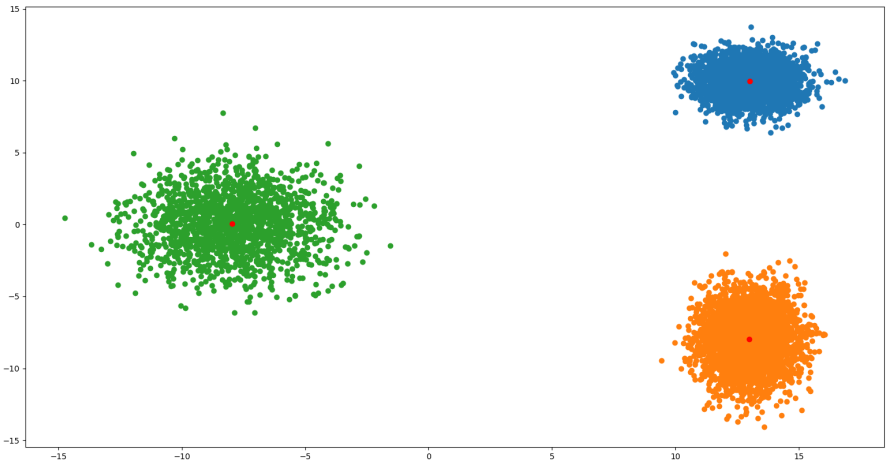
V algoritmu k-means využijeme znalosti o cílovém počtu tříd a libovolně vybereme středy shluků s počtem $R = 3$. Dále jsme rozdělili všechny body množiny T do R shluků podle minimální vzdálenosti ke střední hodnotě. Střední hodnoty jsme následně přepočítali. Pokud se neposunou, algoritmus končí.

Volba počátečních středů je pro algoritmus klíčová, protože není zaručeno dosáhnutí globálního minima. Na obrázku 9 jsou dvě vykreslení chybného k-means algoritmu. Pro obrázek vlevo bylo potřeba 22 kroků a pro obrázek vpravo 14 kroků.



Obrázek 9: Chybný k-means

Na obrázku 10 je zobrazen správný k-means po třech krocích. Výsledek tedy závisí na zvolení středních hodnot, které jsme v našem případě volili libovolně.

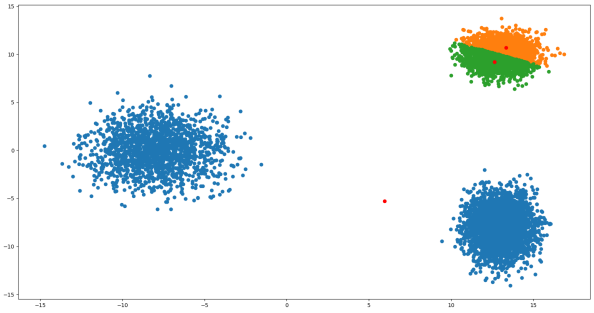


Obrázek 10: Správný k-means

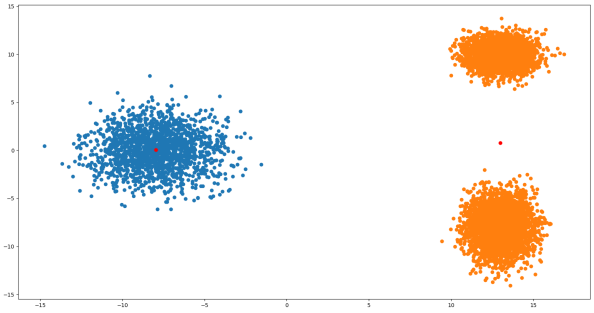
4.2 Nerovnoměrné binární dělení

V algoritmu nerovnoměrného binárního dělení využijeme již vytvořeného k-means a rozdělíme data do dvou shluků. Hledám, který shluk přispívá na zkreslení největší mírou a ten následně rozdělím. Po každém kroku musím následně zjistit, zda jsme nedosáhli cílového počtu shluků.

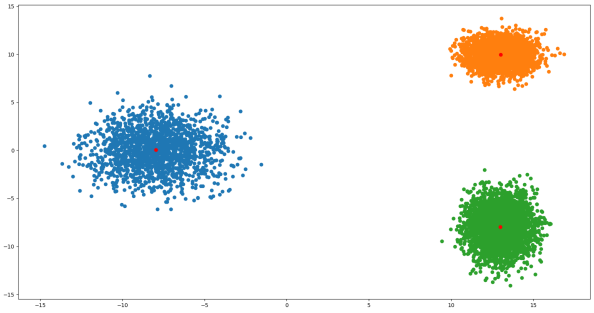
Na obrázcích 11-13 lze vidět průběh algoritmu. Červeně jsou opět vyznačeny středy.



Obrázek 11: k-means



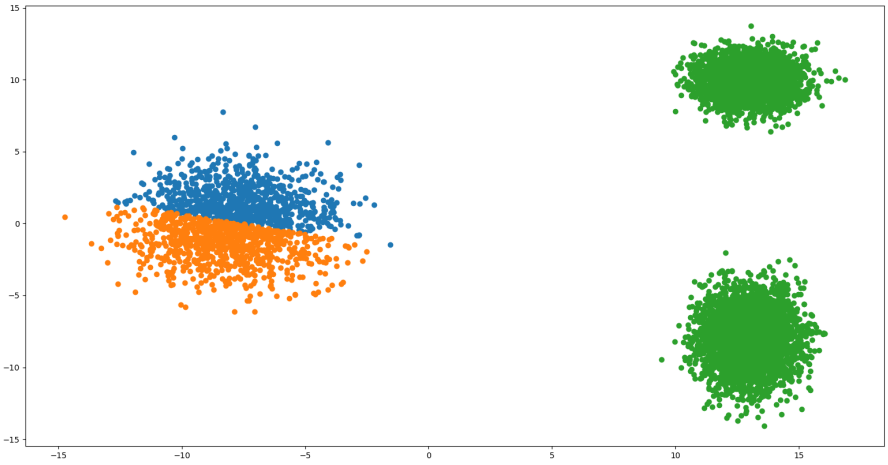
Obrázek 12: Nerovnoměrné binární dělení - rozdělení do dvou shluků



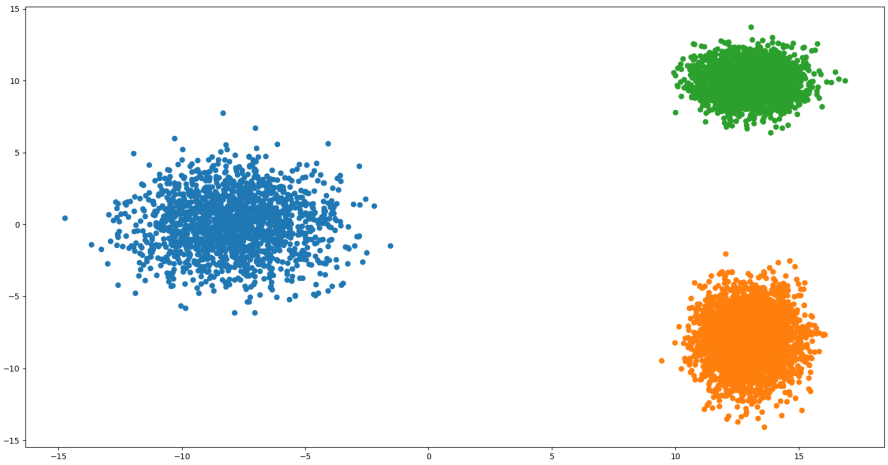
Obrázek 13: Nerovnoměrné binární dělení

5 Iterativní optimalizace

Iterativní optimalizace je algoritmus, která v našem případě vylepšuje minimalizací kriteriální funkce algoritmus k-means.



Obrázek 14: Původní rozdělení k-means - chybné



Obrázek 15: Iterativní optimalizace - správné

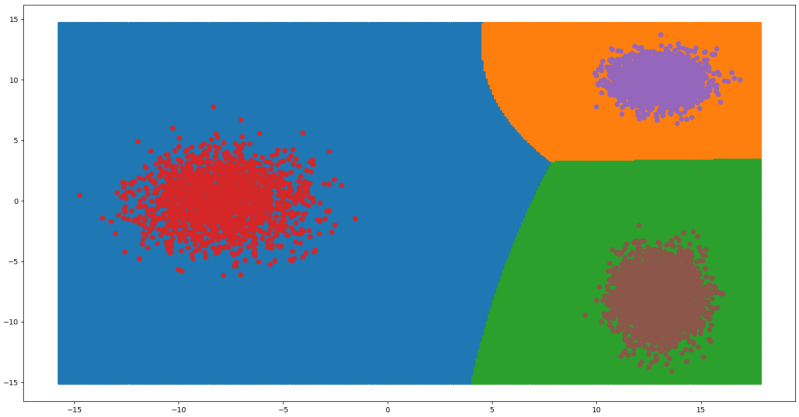
6 Klasifikátory

V této části se budeme zabývat trénováním klasifikátorů. Z předchozích sekcí jsme získali informace o rozdělení a středních hodnotách tříd, které jsme využili pro natrénování klasifikátorů. Před natrénováním jsme si vytvořili rastr.

6.1 Bayesův klasifikátor

Bayesův klasifikátor přiřadí body třídě podle pravděpodobnostní funkce normálního rozdělení každé ze tříd. Pomocí předem získaných tříd a středních hodnot určíme kovarianční matici.

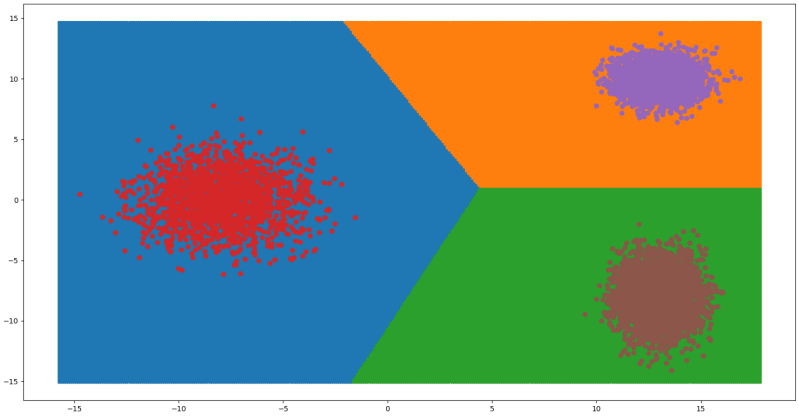
Každému bodu pak určíme, jaké třídě bude náležet podle největší pravděpodobnosti. Na obrázku 16 je zobrazeno výsledné rozdělení.



Obrázek 16: Bayesův klasifikátor

6.2 Vektorová kvantizace

Vektorová kvantizace klasifikuje prostor pomocí kódových vektorů. Bod rastru se přiřazuje shluku, k jehož středu je nejblíže. Metoda rozdělila prostor na tři části, jak lze pozorovat na obrázku 17.

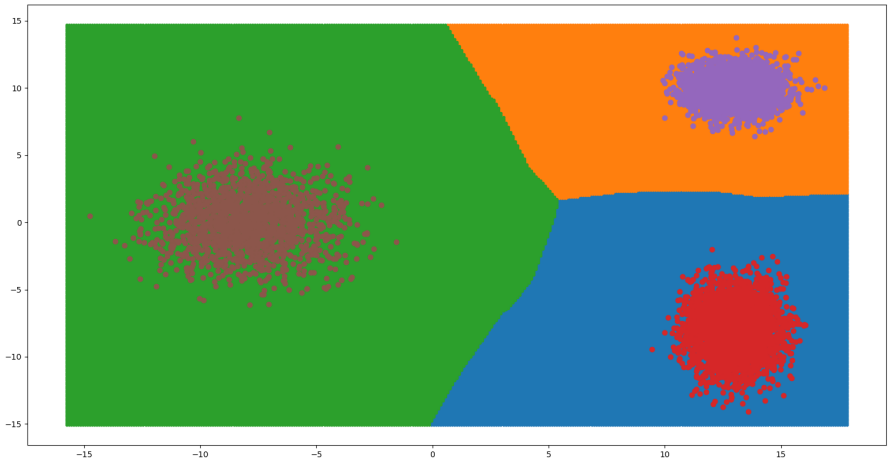


Obrázek 17: Vektorová kvantizace

6.3 Klasifikátor podle nejbližšího souseda

6.3.1 Klasifikace podle jednoho nejbližšího souseda

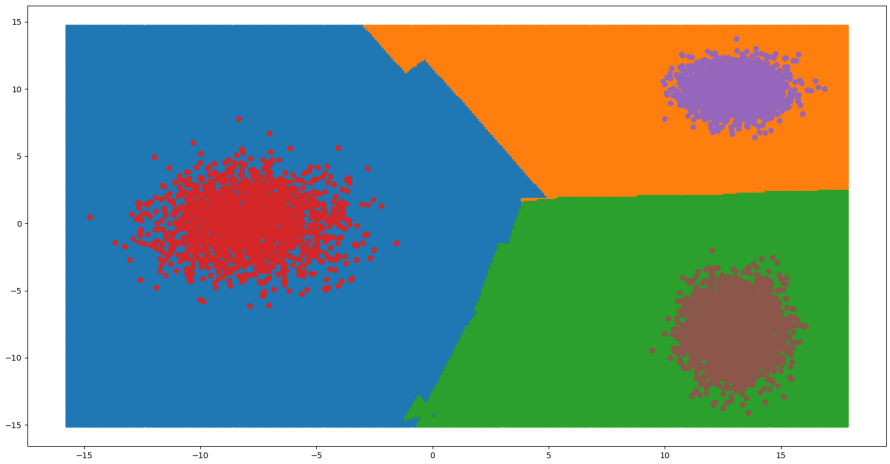
Klasifikace podle jednoho nejbližšího souseda hledá etalon, který má nejblíže ke každému bodu rastru. Tento bod pak zařadí do třídy, ve které etalon je. Tato klasifikace je časově náročnější než vektorová kvantizace a Bayesův klasifikátor.



Obrázek 18: Klasifikátor podle nejbližšího souseda

6.3.2 Klasifikátor podle dvou nejbližších sousedů

V případě klasifikace podle dvou nejbližších sousedů postupujeme stejně jako v předchozím případě, jen hledáme dva nejbližší sousedy. Body pak zařazujeme podle toho, kam dva nejbližší sousedi patří. Toto rozdělení lze pozorovat na obrázku 19. Tento klasifikátor je stejně jako klasifikátor podle nejbližšího souseda náročnější.



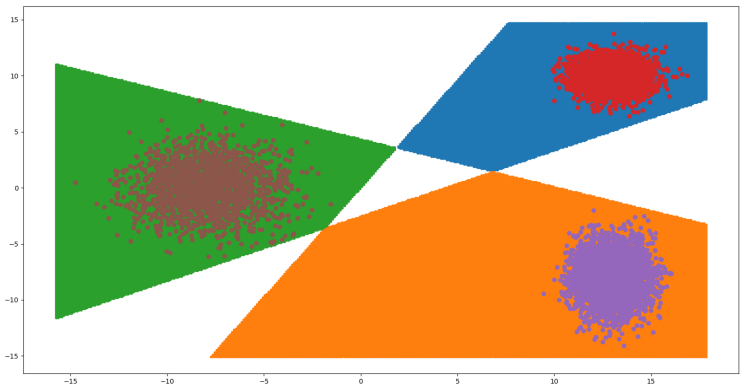
Obrázek 19: Klasifikátor podle dvou nejbližších sousedů

6.4 Klasifikátor s lineárními diskriminačními funkcemi

Abychom mohli využít klasifikátor s lineárními diskriminačními funkcemi, měly by být třídy lineárně separovatelné. Tento předpoklad náš dataset splňuje, protože jsou od sebe třídy dostatečně vzdálené a data jsou kompaktní. Klasifikátor hledá nastavení diskriminační funkce $g(x)$. Při rozdělování prostoru dochází k vytváření oblastí, které nelze klasifikovat.

6.4.1 Rosenblattův algoritmus

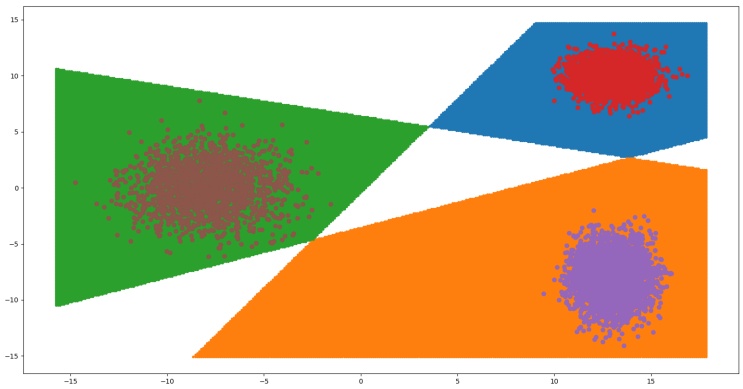
Rosenblattův algoritmus je jeden z nejjednodušších lineárních klasifikátorů. Konstanta učení je $c_k = 1$ a pásmo necitlivosti $\delta = 0$. Vykreslení Rosenblattova algoritmu pozorujeme na obrázku 20.



Obrázek 20: Rosenblattův algoritmus

6.4.2 Metoda konstantních přírůstků

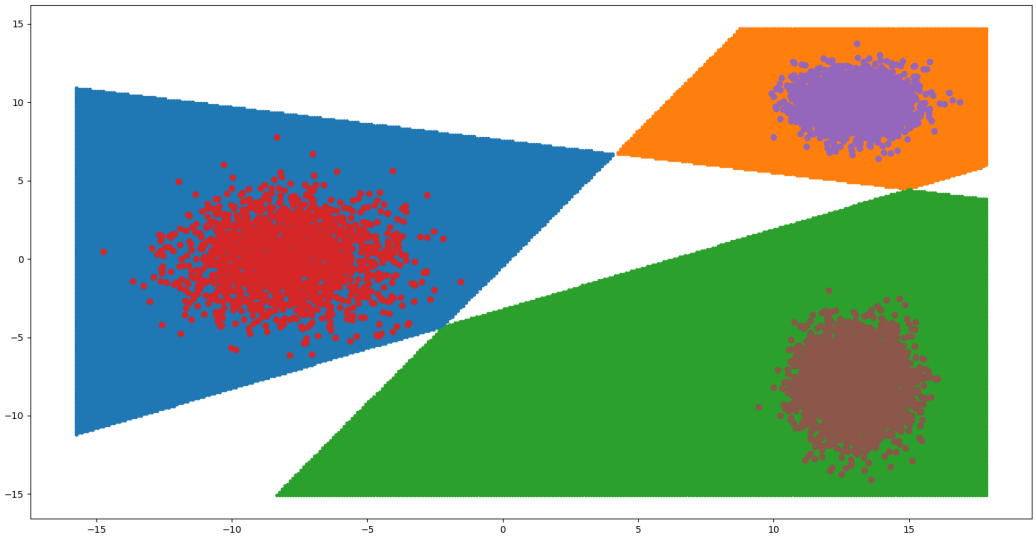
V metodě konstantních přírůstků je změněna hodnota konstanty učení $c_k = \frac{\beta}{||x(k)||^2}$. Tento parametr tlumí přírůstek, což má za následek zvýšení počtu iteračních kroků k dosažení správné diskriminační funkce. Parametry jsme zvolili $\beta = 1$ a $\delta = 1$. Vykreslení metody konstantních přírůstků je na obrázku 21.



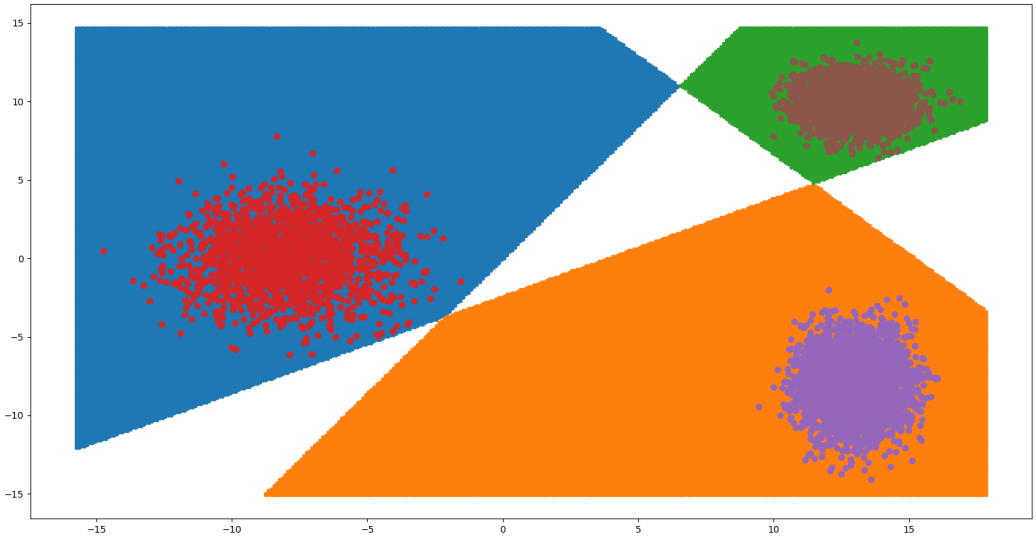
Obrázek 21: Metoda konstantních přírůstků

6.4.3 Upravená metoda konstantních přírůstků

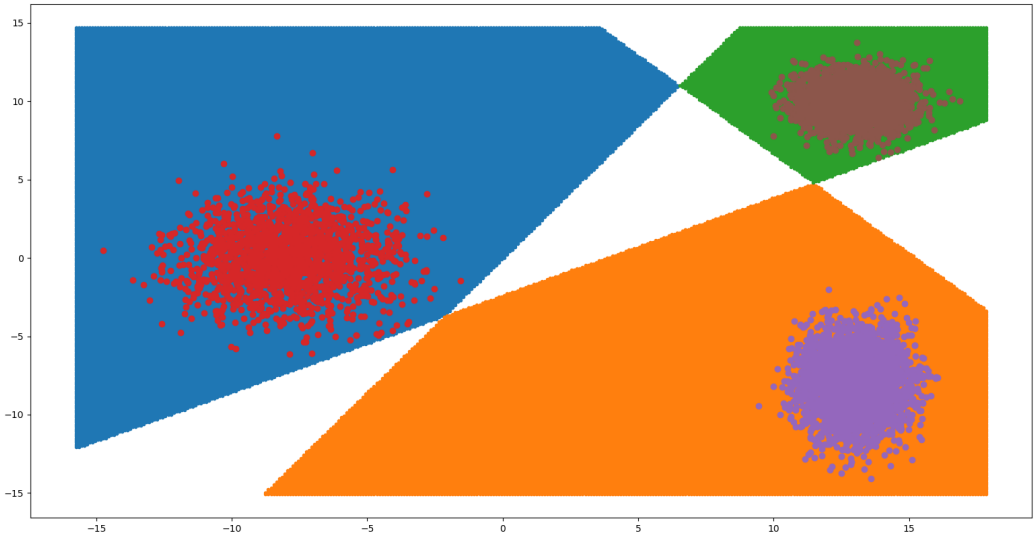
Upravená metoda konstantních přírůstků je téměř shodná s neupravenou metodou konstantních přírůstků. Rozdíl je při přepočítávání parametru c_k, který se opakuje tolikrát, abychom dosáhli správné klasifikace. upravená metoda je na obrázku 22-27 pro různé počáteční nastavení β a δ .



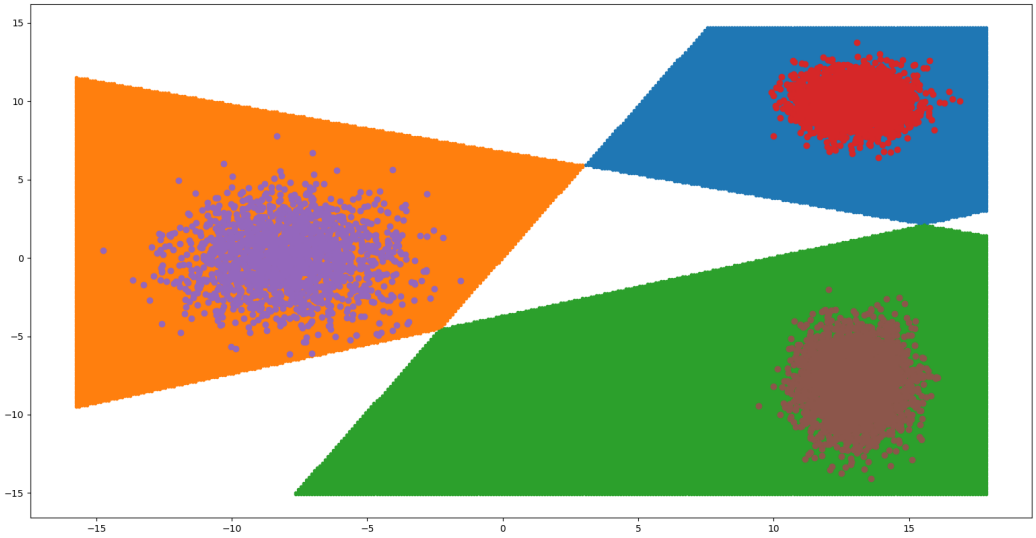
Obrázek 22: Upravená metoda konstantních přírůstků $\beta = 1$ $\delta = 1$



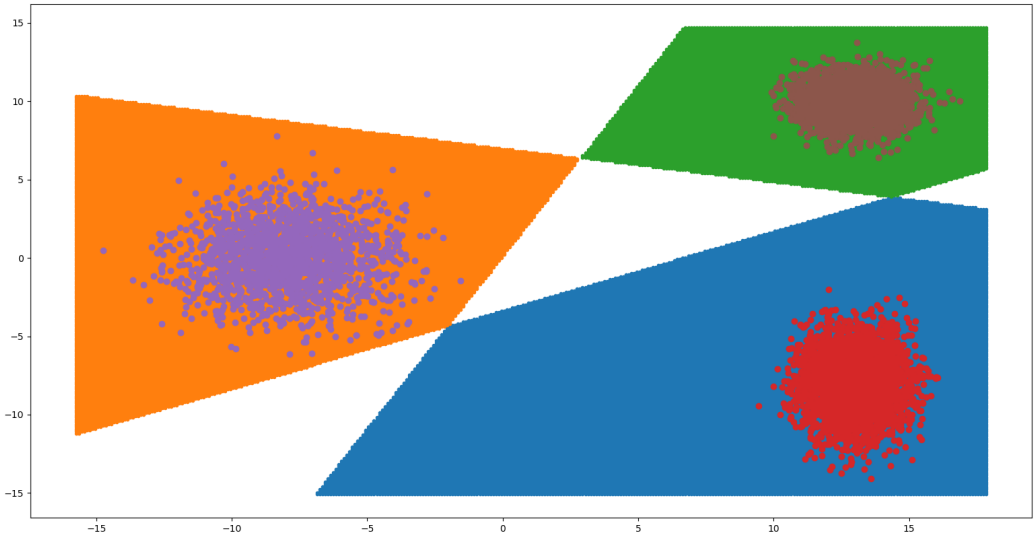
Obrázek 23: Upravená metoda konstantních přírůstků $\beta = 5$ $\delta = 1$



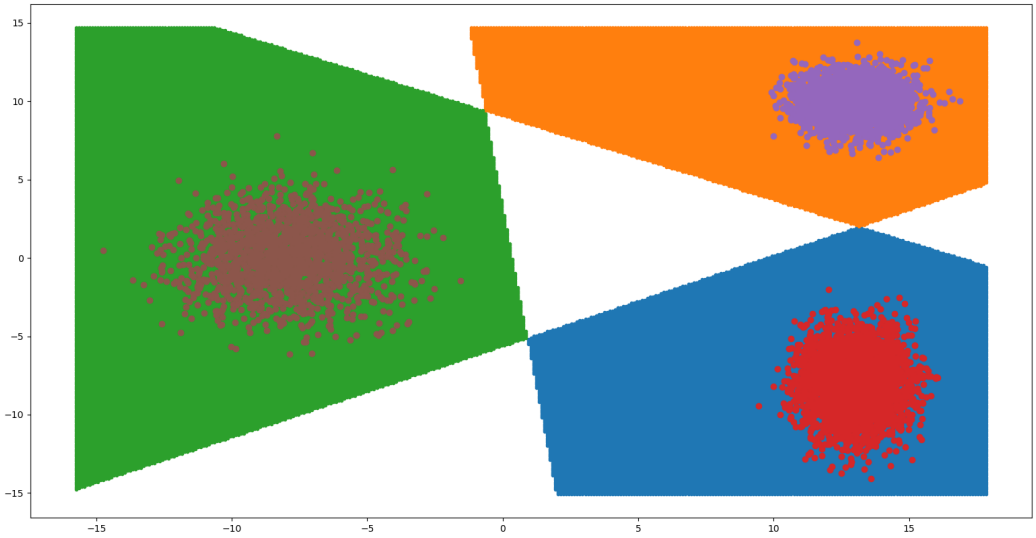
Obrázek 24: Upravená metoda konstantních přírůstků $\beta = 2$ $\delta = 1$



Obrázek 25: Upravená metoda konstantních přírůstků $\beta = 8$ $\delta = 1$



Obrázek 26: Upravená metoda konstantních přírůstků $\beta = 15$ $\delta = 15$



Obrázek 27: Upravená metoda konstantních přírůstků $\beta = 1$ $\delta = 8$

7 Závěr

Semestrální práce se skládala ze čtyř částí.

V první části se nám podařilo automaticky určit počet tříd metodou shlukové hladiny, metodou řetězové mapy a metodou MAXMIN. Metody jsme vzájemně porovnali.

Ve druhé a třetí části jsme metodou k-means rozdělili všechna data do zjištěného počtu tříd. Následně jsme porovnali nerovnoměrné binární dělení a dělení do cílového počtu tříd. Na výsledné rozdělení dat jsme pak vyzkoušeli iterativní optimalizaci.

Čtvrtá část semestrální práce se zabývala klasifikátory. Konkrétně jsme trénovali Bayesův klasifikátor, následovala vektorová kvantizace, klasifikátor podle nejbližšího souseda, kdy jsme si vyzkoušeli klasifikaci podle jednoho a podle dvou nejbližších sousedů a klasifikátor s lineárními diskriminačními funkcemi. Porovnali jsme Rosenblattův algoritmus, metodu konstantních přírůstků a upravenou metodu konstantních přírůstků.