SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE

Stavebná fakulta

Evidenčné číslo: SvF-5343-81238

PRIRODZENÉ SIETE HLBOKÉHO UČENIA NA BÁZE DOPREDNO-SPÄTNEJ NELINEÁRNEJ DIFÚZIE PRE KLASIFIKÁCIU BIOTOPOV NATURA 2000

Diplomová práca

Študijný program:	Matematicko-počítačové modelovanie
Študijný odbor:	Matematika
Školiace pracovisko:	Katedra matematiky a deskriptívnej geometrie
Vedúci záverečnej práce:	prof. RNDr. Karol Mikula, DrSc.
Konzultant:	Ing. Michal Kollár, PhD.

Bratislava 2020 **Bc. Aneta Alexandra Ožvat**

Slovenská technická univerzita v Bratislave
Katedra matematiky a deskriptívnej geometrie

Stavebná fakulta
Akademický rok: 2019/2020
Evidenčné číslo: SvF-5343-81238

STU SvF

ZADANIE DIPLOMOVEJ PRÁCE

Študentka: **Bc. Aneta Alexandra Ožvat**
ID študenta: 81238
Študijný program: matematicko-počítačové modelovanie
Študijný odbor: matematika
Vedúci práce: prof. RNDr. Karol Mikula, DrSc.
Konzultant: Ing. Michal Kollár, PhD.

Názov práce: **Prirodzené siete hlbokého učenia na báze dopredno-spätnej nelineárnej difúzie pre klasifikáciu biotopov Natura 2000**

Jazyk, v ktorom sa práca vypracuje: slovenský jazyk

Špecifikácia zadania:

V práci budú vytvorené nové metódy pre kontrolované hlboké učenie (supervised deep-learning) na báze riešenia dopredno-spätných nelineárnych difúznych rovníc na topologických grafových štruktúrach s aplikáciou na klasifikáciu biotopov Natura 2000.

Riešenie zadania práce od: 01. 10. 2019
Dátum odovzdania práce: 21. 05. 2020

Bc. Aneta Alexandra Ožvat
študentka

prof. RNDr. Radko Mesiar, DrSc.
vedúci pracoviska

prof. RNDr. Karol Mikula, DrSc.
garant študijného programu

Čestné vyhlásenie

Vyhlasujem, že som diplomovú prácu „Prirodzené siete hlbokého učenia na báze dopredno-spätnej nelineárnej difúzie pre klasifikáciu biotopov Natura 2000“ vypracovala samostatne s použitím citovanej literatúry a s odbornou pomocou vedúceho práce a konzultanta.

18. mája 2020 vlastnoručný podpis

.....

Poďakovanie

Týmto spôsobom by som sa chcela poďakovať vedúcemu prof. RNDr. Karolovi Mikulovi, DrSc. za hodnotné rady, usmerňovanie pri realizácii diplomovej práce a predovšetkým za trpezlivosť a ochotu. Moja vďaka patrí aj Ing. Michalovi Kollárovi, PhD. za cenné rady a ochotu pri vytváraní algoritmu a jeho implementácií. Zároveň sa chcem poďakovať aj rodine a blízkym za neustálu podporu a pochopenie.

Súhrn

V diplomovej práci sa zaoberáme odvodením numerického modelu nelineárnej dopredno-spätnej difúzie na neorientovanom úplnom grafe pre potreby klasifikácie biotopov Natura 2000. Naším cieľom je vytvoriť a implementovať algoritmus, ktorý bude automaticky zaraďovať nové pozorovania k príslušným biotopom. V práci budeme klasifikovať vysegmentované chránené oblasti biotopov iba na základe ich čŕt, pomocou kontrolovaného hlbokého učenia. Najprv naučíme algoritmus klasterizovať neoznačené dáta a následne prejdeme na klasifikáciu už označených dát. Proces učenia sme rozdelili na dve fázy, na trénovacú fázu, v ktorej naučíme algoritmus klasifikovať trénovacie dáta, a na validačnú fázu, v ktorej budeme klasifikovať dáta, ktoré algoritmus doposiaľ nevidel. Nakoniec vypočítame relevantnosť úspešnej klasifikácie dát.

Kľúčové slová

Difúzia na grafe, analýza hlavných komponentov, hlboké strojové učenie, klasterizácia, klasifikácia, Natura 2000

Abstract

In this diploma thesis, we deal with the derivation of the numerical model of the forward-backward nonlinear diffusion on the undirected complete graph for classification of Natura 2000 habitats. Our goal is to create and implement an algorithm, which will automatically classify the new observations to the corresponding habitats. We will classify segmented protected areas of habitats only on the basis on their features, using the supervised deep-learning. First the algorithm will learn how to cluster unlabeled data and then the algorithm will begin with learning how to classify the labeled data. The process of learning is divided into two phases, into training phase, in which the algorithm will learn how to classify training data, and into validation phase, in which we will classify the data, which were unknown to the algorithm. At the end, we will define and calculate the coefficient of relevancy of the classified data.

Keywords

Diffusion on graph, principal component analysis, deep-learning, clustering, classification, Natura 2000

Obsah

Kapitola 1

Motivácia

Práca sa zaoberá tvorbou algoritmov na klasifikáciu environmentálnych a botanických dát s cieľom automatického zaradenia nových pozorovaní do príslušnej skupiny. „Manuálne“ zaraďovanie nových dát do skupín je v tomto prípade časovo aj finančne náročná úloha, pretože vyžaduje terénny výskum, a často sa môže stať, že rozpoznanie dát a ich priradenie je zaťažené subjektívnou chybou. Preto sme sa rozhodli vytvoriť algoritmus, ktorý bude schopný, na základe podstatných vlastností (čŕt) dát, samostatne zaradiť nové dáta do skupín, ktoré sú im najviac podobné. Na charakterizovanie dát, využívame informácie z optických kanálov družice Sentinel-2 [3].

Dáta, s ktorými pracujeme, sú odborníkmi, z oblasti botaniky a ochrany prírody, označené chránené oblasti biotopov Natura 2000. Natura 2000 je systém chránených oblastí na území Európskej únie s cieľom chrániť prírodné dedičstvo. Táto sústava chránených území má zabezpečiť ochranu najvzácnejších a najviac ohrozených druhov voľne rastúcich rastlín, voľne žijúcich živočíchov a prírodných biotopov vyskytujúcich sa na území štátov Európskej únie a prostredníctvom ochrany týchto druhov a biotopov zabezpečiť zachovanie biologickej rôznorodosti v celej Európskej únii [15].

Našim cieľom je, po vypočítaní rôznych čŕt (features) označených chránených oblastí biotopov, vybrané oblasti správne roztriediť. Najskôr navrhneme metódu na triedenie dát do vopred neurčených skupín. Algoritmus, ktorý vytvoríme, nielenže sám zoskupí dáta do skupín, ale si aj sám vytvorí skupiny na základe charakteristík, ktoré si vypočíta z dát. Takýto spôsob zoskupovania dát, keď nezadávame druh skupiny a ani počet

skupín, je označovaný ako nekontrolované učenie (unsupervised learning), konkrétne ide o klasterizáciu, keďže dáta zoskupujeme do skupín - klastrov (clusters). Druhý spôsob triedenia, pre ktorý v práci vytvárame algoritmus, je klasifikácia, ktorá sa radí do triedy kontrolovaného učenia (supervised learning). Algoritmu, spolu zo vstupnými dátami, dodáme aj informáciu o počte a druhu skupín. Následne algoritmus má za úlohu zo vstupných dát vytvoriť skupiny a v prípade ak existujú hodnoty, ktorým nie je priradená žiadna zo skupín, teda nové pozorovania, na základe charakteristík, ktoré si algoritmus vypočíta z dát, zaradí novú hodnotu do jednej z už existujúcich skupín.

Algoritmus, ktorý budeme v práci prezentovať, predstavuje takzvanú sieť hlbokého učenia (deep-learning network). Za hlbokým učením (deep-learning) sa zvyčajne skrýva umelá neurónová sieť (artificial neural network) s veľkým množstvom skrytých vrstiev. Našu sieť môžeme označiť ako sieť hlbokého učenia, pretože jedna skrytá vrstva zodpovedá jednému časovému kroku riešenia nelineárnych difúznych rovníc na úplných grafoch a, ako uvidíme ďalej v práci, vždy pracujeme s väčším počtom časových krokov. Keďže naša sieť pri svojej konštrukcii nevyužíva umelé neurónové siete, ale numerickú aproximáciu dopredno-spätných difúznych rovníc, čo sa nám zdá prirodzený postup pre klasterizáciu a klasifikáciu dát pomocou hlbokého učenia, nazývame ju „prirodzenou sieťou“ (natural network) na odlíšenie od „umelej neurónovej siete“, ktorej konštrukcia využíva iné princípy. V našej práci sme sa tiež inšpirovali prácou [6], v ktorej E. Haber a L. Ruthotto ukázali vzťah medzi úspešným modelom hlbokého učenia, takzvanou reziduálnou neurónovou sieťou (Residual Neural Network - ResNet) [7] a numerickým riešením systémov obyčajných diferenciálnych rovníc doprednou Eulerovou metódou. Následne navrhli parabolické a hyperbolické siete pre hlboké učenie na báze riešenia príslušných parciálnych diferenciálnych rovníc. My v našej práci uvažujeme nelineárne dopredno-spätné difúzne rovnice ako nový nástroj klasifikácie pomocou kontrolovaného hlbokého učenia (supervised deep-learning). V prípade hlbokého učenia, a teda aj v našom prípade, sa snažíme v učiacej fáze získať (naučiť) optimálne parametre siete. Keď si to povieme matematicky je to ekvivalentné problému odhadovania parametrov modelu [1]. V našej práci vytvorený algoritmus je teda založený na numerickom riešení systémov nelineárnych parciálnych diferenciálnych rovníc, pri ktorom sa snažíme vykonať čo najpresnejšie triedenie dát pri optimálnych parametroch siete a za optimálny výpočtový čas.

Kapitola 2

Matematický model a numerická aproximácia

2.1 Tvorba matematického modelu

Definujeme si graf G ako usporiadanú dvojicu $G = (V(G), E(G))$, kde $V(G)$ je konečná množina vrcholov a $E(G)$ je množina dvojprvkových podmnožín množiny $V(G)$, ktorá predstavuje hrany grafu [8]. Počet vrcholov grafu G označíme ako N_V. Uvažujme, že graf G je úplný graf, teda všetky vrcholy $v \in V(G)$, sú medzi sebou prepojené hranou. Navyše graf G je neorientovaný graf a teda hrany neorientovaného grafu nemajú danú orientáciu.

2.1.1 Difúzia na grafoch

Uvažujme skalárnu funkciu f, ktorá bude predstavovať hodnotu vo vrchole grafu G. Potom difúziu skalárnej funkcie f vieme vyjadriť v podobe parabolickej PDR v tvare

$$\partial_t f = \nabla \cdot (g \nabla f), \tag{2.1}$$

kde g reprezentuje difúzny koeficient. V prípade ak by sme uvažovali $g = 1$ v rovnici (2.1), dostali by sme rovnicu, ktorá zodpovedá rovnici lineárnej difúzie, v tvare

$$\partial_t f = \Delta f, \qquad t \in [0, T]. \tag{2.2}$$

Rovnicu (2.2) uvažujeme s počiatočnou podmienkou $f(v,0) = f^0(v)$, kde $v \in V(G)$. Okrajové podmienky nám v tomto prípade netreba definovať, pretože v úplnom neorientovanom grafe prebieha difúzia medzi všetkými vrcholmi grafu.

Definujme si vzdialenosť vrcholov grafu v a u, keď $v, u \in V(G)$, ako Euklidovskú vzdialenosť dvoch bodov a označme si ju $dist_G(v,u)$. Keď si uvedomíme, že každé dva vrcholy grafu v a u vždy tvoria jednu hranu $e = \{v,u\}$, môžeme zaviesť jednoduchšie značenie vzdialenosti $dist_G(e)$, ktorú budeme v celom nasledujúcom texte používať.

Ďalej uvažujme modely, kde neznáma funkcia f reprezentuje priestorové súradnice $X(v) = (x_1(v), \ldots, x_k(v))$ vrchola v grafu $G(V(G), E(G))$ a difúzny koeficient, z rovnice (2.1), bude závisieť od vzdialenosti vrcholov, čo reprezentuje nelineárny difúzny model na grafe, ktorý je zovšeobecnením Perona-Malikovho modelu zo spracovania obrazu [13]. Teda v prípade nelineárnej difúzie vychádzajme z rovnice (2.1), ktorá bude mať ale nasledujúci tvar

$$\partial_t X(v) = \nabla \cdot (g(|dist_G(e)|)\nabla X(v)), \qquad v \in V(G), \tag{2.3}$$

pričom difúzny koeficient g je daný funkciou, ktorá závisí od dĺžky hrán grafu, a má tvar

$$g(|dist_G(e)|) = \frac{1}{1 + K|dist_G(e)|^2}, \qquad K \geq 0. \tag{2.4}$$

Konštanta K reprezentuje váhu, ktorou udávame ako ovplyvňuje dĺžka hrany $e = \{v,u\}$ vývoj vrcholov v a u v čase. Difúzny koeficient g nadobúda hodnoty z intervalu $0 < g(|dist_G(e)|) \leq 1$. Ak sú hodnoty difúzneho koeficienta blízke nule, hovorí nám to, že celý difúzny proces bude pomalší a teda bude potrebný dlhší čas, aby sme prišli k požadovanému výsledku. V prípade ak sú hodnoty blízke jednotke, celý proces bude rýchlejší a bude sa podobať lineárnej difúzii. Parameter K získame experimentálnym ladením parametrov v učiacej fáze algoritmu tak, aby sme dostali optimálne výsledky.

2.2 Numerická diskretizácia

2.2.1 Priestorová diskretizácia

Na diskretizáciu rovnice (2.1) využijeme bilanciu difúznych tokov (vtokov a výtokov) v každom vrchole $v \in V(G)$ a aproximáciu difúzneho toku do vrcholu v pozdĺž hrany e, pre ktorú je v jedným z vrcholov.

Najskôr definujme aproximáciu difúzneho toku

$$\mathcal{F}(v,e) = g_e \frac{f(u) - f(v)}{dist_G(e)}, \tag{2.5}$$

pre každú hranu $e = \{v, u\}$, kde g_e predstavuje difúzny koeficient na hrane e. Pokiaľ $\mathcal{F}(v,e) > 0$, predstavuje vtok veličiny f do vrchola v, naopak ak $\mathcal{F}(v,e) < 0$, predstavuje výtok veličiny f z vrchola v. Bilanciu difúznych tokov vo vrchole v, potom vyjadríme v tvare

$$\partial_t f(v) = \sum_{e \ni v} \mathcal{F}(v,e), \tag{2.6}$$

ktorý hovorí, že časová zmena veličiny f vo vrchole v je kladná (hodnota veličiny f narastá v čase), ak je súčet vtokov a výtokov vo vrchole v kladný, teda viac do vrchola vtečie, ako vytečie. Naopak, časová zmena veličiny f vo vrchole v je záporná, ak je súčet vtokov a výtokov vo vrchole v záporný, teda z vrchola viac vytečie, ako vtečie. Po dosadení aproximácie difúznych tokov (2.5) do ich bilancie (2.6) dostaneme

$$\partial_t f(v) = \sum_{e \ni v} g_e \frac{f(u) - f(v)}{dist_G(e)}. \tag{2.7}$$

Pravá strana rovnice (2.7) pritom v teórii grafov reprezentuje tzv. „graph-Laplacian“ (viď napríklad rovnicu (12) v [4]), ktorý je pre tzv. ohodnotený úplný neorientovaný graf daný vzťahom

$$\nabla \cdot (\nabla f)(v) = \frac{1}{\nu(v)} \sum_{e \ni v} g_e \frac{f(u) - f(v)}{dist_G(e)}, \tag{2.8}$$

kde $\nu(v)$ reprezentuje mieru vrchola v a g_e reprezentuje „váhu“ hrany ohodnoteného grafu. V numerickej matematike by sme takto definovaný „Laplacián“ chápali ako priemerný Laplaceov operátor na konečnom objeme v s mierou (plochou/objemom) $\nu(v)$. Naša numerická diskretizácia difúznej rovnice na úplnom neorientovanom grafe zodpovedá voľbe $\nu(v) = 1$, čo je aj štandardná voľba pre mieru vrchola v teórii grafov, viď [4].

Takisto štandardnou v teórii grafov je voľba $dist_G(e) = 1$ vo vzťahu (2.8), ktorú urobíme aj my, a to preto, že $dist_G(e)$ bude vystupovať v definícii difúzneho koeficienta g_e. Takto dostaneme aproximáciu difúznej rovnice na grafe v tvare

$$\partial_t f(v) = \sum_{e \ni v} g_e (f(u) - f(v)). \tag{2.9}$$

2.2.2 Časová diskretizácia

Pre časovú diskretizáciu modelu (2.1) si odvodíme semi-implicitnú numerickú schému [9]. Rovnicu budeme riešiť na časovom intervale $t \in [0, T]$, ktorý si rozdelíme na M časových krokov $t_i,\ i = 1, \ldots, M$ a τ bude predstavovať veľkosť časového kroku. Aproximáciu časovej derivácie budeme riešiť použitím metódy konečných diferencií, konkrétne zvolíme spätnú diferenciu, pričom zavedieme značenie pre $f(t_n) \equiv f^n$

$$\partial_t f(v) \approx \frac{f^n(v) - f^{n-1}(v)}{\tau}.$$

V prípade rovnice (2.1) bude mať semi-implicitná schéma tvar

$$\frac{f^n(v) - f^{n-1}(v)}{\tau} = \sum_{e \ni v} g_e^{n-1}(f^n(u) - f^n(v)). \tag{2.10}$$

Keďže difúzny koeficient g_e na hrane $e = \{v, u\}$ sa bude môcť v čase meniť a to v závislosti od riešenia, v rovnici (2.10) hodnoty použité vo funkcii g_e vezmeme z predošlého časového kroku, teda difúzny koeficient si môžeme označiť ako g_e^{n-1}.

Semi-implicitná schéma sa dá v každom časovom kroku $n = 1, \ldots, M$ prepísať do tvaru systému lineárnych rovníc

$$(1 + \sum_{e \ni v} g_e^{n-1}) f^n(v) - \tau \sum_{e \ni v} g_e^{n-1} f^n(u) = f^{n-1}(v). \tag{2.11}$$

Tento systém rovníc je reprezentovaný plnou maticou, a ako sme povedali už aj predtým, pre úplný neorientovaný graf prirodzene netreba zadávať žiadne okrajové podmienky, keďže difúzia prebieha medzi všetkými vrcholmi grafu.

2.2.3 Numerický model pre klasifikáciu/klasterizáciu dát

V prípade klasifikácie/klasterizácie dát z k-rozmerného priestoru čŕt (feature space) budú našimi difundujúcimi veličinami Euklidovské súradnice $X(v) = (x_1(v), \ldots, x_k(v))$ vrcholov v grafu $G(V(G), E(G))$ a dostaneme tak vo všeobecnosti v každom časovom kroku k systémov rovníc

$$(1 + \sum_{e \ni v} g_e^{n-1}) x_i^n(v) - \tau \sum_{e \ni v} g_e^{n-1} x_i^n(u) = x_i^{n-1}(v), \qquad i = 1, \ldots, k, \tag{2.12}$$

ktoré sú vzájomne previazané voľbou difúzneho koeficienta g_e^{n-1} v závislosti od vzdialeností vrcholov v grafe.

Pri základnom modeli klasterizácie dát (nekontrolovanom učení) používame iba model s doprednou difúziou a v tom prípade difúzny koeficient má tvar

$$g_e^{n-1} = \frac{1}{1 + K|dist_G(e^{n-1})|^2}, \qquad K \geq 0. \tag{2.13}$$

Dopredná difúzia je daná kladným difúznym koeficientom a spriemerováva hodnoty, čo je odrazom vlastnosti zhladzovania rovnice difúzie. Dopredná sa nazýva preto, lebo popisuje proces difúzie smerom do budúcnosti, kedy ho vieme vypočítať. V našej aplikácii sa prejaví priťahovaním bodov k sebe navzájom.

Ak budeme požadovať aplikáciu spätnej difúzie (pri kontrolovanom učení), v tom prípade difúzny koeficient bude mať nasledovný tvar

$$g_e^{n-1} = \varepsilon \, \frac{1}{1 + K|dist_G(e^{n-1})|^2}, \qquad K \geq 0, \tag{2.14}$$

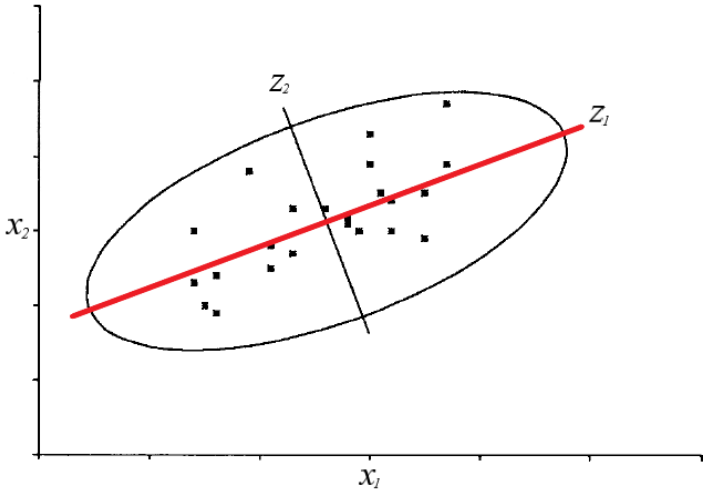
kde napríklad $\varepsilon = -0.01$. Spätná difúzia má, práve naopak, difúzny koeficient záporný, a dá sa chápať tak, že sa s difúziou chceme vrátiť do minulosti. Je to akoby opak zhladzovania, spriemerovania hodnôt, čo sa v našom modeli prejaví odpudzovaním bodov. Ak by sme použili iba model so spätnou difúziou, body by sa od seba len vzďaľovali a celý systém by sa stal nestabilným. Ale pri rozumnej kombinácii doprednej a spätnej difúzie, kedy pri spätnej difúzii volíme malý koeficient ε nepozorujeme žiadnu nestabilitu výpočtov.

2.2.4 Analýza hlavných komponentov

Pred samotnou difúziou aplikujeme na veľkorozmerné dáta, korešpondujúce počiatočnej podmienke $X^0(v) = (x_1^0(v), \ldots, x_k^0(v))$, $\forall v \in V(G)$, analýzu hlavných komponentov (PCA – Principal Component Analysis) a zredukujeme tak dimenziu dát. Vyberieme iba charakteristické črty popisujúce dáta a menej podstatné črty sa zanedbajú.

Zvolili sme analýzu hlavných komponentov kvôli tomu, že je jednou zo základných metód kompresie dát, kedy sme schopný pôvodné veľkorozmerné dáta reprezentovať menším počtom rozmerov. Dáta v novom súradnicovom systéme predstavujú systém hlavných komponentov a pozostávajú z lineárnych kombinácií pôvodných dát, kde nosnú úlohu má prvý hlavný komponent, ktorý vystihuje najväčšiu časť variability pôvodných dát. V podstate hľadáme smer, pozdĺž ktorého sú dáta maximálne rozptýlené. Druhý hlavný komponent je lineárnou kombináciou s maximálnym rozpty-

lom v smere kolmom na os prvého hlavného komponentu (Obr. 2.1) a analogicky to pokračuje ďalej.



Obr. 2.1: Ukážkové 2D dáta v pôvodných súradniciach (x_1, x_2) rozmiestnené do tvaru elipsy. Červená čiara reprezentuje smer najväčšieho rozptylu dát (os prvého hlavného komponentu), teda aj nový súradnicový systém (z_1, z_2).

Analýza hlavných komponentov môže byť použitá na akékoľvek rozmiestnenie vstupných dát $X^0(v)$, ale pre jednoduchú ilustráciu sú najvhodnejšie dáta rozmiestnené do tvaru elipsoidu (v prípade 2D ako na Obr. 2.1). Pre potreby intuitívneho vysvetlenia analýzy hlavných komponentov budeme uvažovať dáta rozložené do tvaru elipsoidu. V prípade ak sú hodnoty $x_1^0(v), \ldots, x_k^0(v)$ korelované (môžeme pozorovať ich vzájomnú závislosť), hlavné osi elipsoidu, ktorý dáta tvoria, nie sú paralelné so žiadnou osou súradnicového systému, v ktorom sú reprezentované hodnoty $x_1^0(v), \ldots, x_k^0(v)$. Našim cieľom je nájsť prirodzené osi takto rozmiestnených hodnôt (osi elipsoidu) so stredom v $\bar{X^0}$, pričom $\bar{X^0}$ je vektor stredných hodnôt $X^0(v)$. Zrealizujeme to posunutím stredu súradnicovej sústavy do $\bar{X^0}$ a následne pootočíme celú súradnicovú sústavu. Týmto spôsobom získame nové premenné (hlavné komponenty) $Z^0(v) = (z_1^0(v), \ldots, z_k^0(v))$, $\forall v \in V(G)$, ktoré budú nekorelované.

Skôr než aplikujeme analýzu hlavných komponentov, vstupné dáta $X^0(v)$ preškálujeme do intervalu $[0, 1]$. V prípade ak by sme preškálovali iba jednu alebo len niekoľko hodnôt z $X^0(v)$, zmenilo by sa rozmiestnenie dát a potrebovali by sme iné hlavné komponenty pre reprezentáciu nových premenných $Z^0(v)$. Teda analýza hlavných komponentov nie je invariantná voči škálovaniu jednotlivých dát rôznymi škálovacími faktormi

[14].

Postup pri analýze hlavných komponentov by sa dal jednoducho zhrnúť a to tak, že dáta najprv vycentrujeme, teda vypočítame strednú hodnotu v jednotlivých súradniciach všetkých bodov a odčítame ju od každého bodu v príslušnej súradnici. Následne zostavíme kovariačnú maticu a vypočítame vlastné čísla a zodpovedajúce vlastné vektory tejto kovariačnej matice. Vlastné vektory musíme normalizovať a vytvoriť z nich jednotkové vlastné vektory, čím dostaneme hlavné osi elipsoidu. Nakoniec stačí takto vytvorenou maticou vynásobiť vstupné dáta a tým získame nové hlavné komponenty.

Pre potreby tejto časti označme $X(v) = X^0(v) - \bar{X}^0$ a tiež označme hlavné komponenty $Z(v) \equiv Z^0(v)$. Zavedieme aj označenie matice dát X, ktorej jednotlivé riadky sú dané vektormi $X(v)$, $\forall v \in V(G)$, pričom jeden jej riadok označíme $x_{(i)}$, $i = 1, \ldots, N_V$. Matica dát X má teda N_V riadkov, čo je vlastne počet vrcholov v grafe, a k stĺpcov, čo predstavuje črty dát. Analogicky zavedieme aj označenie matice hlavných komponentov Z, ktorej riadky sú dané vektormi $Z(v)$, $\forall v \in V(G)$ a jeden jej riadok označíme $z_{(i)} = (z_1, \ldots, z_k)_{(i)}$, $i = 1, \ldots, N_V$. Rovnako ako matica X, aj matica Z je rozmeru $N_V \times k$.

Analýza hlavných komponentov predstavuje ortogonálnu lineárnu transformáciu, ktorá je matematicky definovaná transformačnou maticou koeficientov W a zobrazuje pôvodné súradnice bodov $X(v)$ do nových súradníc, hlavných komponentov, $Z(v)$. Transformačná matica koeficientov W je vytvorená tak, že jej stĺpce $w_{(j)}$, $j = 1, \ldots, k$, sú tvorené vlastnými vektormi (kovariančnej) matice dát $X^T X$. Teda transformované dáta získame súčinom matíc, ako

$$Z = X\ W, \tag{2.15}$$

respektíve jednotlivé súradnice transformovaných bodov ako

$$z_{j(i)} = x_{(i)} \cdot w_{(j)}, \qquad i = 1, \ldots, N_V, \qquad j = 1, \ldots, k. \tag{2.16}$$

Prečo nám v transformácii vystúpia vlastné vektory kovariančnej matice vidno z nasledujúcich úvah.

Za účelom maximalizácie rozptylu, musí prvý hlavný komponent spĺňať

$$w_{(1)} = \arg\max_{\|w\|=1} \left\{ \sum_{v \in V(G)} (z_1(v))^2 \right\} = \arg\max_{\|w\|=1} \left\{ \sum_{v \in V(G)} (x_{(i)} \cdot w)^2 \right\}.$$

Ekvivalentne zapísané v maticovom tvare

$$w_{(1)} = \arg\max_{\|w\|=1}\{\| Xw \|^2\} = \arg\max_{\|w\|=1}\{w^T X^T X w\}.$$

Keďže $w_{(1)}$ je definovaný ako jednotkový vektor, tiež platí

$$w_{(1)} = \arg\max\{\frac{w^T X^T X w}{w^T w}\}.$$

Veličina, ktorú maximalizujeme zodpovedá takzvanému Rayleighovmu kvocientu [18]. Štandardným výsledkom pre pozitívne semidefinitnú maticu, akou je aj (kovariačná) matica $X^T X$, je, že Rayleighov kvocient je maximalizovaný vlastným vektorom prislúchajúcim najväčšiemu vlastnému číslu tejto matice. Prvý stĺpec transformačnej matice W je teda daný týmto vlastným vektorom a na získanie ďalších hlavných komponentov sa postupuje analogicky, viď [17].

Prvá súradnica hlavných komponentov $z_1(v)$ má najväčší rozptyl, pretože rozptyl $z_1(v)$ je rovný najväčšiemu vlastnému číslu λ_1, kým posledná súradnica $z_k(v)$ má najmenší rozptyl, lebo rozptyl $z_k(v)$ je rovný najmenšiemu vlastnému číslu λ_k. Ak sú niektoré vlastné čísla príliš malé môžeme ich zanedbať, čím ale nestratíme takmer žiadnu informáciu, keďže pri malých vlastných číslach je aj rozptyl malý, a teda na reprezentáciu dát nám postačí menší počet dimenzií, najčastejšie si zvolíme $k = 2$. Napríklad, ak na začiatku máme dané 3D dáta, teda $k = 3$, a λ_3 je príliš malé, vstupné dáta sa podobajú na „eliptickú palacinku“ a teda nám postačia iba dve dimenzie na popísanie dát, teda redukujeme k, a položíme $k = 2$.

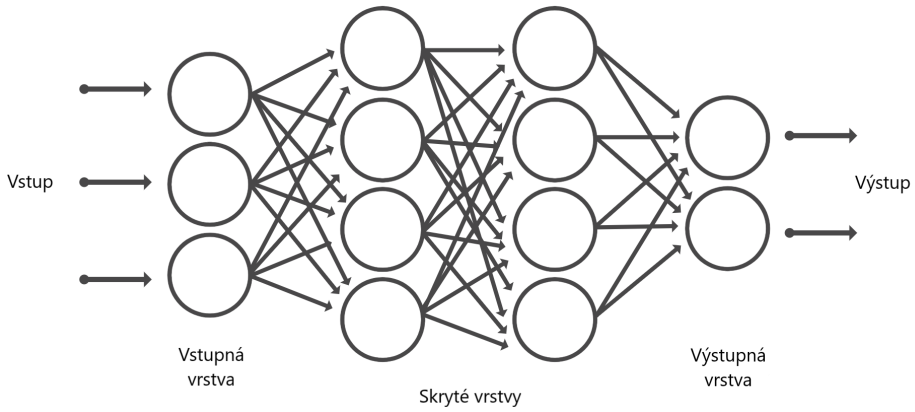
2.3 Hlboké strojové učenie

Pri strojovom učení (machine-learning) sa snažíme naučiť program správať sa v rôznych situáciach tak, ako by to bolo prirodzené pre človeka. Najprv musí prejsť učiaca fáza, v ktorej máme k dispozícii rôzne situácie a ich riešenia, a snažíme sa vytvoriť taký algoritmus, kedy sa program rozhodne správne reagovať v čo najviac prípadoch. So zvyšovaním počtu rôznych situácií, ktoré môžu nastať, sa zvyšuje aj presnosť naprogramovaného algoritmu.

Hlboké strojové učenie (deep-learning) je špeciálnym odvetvím strojového učenia, založené na neurónových sieťach (neural networks), ktoré obsahujú množstvo vrstiev.

Kvôli tomu, že hlboké strojové učenie je založené na architektúre neurónových sietí, často sa označuje ako hlboké neurónové siete (deep neural networks). Modely hlbokého strojového učenia dosahujú veľkú presnosť, mnohokrát až takú, že sú porovnateľné s ľudským správaním. To je zapríčinené aj tým, že v trénovacej časti prebieha učenie na veľkom množstve vzoriek, ktoré sú tvorené vopred označenými dátami.

Pojem hlboké poukazuje na počet skrytých vrstiev v neurónovej sieti. Klasická neurónová sieť obsahuje iba dve až tri skryté vrstvy (Obr. 2.2), kým hlboké strojové učenie môže mať aj viac ako sto skrytých vrstiev. Ďalší rozdiel, kvôli ktorému je hlboké učenie pre nás zaujímavým je, že presnosť algoritmu narastá so zväčšovaním počtu vstupných dát, kým pri iných algoritmoch s narastajúcim počtom dát sa presnosť najprv náhle zväčší a následne začne stagnovať [12]. Pri hlbokom strojovom učení sa dôležité vlastnosti z dát získavajú priamo z dát a nie je potrebné žiadne explicitné zadávanie podstatných vlastností, ako pri strojovom učení. Teda algoritmus si sám vypočíta charakteristiky, ktoré sú potrebné na ďalší postup a nie je potrebné, aby takéto hodnoty vnášal užívateľ. Hlboké strojové učenie predstavuje „end-to-end“ učenie, kedy sú programu dané pôvodné vstupné dáta a úloha, ktorú má program vykonať. Algoritmus si sám vyhľadá dôležite popisujúce vlastnosti v dátach a podľa natrénovanej fázy splní svoju úlohu a vráti vhodný výstup.

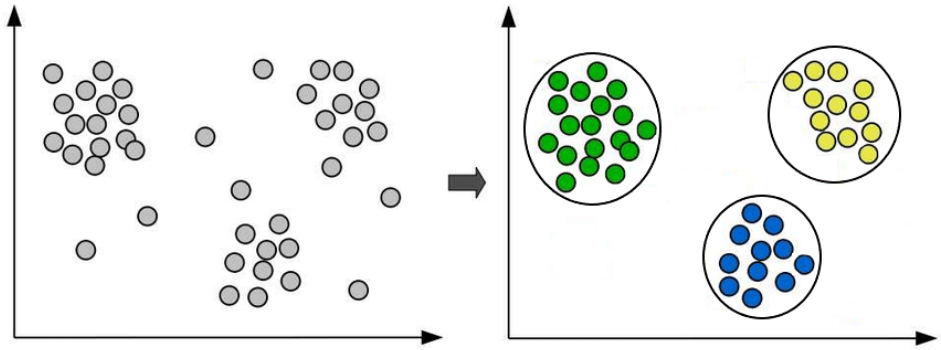


Obr. 2.2: Klasická neurónová sieť s dvomi skrytými vrstvami.

Učenie programu prebieha v dvoch fázach a to fázou učenia, nazývanou aj trénovacia fáza, a fázou validácie. V trénovacej fáze sa vytvorí a implementuje matematický model, ktorý sa následne natrénuje na určenú úlohu na trénovacej vzorke dát. Nasleduje fáza validácie, kedy preveríme funkčnosť programu. Na základe toho ako prebieha proces učenia, môžeme algoritmy rozdeliť na:

- nekontrolované učenie - učenie bez učiteľa (unsupervised learning) a
- kontrolované učenie - učenie s učiteľom (supervised learning).

My v ďalšom texte budeme používať termíny nekontrolované a kontrolované učenie.

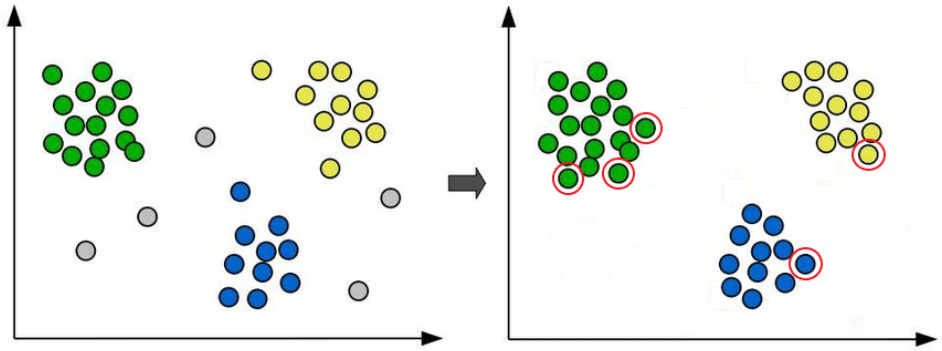


Obr. 2.3: Vstupné neoznačené dáta (naľavo) a zoskupené dáta nekotrolovaným učením, klasterizáciou, na základe ich čŕt do troch klastrov (napravo).

Nekontrolované učenie prebieha na dátach, ktoré neboli vopred označené (Obr. 2.3 (naľavo)), a teda nemáme explicitne dané ako má vyzerať výstup. Algoritmus preto nemá ako prísť na to čo by bolo správnym výstupom, nemá žiadnu naučenú vzorku, a preto preskúma dáta a popíše štruktúry v neoznačených dátach. Úloha, ktorú rieši nekontrolované učenie je zoskupovanie množiny objektov do skupín, ktoré nazývame klastre (clusters), podľa toho ako sú si objekty navzájom podobné (názorná ukážka na Obr. 2.3 (napravo)). Tento spôsob spracovania dát nazývame klasterizácia (clustering) a je to jeden z najčastejších problémov, ktoré rieši nekontrolované učenie.

Naopak kontrolované učenie prebieha na veľkej vzorke dát, ktoré boli vopred správne označené, a vieme presne v prípade jedného vstupu, aký výstup máme dostať. Na takýchto dátach prebehne trénovací proces, kedy budeme žiadať od algoritmu pre daný vstup vyprodukovať očakávaný výstup. Funkčnosť takto natrénovaného algoritmu si vo validačnej časti overíme tak, že mu na vstup dáme nové dáta, ktoré v trénovacej fáze nevidel, a výstup vyhodnotíme. Presnosť kontrolovaného učenia teda veľmi závisí od množstva kvalitných dát v učiacej fáze. Medzi hlavné problémy, ktoré riešime kontrolovaným učením, patrí klasifikácia (classification). Pri klasifikácii dát sa snažíme identifikovať, ku ktorému klastru patrí nová nezaradená hodnota (názorná ukážka na Obr. 2.4). Teda na začiatku máme dané vstupné dáta, počet a označenie klastrov. Vstupné dáta majú predpísané, ku ktorému klastru patria, a ku ním následne pridávame novú

nezaradenú hodnotu. Úlohou klasifikácie je správne zaradiť novú hodnotu, na základe jej čŕt, ktoré sa získajú priamo z dát.



Obr. 2.4: Vstupné označené dáta s piatimi novými bodmi(naľavo) a zaradené nové body kontrolovaným učením, klasifikáciou, podľa čŕt získaných z dát (napravo).

Môžeme vi všimnúť, že rozdiel medzi nekontrolovaným a kontrolovaným učením je hlavne v tom, že nekontrolované učenie si samo vytvára klastre na základe analýzy čŕt vstupných dát a kontrolované učenie priraďuje dáta k už existujúcim klastrom.

Kapitola 3

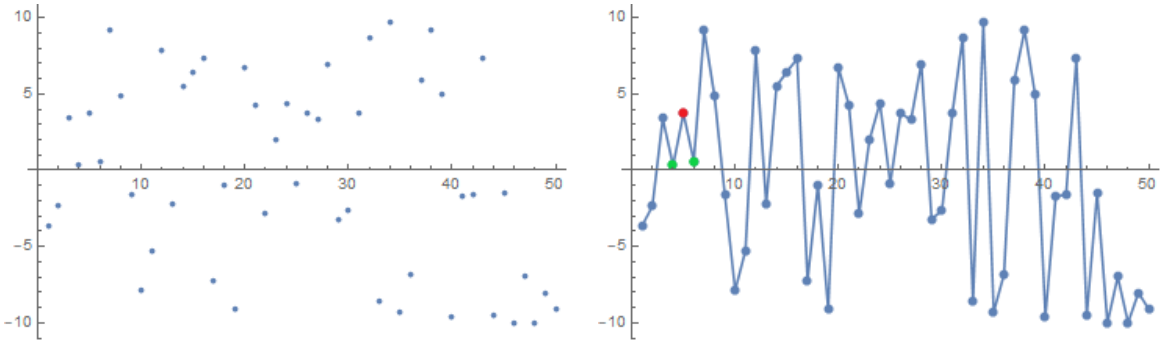
Implementácia algoritmu

3.1 Jednoduchý 1D príklad

Proces implementácie sme začali jednoduchým príkladom v softvéri Mathematica, a to implementáciou štandardnej rovnice difúzie (2.2) v 1D oblasti, nie na úplnom grafe. Interval $[0, 50]$, sme rozdelili na 50 rovnakých podintervalov, následne sme náhodne vygenerovali hodnoty, ktoré sme priradili ku každému podintervalu. Takto nám vznikli 2D body (Obr. 3.1 (naľavo)). Môžeme si za tým predstaviť nerozvetvené potrubie dĺžky 50, v ktorom dominuje difúzia v x-ovej osi potrubia a teda problém môžeme uvažovať ako 1D priestorovú úlohu nestacionárnej difúzie koncentrácie látky v potrubí. Teda na x-ovej osi je zaznačená dĺžka potrubia rozdelená na 50 podintervalov a na y-ovej osi sú zaznačené koncentrácie v každej časti potrubia (Obr. 3.1 (naľavo)). Nás zaujíma, ako sa budú body správať po aplikácii štandardnej rovnice difúzie. Musíme si uvedomiť, že pri rovnici difúzie, na konkrétny bod budú bezprostredne vplývať iba jeho susedia. V prípade bodov na okrajoch, aplikujeme nulovú Neumannovu okrajovú podmienku, čo znamená, že potrubie na okrajoch je izolované.

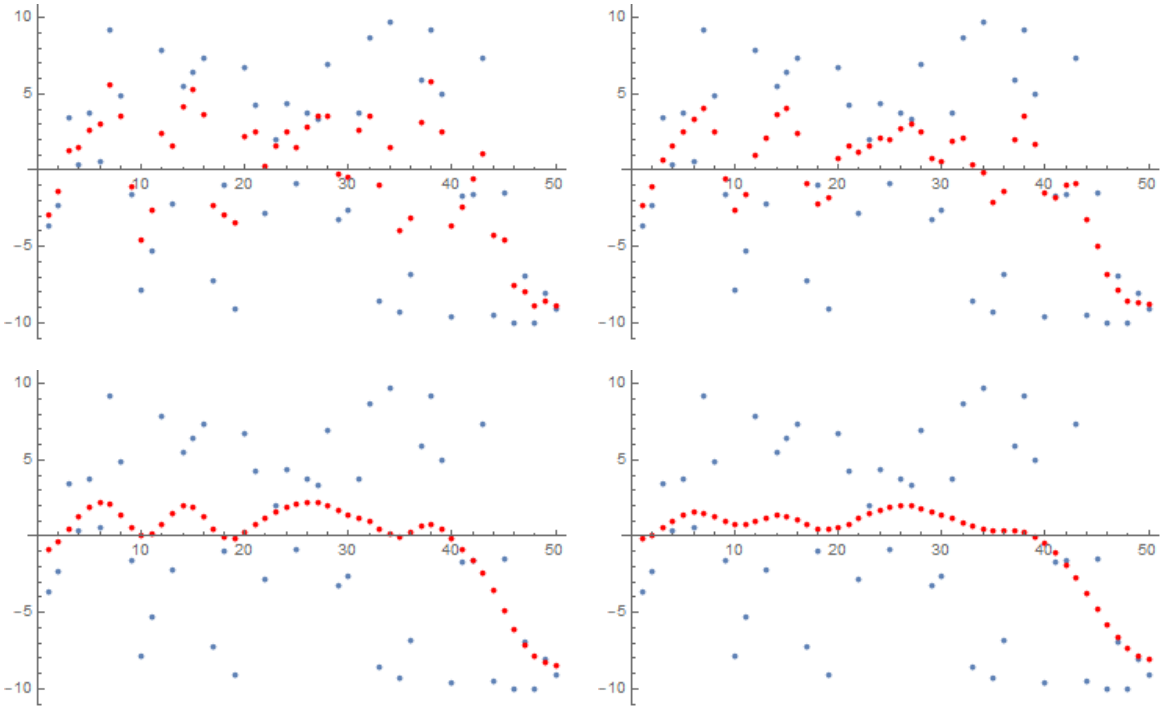
Pri numerickom riešení používame numerickú schému (2.11), pre skalárnu veličinu f. Pre numerické riešenie lineárneho systému rovníc použijeme SOR (successive over-relaxation) iteračnú metódu, v ktorej nastavíme hodnotu relaxačného parametera na $\omega = 1.25$, ktorú sme získali experimentálne a toleranciu konvergencie sme zvolili

$tol = 10^{-6}$. Počet časových krokov v tomto prípade bude $n = 50$ a veľkosť časového kroku zvolíme $\tau = 0.1$.



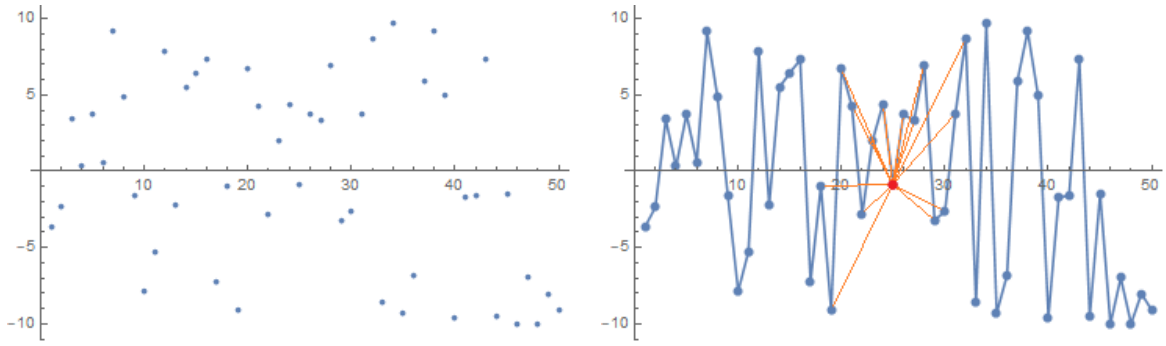
Obr. 3.1: Vstupné dáta bodovo reprezentované (naľavo) a čiarovo reprezentované (napravo) s vyznačeným konkrétnym bodom (červený bod) a jeho bezprostrednými susedmi (zelené body).

Na obrázku (Obr. 3.2) si môžeme všimnúť jasnú charakteristiku vplyvu štandardnej rovnice difúzie a to lineárne spriemerňovanie hodnôt. V príklade, ktorý sme si uviedli, by to znamenalo vyrovnávanie koncentrácií v potrubí, až nakoniec by sme dospeli k stavu rovnováhy koncentrácií. V prípade, ak by sme aplikovali viac krokov lineárnej difúzie dospeli by sme do stavu konštantnej hodnoty v každom bode potrubia.



Obr. 3.2: Vývoj vstupných dát (modré body) a výsledná pozícia dát (červené body) po aplikácii štandardnej rovnice difúzie v časových krokoch $n = 5, 10, 30, 50$.

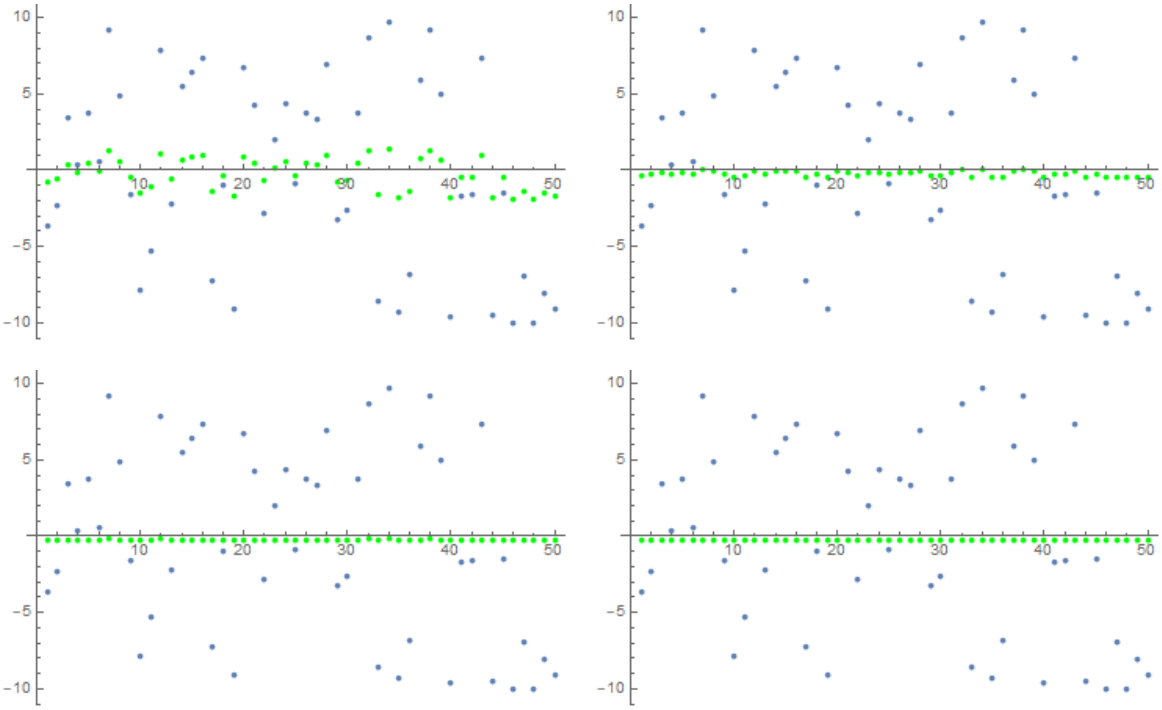
Jednoduchý príklad popísaný vyššie sme mierne pozmenili a to tak, že sme aplikovali rovnicu difúzie na úplný neorientovaný graf. Konkrétne v tomto príklade použijeme rovnicu lineárnej difúzie (2.2), aby sme mohli porovnať výsledky. Aj v tomto prípade si interval $[0, 50]$ rozdelíme na 50 rovnakých podintervalov a vygenerujeme náhodné hodnoty, ktoré priradíme ku každému podintervalu. Úlohu si môžeme znovu predstaviť ako rozvetvené potrubie dĺžky 50, ale v tomto prípade máme v každej časti definovaný uzol. Predpokladajme, že dominuje difúzia v osi potrubia a teda problém môžeme uvažovať ako 1D priestorovú úlohu nestacionárnej difúzie koncentrácie látky v potrubí. Na x-ovej osi je zaznačená poloha rovnomerne rozmiestnených uzlov a na y-osi sú hodnoty v každom uzle potrubia (Obr. 3.3 (naľavo)). Všetky uzly potrubia sú prepojené medzi sebou tak, aby štruktúra potrubia tvorila úplný neorientovaný graf. V tomto príklade neuvažujeme možnosť dvoch uzlov potrubia na rovnakej x-ovej pozícii, kvôli jednoduchému porovnaniu s predchádzajúcim príkladom. Nás bude, rovnako ako v predchádzajúcom prípade, zaujímať ako sa budú body správať po aplikácii lineárnej difúzie na grafe. Hovoríme o úplnom neorientovanom grafe, teda všetky body na seba pôsobia a ovplyvňujú sa rovnakým spôsobom, pretože difúzny koeficient $g = 1$. Ako bolo uvedené v kapitole 2, v prípade úplného neorientovaného grafu nie sú potrebné okrajové podmienky, pretože každý vrchol pôsobí na všetky ostatné.



Obr. 3.3: Vstupné dáta bodovo reprezentované (naľavo) a čiarovo reprezentované (napravo), kde vidíme, kvôli prehľadnosti, iba pár susedských vzťahov (oranžové čiary) pre vyznačený bod (červený bod) v úplnom neorientovanom grafe.

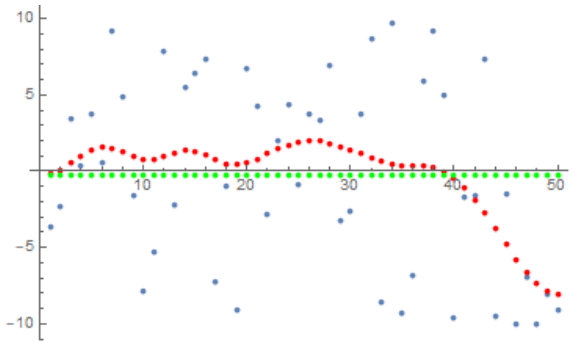
Na vstupné dáta aplikujeme $n = 50$ časových krokov lineárnej difúzie na grafe s krokom $\tau = 0.1$ a výsledky si môžeme pozrieť na obrázku (Obr. 3.4). V popise obrázka môžeme vidieť, že zobrazené sú iba prvé štyri časové kroky a to z toho dôvodu, že všetky nasledovné sú už rovnaké konštantné hodnoty. Teda zisťujeme, že body sa veľmi rýchlo

dostali do stavu rovnováhy a v nej zotrvali. Takýto vývoj sme mohli očakávať, keďže difúzia prebiehala rovnako medzi všetkými vrcholmi grafu. Navyše zatiaľ neberieme do úvahy vzdialenosti medzi vrcholmi alebo či majú niektoré susedné vrcholy rovnaké súradnice, čo, ako uvidíme neskôr, bude ovplyvňovať difúziu. Keď si to skúsime popísať pomocou príkladu potrubia, znamená to, že koncentrácia v uzle sa prerozdelí nie iba na susedné uzly, ale na všetky uzly v potrubí, čo zapríčiní, že proces vyrovnávania koncentrácií je rýchlejší. Po dosiahnutí konštantnej hodnoty v každom bode potrubia sa už hodnota v uzle nemá dôvod meniť a preto po $n = 4$ časovom kroku graf vývoja zostáva nemenný.



Obr. 3.4: Vývoj vstupných dát (modré body) a výsledná pozícia dát (zelené body) po aplikácii lineárnej difúzie na grafe v časových krokoch $n = 1, 2, 3, 4$.

Z prvých dvoch príkladov nám vyplýva, že pri aplikácii rovnakej rovnice lineárnej difúzie, záleží na systéme aký riešime, teda či sú body ovplyvňované iba susednými bodmi, ako v prípade štandardnej lineárnej difúzie, alebo všetky body vplývajú na vývoj každého jedného bodu, ako v prípade lineárnej difúzie na grafe. Na obrázku (Obr. 3.5) je vidieť, že štandardná lineárna difúzia má pomalší vývoj a aj keď sa blíži k ustálenému stavu, v čase $n = 50$ je ešte od neho veľmi ďaleko. Kým lineárna difúzia na grafe, v rovnakom časovom kroku $n = 50$, dosiahla ustálený stav.

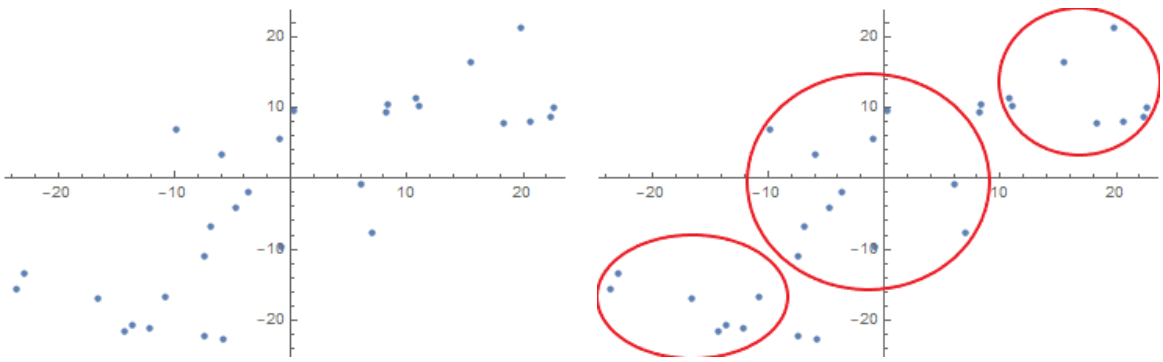


Obr. 3.5: Porovnanie štandardnej lineárnej difúzie (červené body) s difúziou na grafe (zelené body) pri časovom kroku $n = 50$.

3.2 2D nekontrolovaná klasterizácia

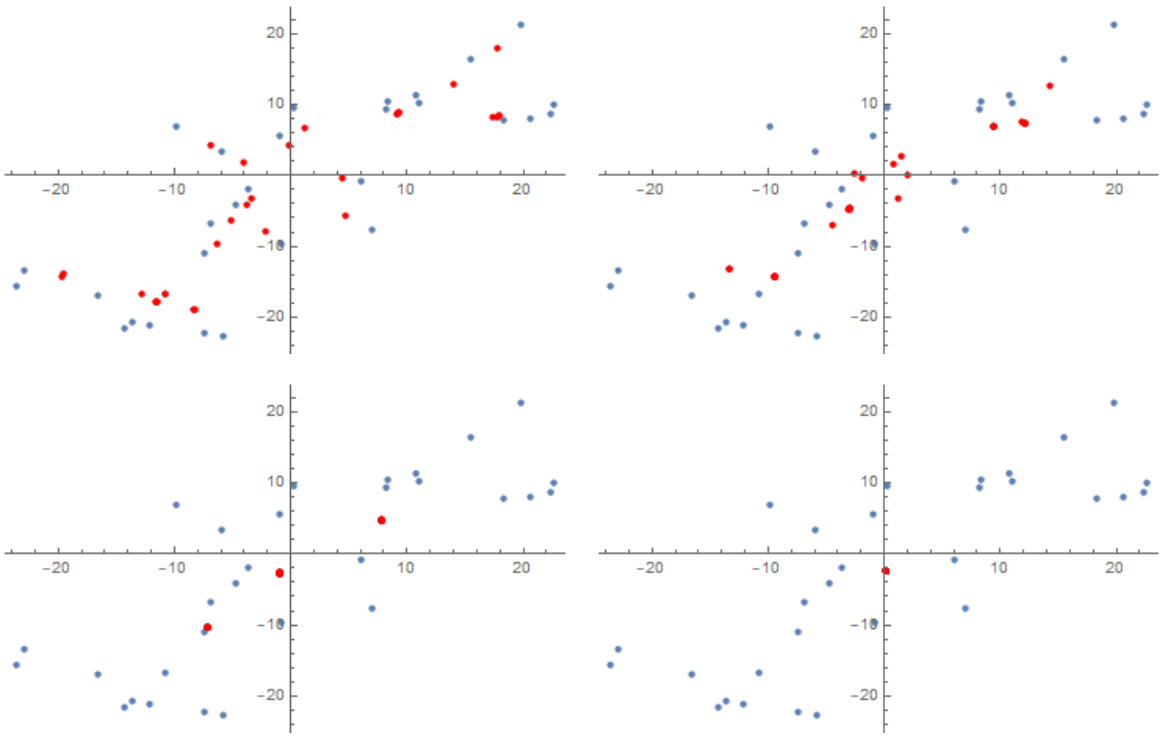
Metóda (2.12) sa dá aplikovať priamočiaro na klasterizáciu dát nekontrolovaným hlbokým strojovým učením, pričom jej úspešnosť je daná vhodným rozložením dát v priestore čŕt.

Vytvorili sme náhodné body pozdĺž priamky $y = x$ a po vizuálnej analýze sme prišli na to, že takto vygenerované body by mali tvoriť tri klastne (Obr. 3.6). Samozrejme toto je len jeden ukážkový príklad, kedy sa náhodným generovaním bodov vytvorili akoby tri klastre. Pri ďalšom náhodnom generovaní môže vzniknúť iná kombinácia rozostavenia bodov. Keďže body boli neoznačené aplikovali sme na ne algoritmus nekontrolovaného učenia, teda klasterizáciu. Očakávali sme, že sa pod vplyvom nelineárnej doprednej difúzie (2.3)-(2.4) vytvoria presne tri klastre, ktoré sme dokázali rozpoznať aj vizuálnou analýzou. Dopredná difúzia funguje tak, že sa body snažia vyrovnať hodnoty v každej súradnici a to zapríčiňuje pohyb bodov k sebe, ako sme popísali v kapitole 2.



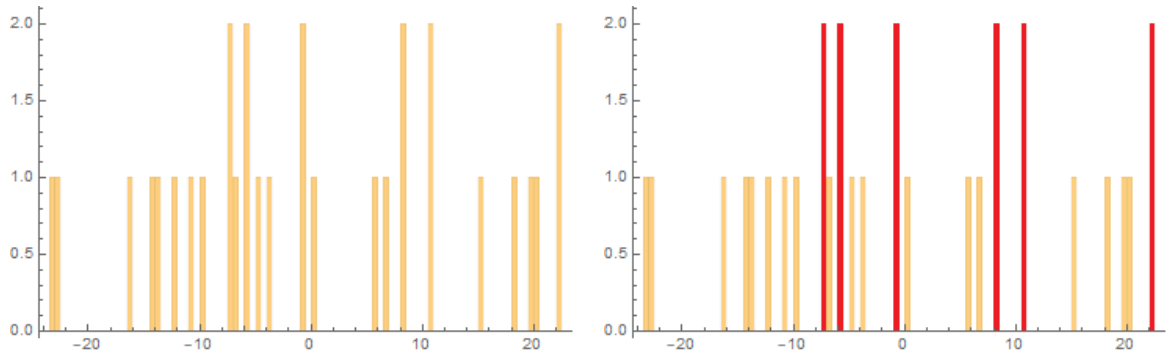
Obr. 3.6: Náhodne rozložené body pozdĺž priamky $y = x$ (naľavo) a možné klastre po vizuálnej analýze (napravo).

V matematickom ponímaní, pre 2D klasterizáciu používame numerickú schému (2.12) pre vektorovú veličinu X, teda riešime systém rovníc (2.12) v prvej súradnici bodov a následne riešime systém rovníc (2.12) v druhej súradnici. V oboch systémoch rovníc váha K pri difúznom koeficiente bola nastavená na hodnotu $K = 0.5$, pri ktorej sme získali dobré výsledky po odskúšaní viacerých hodnôt. Rovnako ako pri jednoduchom 1D príklade, aj v tomto prípade sme zvolili iteračnú SOR metódu s rovnakými parametrami $\omega = 1.25$ a $tol = 10^{-6}$. Výpočet sme spustili na $n = 200$ časových krokov s veľkosťou kroku $\tau = 0.1$. Výsledok však nebol taký, aký sme na začiatku predpokladali. Dopredná difúzia spôsobila, že sa body nevyklastrovali ako sme čakali, ale neustálym pohybom k sebe vytvorili iba jeden klaster. Ako môžeme vidieť na obrázku (Obr. 3.7), vývoj sa nezastavil v momente, keď sa body zoskupili do klastrov v časovom kroku $n = 56$, ale pokračoval ďalej až kým neprišiel po posledný časový krok $n = 200$, v ktorom už boli všetky body v jedom klastri. Z toho vyplýva, že je potrebné definovať zastavovacie kritérium, ktorým by sme dosiahli to, že po zoskupení bodov do klastrov sa celý proces zastaví. Ako zastavovacie kritérium sme implementovali metódu založenú na pozorovaní histogramu.



Obr. 3.7: Vývoj vstupných dát (modré body) a výsledná pozícia dát (červené body) po aplikácii 2D nekontrolovanej klasterizácii bez zastavovacieho kritéria v časových krokoch $n = 16, 36, 56, 200$.

Histogramové zastavovacie kritérium spočíva v tom, že sa pre každý časový krok a pre každú súradnicu dát vypočíta histogram. Teda vytvorí sa interval z minimálnej a maximálnej hodnoty v každej súradnici a tento interval sa rozdelí na menšie intervaly, podintervaly, s definovaným delením, napr. $spacing = 0.5$. Pozrieme sa koľko bodov sa nachádza v každom z podintervalov a zapíšeme si početnosť do histogramu (Obr. 3.8).

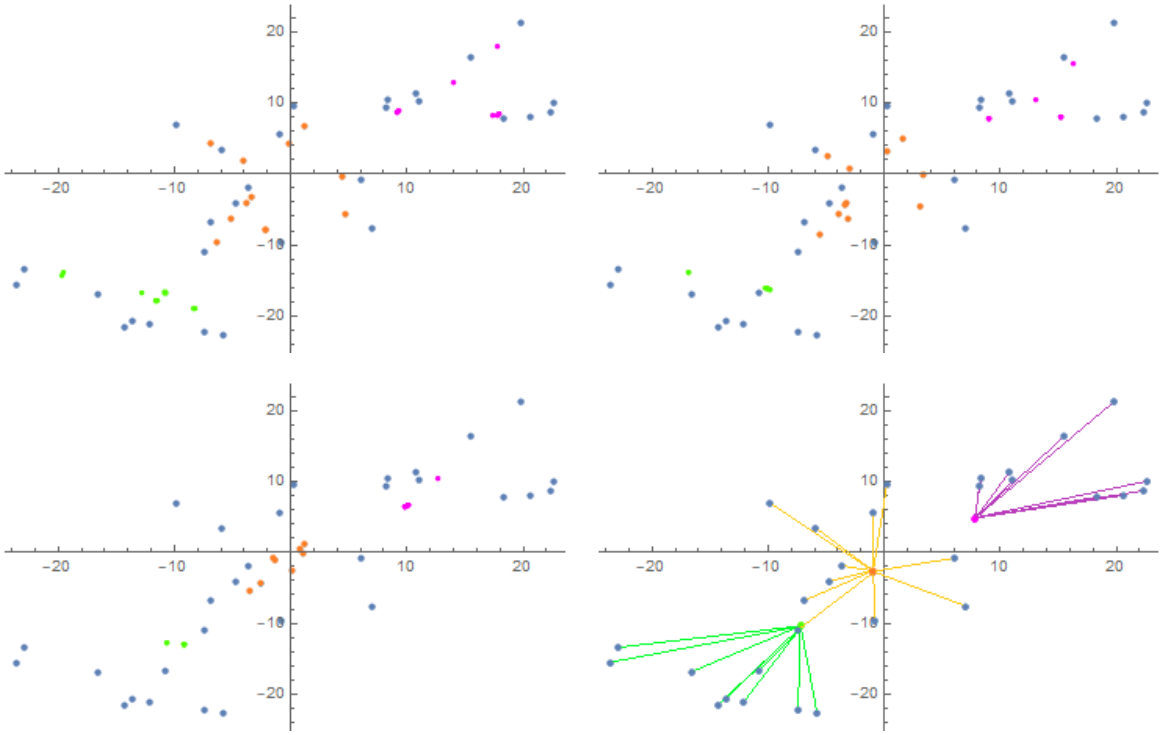


Obr. 3.8: Histogram prvej súradnice pre histogramové zastavovacie kritérium v časovom kroku $n = 1$ (naľavo) a vyznačené kontrolné indexy (červené stĺpce) v časovom kroku $n = 1$ (napravo). Na x-ovej osi histogramu je delenie intervalu minimálnej a maximálnej hodnoty prvej súradnice na podintervaly a na y-ovej osi histogramu sú početnosti výskytu bodov v podintervale.

Následne prejdeme histogram a zapíšeme si indexy podintervalov, v ktorých sa nachádza dva alebo viac bodov. Tieto indexy nám budú predstavovať kontrolné indexy, v ktorých budeme overovať ukončovaciu podmienku (Obr. 3.8 (napravo)). Na začiatku klasterizácie bude takýchto indexov veľa, keďže sú body roztrúsené v priestore. Následne overíme, či sa napravo a naľavo od kontrolného indexu nachádza n podintervalov s početnosťou nula (v našich výpočtoch používame $n = 8$). Ak áno kontrolný index predstavuje histogramovú pozíciu pre bod, v ktorom vznikol zoskupený klaster. V prípade, ak sa nachádza kontrolovaný index v blízkosti hraníc, prispôsobí sa tomu aj preskúmavané okolie. Nakoniec ak sa v histograme nachádzajú iba kontrolné indexy, pri ktorých sú v ich okolí iba nulové početnosti funkcia $clusteringDone()$, ktorá predstavuje implementáciu zastavovacieho kritéria, vráti hodnotu $true$.

Na každú súradnicu bodov aplikujeme histogramové zastavovacie kritérium a v prípade, ak funkcia $clusteringDone()$ vráti pre každú súradnicu $true$, zastavíme proces výpočtu. To znamená, že v každej dimenzii sa našli kontrolované indexy, v ktorých okolí sa nenachádzajú žiadne iné body. Nastane situácia ako na obrázku (Obr. 3.9)

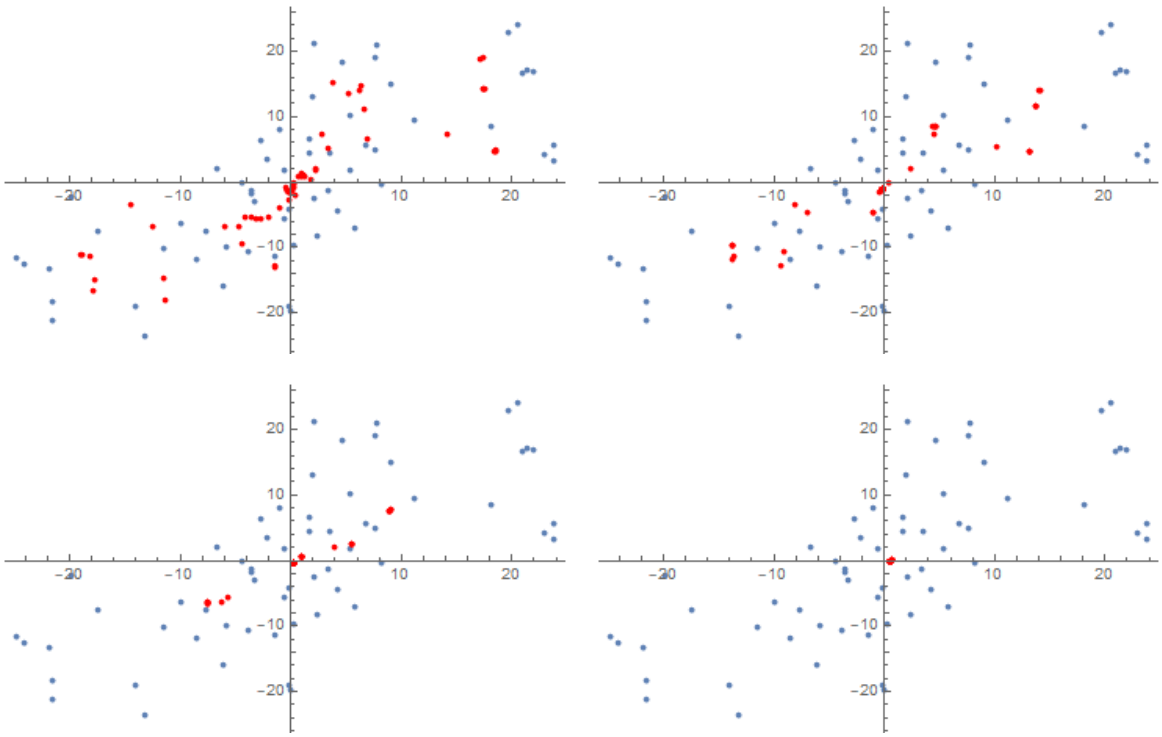
pri časovom kroku $n = 56$, kedy sa vytvorili klastre presne tak, ako sme očakávali. Na obrázku (Obr. 3.9) môžeme vidieť aj farebne odlíšené body, podľa toho ku ktorému klastru patria.



Obr. 3.9: Vývoj vstupných dát (modré body) a výsledná pozícia dát s farbou klastra, ktorému prislúchajú (farebné body), po aplikácii 2D nekontrolovanej klasterizácii so zastavovacím kritérom v časových krokoch $n = 16, 26, 42, 56$.

Výsledok nekontrolovanej klasterizácie definovanej základným modelom (2.12)-(2.13), nemusí byť vždy správny. Ako sme spomínali na začiatku časti o nekontrolovanej klasterizácii, jej úspešnosť závisí od rozloženia vstupných dát. Keďže sú dáta neoznačené, algoritmus pracuje iba s črtami, ktoré mu poskytneme v podobe súradníc bodov. Ak sú body príliš roztrúsené, čo znamená, že ich črty sú výrazne rozdielne, algoritmus zoskupí všetky body do jedného klastra (Obr. 3.10). Všetky body sa zoskupia do jedného klastra z dôvodu, že základný model obsahuje iba doprednú difúziu, ktorá spôsobuje pohyb bodov a aj už vytvorených klastrov, iba smerom k sebe. Histogramové kritérium nezastaví vývoj, pretože nenájde jednoznačne od seba oddelené klastre a celý proces skončí vytvorením jedného klastra. V takomto prípade je potrebné rozšíriť základný model o spätnú difúziu, ktorú ale môžeme aplikovať iba ak budeme vopred poznať, ku ktorému klastru bodyG patria, čiže ak budeme pracovať s vopred

označenými bodmi.

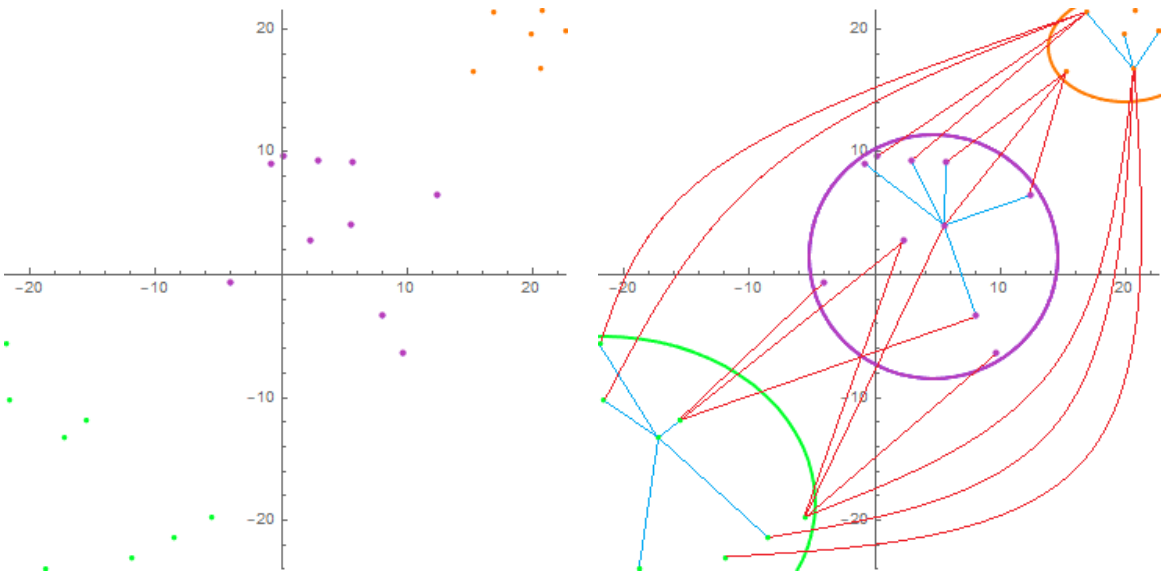


Obr. 3.10: Vývoj nevhodných vstupných dát (modré body) a výsledná pozícia dát (červené body) po aplikácii 2D nekontrolovanej klasterizácii so zastavovacím kritérom v časových krokoch $n = 12, 22, 32, 42$.

3.3 Kontrolovaná klasifikácia

Okrem aplikácie pri klasterizácii sa model (2.12) môže jednoducho použiť aj pre účely klasifikácie dát kontrolovaným hlbokým strojovým učením. V tomto prípade, ale presnosť algoritmu nezávisí od rozloženia vstupných dát. Je to zapríčinené tým, že pri klasterizácii sa aplikovala iba dopredná difúzia, keďže dáta, s ktorými sme pracovali boli neoznačené. Pri klasifikácii pracujeme už s označenými dátami, pri ktorých vieme vopred povedať kam patrí ktorá hodnota, a teda vieme aplikovať aj spätnú difúziu v difúznom koeficiente g_e.

Aplikácia dopredno-spätnej difúzie znamená, že body, ktoré patria jednému klastru sa budú navzájom priťahovať, na základe difúzneho koeficienta (2.13) doprednej difúzie (Obr. 3.11 (napravo - bledo modré čiary)). Kým klastre medzi sebou, teda body v nich, sa budú pohybovať smerom od seba, s difúznym koeficientom (2.14) spätnej difúzie (Obr. 3.11 (napravo - červené čiary)).



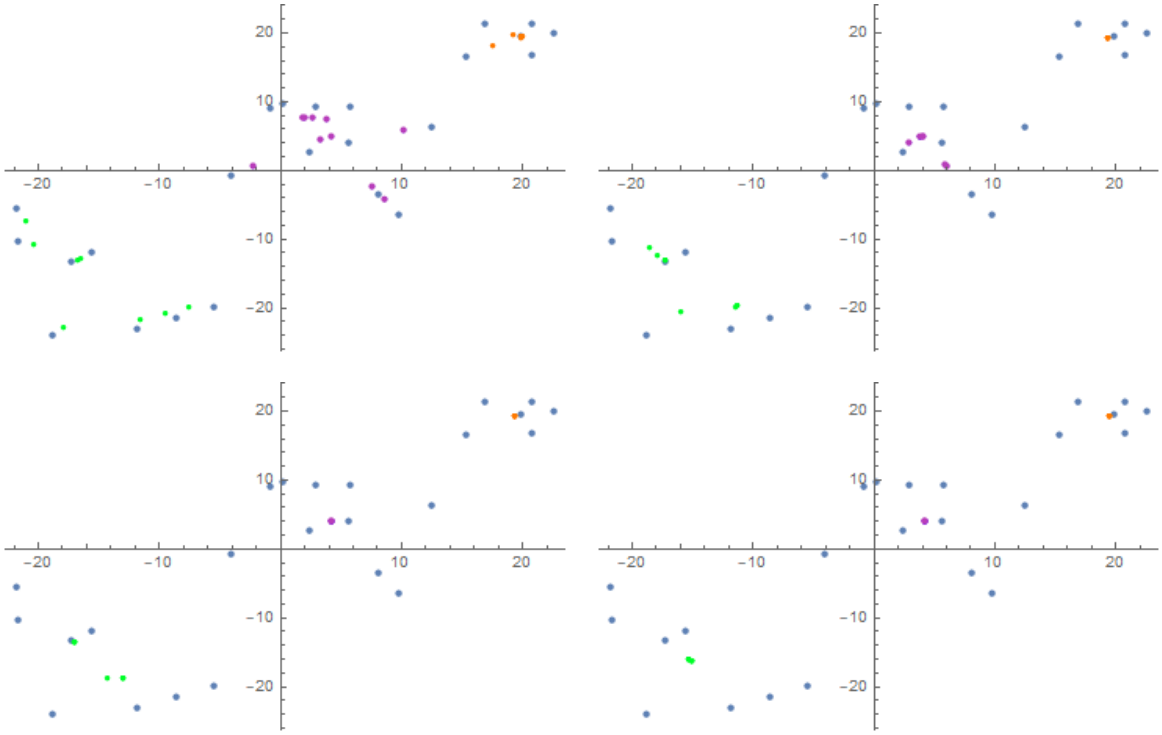
Obr. 3.11: Náhodne vygenerované 2D body do troch klastrov s farebným rozlíšením príslušnosti bodov v klastroch (naľavo). Na rovnakých dátach je znázornených pár väzieb doprednej difúzie (slabo modré čiary) a spätnej difúzie (červené čiary), ktorá prebieha medzi bodmi (napravo).

3.3.1 Kontrolovaná klasterizácia

Na začiatku klasifikácie bolo potrebné najskôr naučiť algoritmus vytvárať klastre, ale už z označených dát. Štandardne sme začali s jednoduchým cvičným príkladom, kedy sme si vygenerovali 2D body v troch intervaloch, teda získali sme body rozdelené do troch klastrov (Obr. 3.11 (naľavo)). Ku všetkým bodom sme pridali príslušnosť ku klastrom, čiže každému bodu sme priradili číslo klastra, ku ktorému patrí (na Obr. 3.11 jednotlivé klastre sú farebne rozlíšené). Algoritmus sme doplnili o spätnú difúziu (2.14), ktorá sa aplikovala vždy medzi bodmi z rôznych klastrov. Spustili sme kontrolovanú klasterizáciu, s hodnotou $K = 0.5$ v difúznom koeficiente, na $n = 200$ časových krokoch s veľkosťou kroku $\tau = 0.1$. Zastavovacie kritérium, ktoré sme použili pri nekontrolovanej klasterizácii sme použili aj pri kontrolovanej klasterizácii, teda ide o histogramové zastavovacie kritérium. Pre riešenie systému rovníc sme použili SOR iteračnu metódu s relaxačným parametrom $\omega = 1.25$ a s parametrom tolerancie $tol = 10^{-6}$.

Na obrázku (Obr. 3.12) vidíme ako sa postupne formujú klastre. Môžeme si všimnúť, že už v časovom kroku $n = 40$ sa vytvoril prvý klaster oranžových bodov a hneď v $n = 55$ sa sformoval aj klaster fialových bodov. V časovom kroku $n = 74$ sa vytvoril aj klaster zelených bodov a zastavila sa klasterizácia histogramovým zastavovacím

kritériom. Môžeme pozorovať, že aj napriek tomu, že sa dva klastre vytvorili skorej ako posledný klaster, klastre sa nepohybovali k sebe, ale ostali oddelené od seba. Tento stav nám zapríčinila spätná difúzia, ktorú sme aplikovali na body z rôznych klastrov. Kdežto, kým sme mali iba doprednú difúziu, aj po sformovaní klastrov by sa klastre pohybovali k sebe a vznikol by iba jeden veľký klaster.



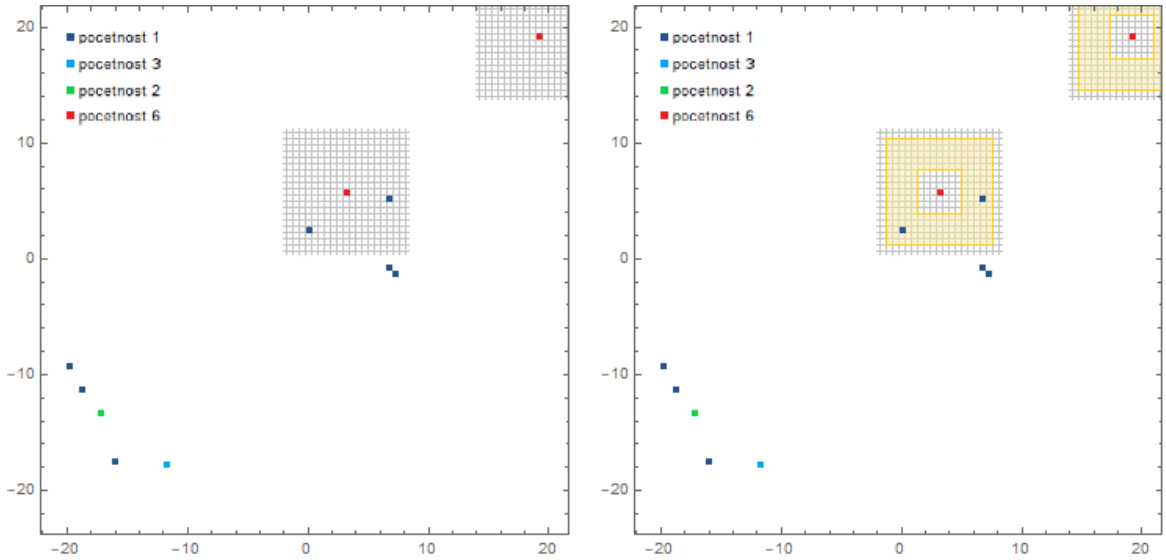
Obr. 3.12: Klasterizácia náhodne vygenerovaných 2D označených dát do klastrov s aplikáciou dopredno-spätnej difúzie v časových krokoch $n = 15, 40, 55, 74$.

3.3.2 Jednoduchý príklad kontrolovanej klasifikácie

Pri klasifikácii ide o to, aby sa novo prichádzajúca hodnota (newcomer) priradila ku správnemu klastru, ktorý sme vytvorili kontrolovanou klasterizáciou.

Proces učenia algoritmu sme začali zaraďovaním vopred označených bodov do klastrov, kedy sme použili 2D body vygenerované v príklade kontrolovanej klasterizácie (Obr. 3.11 (naľavo)). Keďže sme presne vedeli, ku ktorému klastru patrí ktorý bod, po jednom sme body označovali ako novo prichádzajúce hodnoty a sledovali sme, či sa správne zaradia. Na novo prichádzajúci bod, keďže sa tvárime, že nevieme kam sa má zaradiť, je aplikovaná iba nelineárna dopredná difúzia (2.13), takže všetky body ovplyvňujú pohyb novo prichádzajúceho bodu smerom k sebe - všetky body ho priťahujú

v závislosti od jeho vzdialenosti.

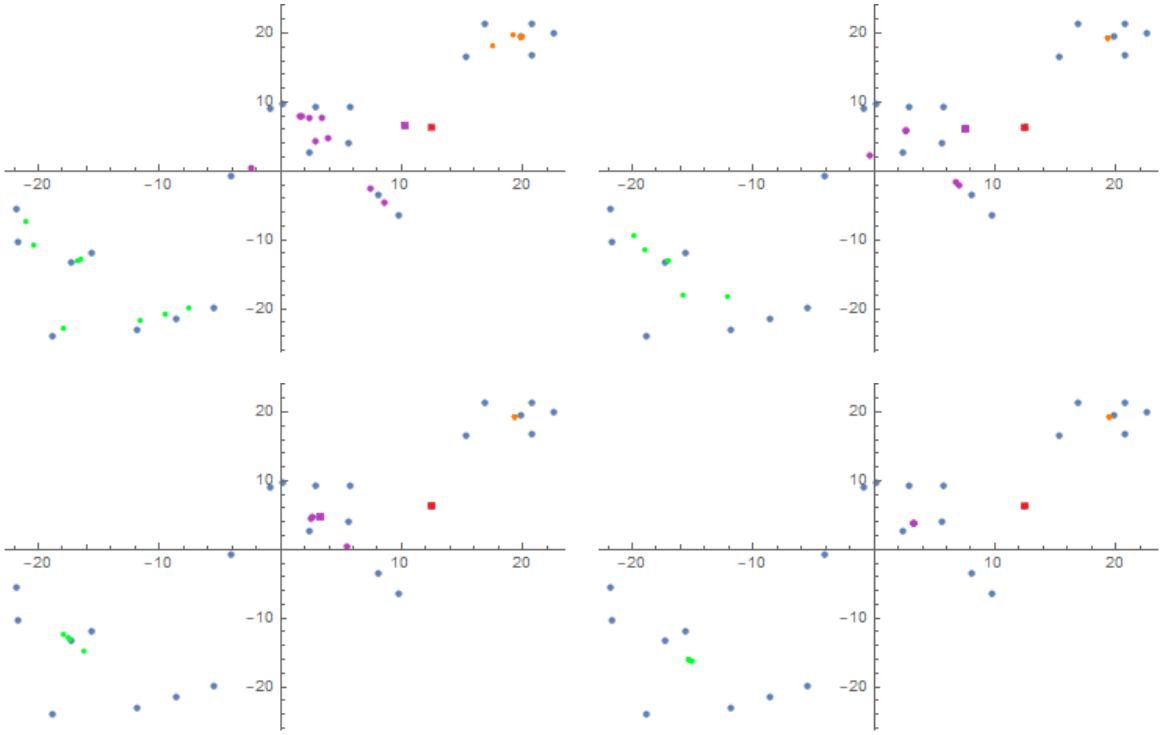


Obr. 3.13: 2D histogram pre pozmenené histogramové kritérium so *spacing* = 0.5 (naľavo) a vyznačené žltým rámikom okolie kontroly okolo kontrolovaného indexu (napravo), keďže v tomto príklade $minSize = 6$. Môžeme si všimnúť, že napravo hore vznikol klaster, keďže v jeho okolí nie sú žiadne hodnoty, ale klaster v strede sa ešte musí vyvíjať, keďže v oblasti kontroly sa nachádzajú dve hodnoty s početnosťou 1.

Pre potreby kontrolovanej klasifikácie sme histogramové zastavovacie kritérium museli pozmeniť. Teraz pre všetky časové kroky sa pri volaní *clusteringDone*() vypočíta 2D histogram, ktorý si vytvorí mriežku s krokom *spacing* (názorná ukážka 2D histogramu Obr. 3.13 (naľavo)). Zo vstupných dát vieme presne určiť, ktorý klaster je najmenší a zapamätáme si počet bodov v ňom do premennej *minSize*. Následne sledujeme vývoj histogramu a vždy keď sa v ňom vyskytne hodnota rovná alebo väčšia ako *minSize* označíme si tento index ako kontrolovaný index (Obr. 3.13 (napravo - červené body)). Overíme okolie kontrolovaného indexu, či sú v ňom nenulové hodnoty a v prípade ak nie sú, vyhlásime, že vznikol klaster. Pozeráme sa vždy v okolí 8×8, ale až od 3×3 oblasti (Obr. 3.13 (napravo - žltý rámik)), čo znamená, že v bezprostrednom okolí 3×3 klastru povoľujeme, aby boli nenulové hodnoty. Je to výhodné kvôli tomu, že celý systém je neustále v pohybe a môže nastať situácia, že niektoré hodnoty sa budú nachádzať presne na rozhraní mriežky pri indexe možného klastra, a v skutočnosti sú body už pri sebe. V prípade, ak by sme kontrolovali aj bezprostredné okolie okolo kontrolovaného indexu, mohol by sa už vytvorený zoskupený klaster v ďalšom časovom

kroku javiť ako nezoskupený. Histogramové kritérium zastaví vývoj v okamihu keď sa počet vytvorených klastrov bude rovnať počtu zadaných klastrov v podobe vstupných dát.

Keď je vývoj ukončený, pozrieme sa na pozíciu novo prichádzajúceho bodu. Vypočítame vzdialenosť od všetkých bodov a nájdeme tu najmenšiu. Ak je táto vzdialenosť menšia ako 0.05, tak novo prichádzajúci bod priradíme ku klastru bodu, ku ktorému je najbližšie. Ak je vzdialenosť väčšia ako 0.05 bod bude nezaradený. Hodnotu 0.05 sme si zvolili na základe pozorovaní, pri ktorých sa novo prichádzajúce body, pri tejto hodnote, rozumne zaradili. Keď bola hodnota menšia, bod, ktorý patrí k danému klastru sa k nemu nepriradil, lebo nebol dostatočne blízko, ale na animáciách sme videli, že je pri danom klastri, do ktorého sa mal zaradiť. A naopak, keď sme zvolili väčšiu hodnotu, tak sa stalo, že aj body, ktoré mali ostať nezaradené, sa priradili k najbližšiemu klastru.

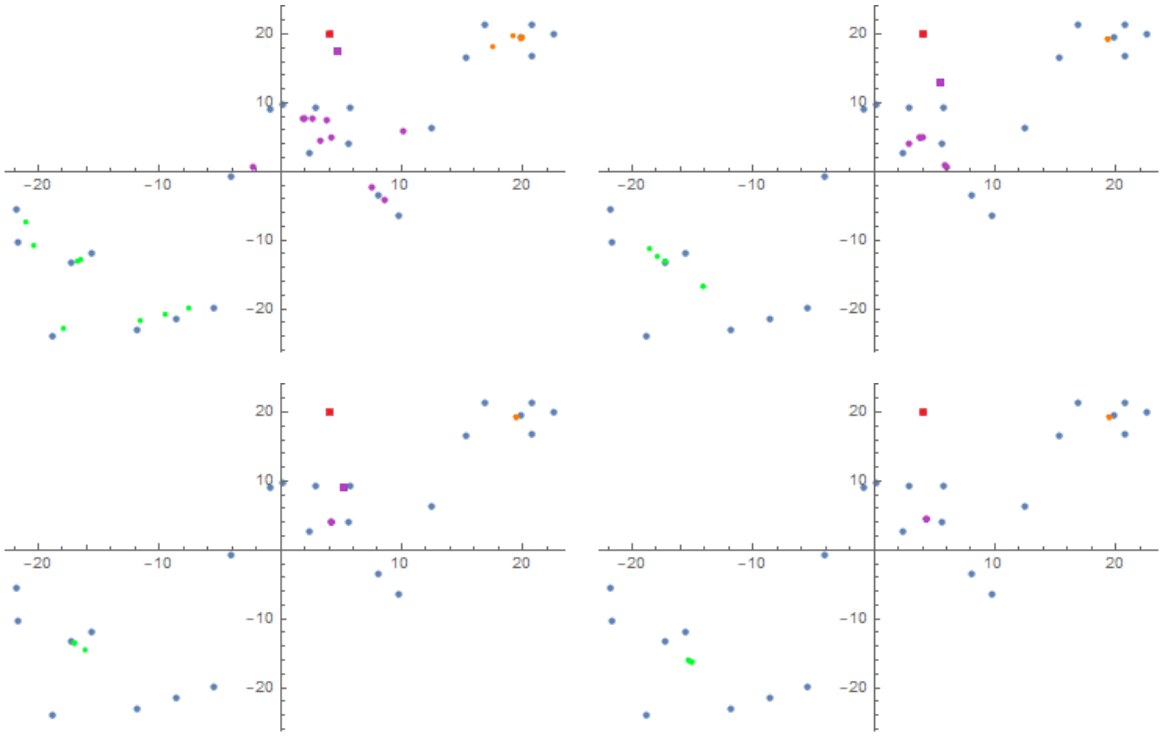


Obr. 3.14: Klasifikácia náhodne vygenerovaných 2D označených dát do klastrov s aplikáciou dopredno-spätnej difúzie v časových krokoch $n = 15, 30, 45, 74$ s jedným novo prichádzajúcim bodom (červený štvorček). Následne pri vývoji, novo prichádzajúci bod zmení farbu na farbu klastra, ktorému patrí, ale tvar štvorčeka mu ostane, aby sme vedeli sledovať jeho pohyb.

Vizuálne znázornenie zaraďovania bodov kontrolovanou klasifikáciou s jedným novo

prichádzajúcim bodom môžeme vidieť na obrázku (Obr. 3.14). Jeden bod zo vstupných dát sme označili ako novo prichádzajúci, znázornili sme ho červeným štvorčekom a pri nasledovnom vývoji je označený štvorčekom, ale má farbu klastra, ku ktorému sa nakoniec priradil. Môžeme vidieť ako sa pohybuje smerom ku klastru, ku ktorému naozaj patrí a to preto, že vzdialenosti od ostatných bodov toho klastru boli najmenšie, teda najsilnejšie ho priťahovali a následne sa daný bod zaradil správne.

Po úspešnom zaradení vopred označených bodov do klastrov sme skúsili pridať nový bod, ktorý algoritmus ešte nevidel, k už vygenerovaným bodom a sledovali sme kam sa zaradí.

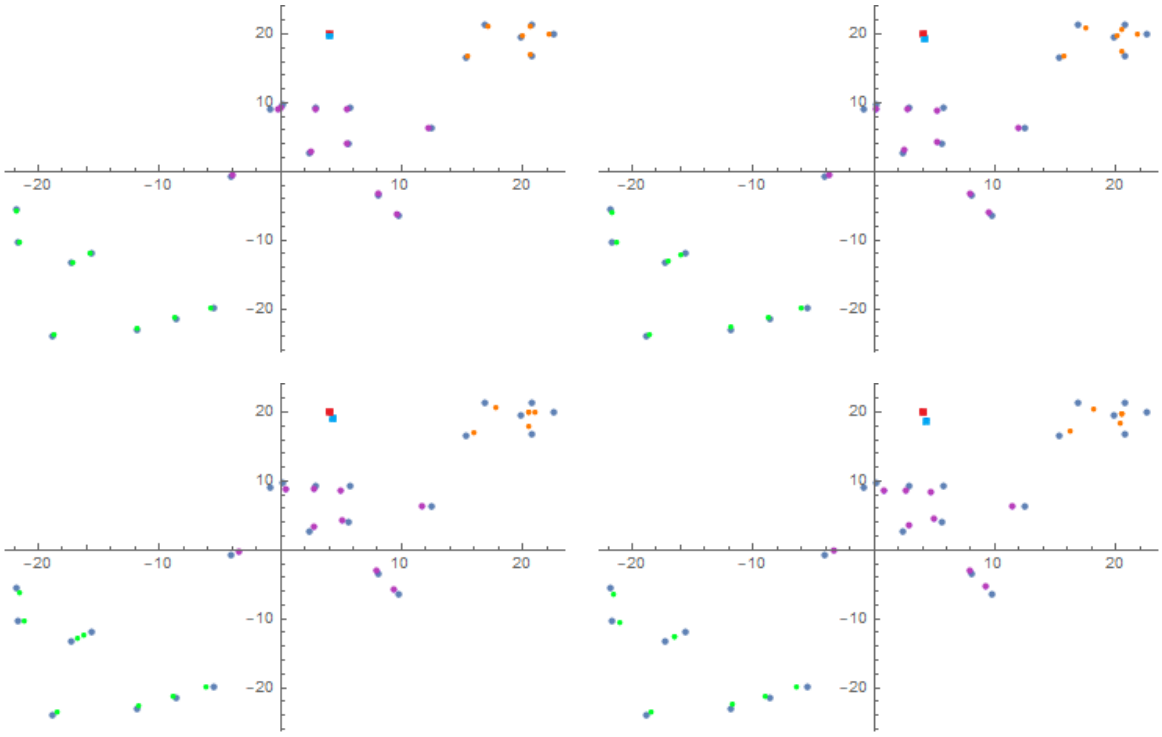


Obr. 3.15: Klasifikácia novo prichádzajúceho bodu (červený štvorček) s aplikáciou dopredno-spätnej difúzie s $K = 0.5$ v časových krokoch $n = 15, 40, 55, 74$.

Parametre, ktoré sme použili pri klasifikácii novo prichádzajúceho bodu sú rovnaké ako v predchádzajúcom popise. Na obrázku (Obr. 3.15) vidíme, že po pár krokoch algoritmu sa novo prichádzajúci bod zaradil ku klastru fialových bodov. To znamená, že aj keď ho rovnako priťahovali body všetkých klastrov, prevážili body fialového klastru, kvôli tomu, že boli k novo prichádzajúcemu bodu bližšie a sú aj početnejšie v porovnaní s bodmi z oranžového klastru.

Na tomto príklade zaraďovania novo prichádzajúceho bodu si môžeme ukázať aký

je dôležitý parameter K v difúznom koeficiente. Doteraz sme vždy pracovali s $K = 0.5$. Na vyššie popísaný príklad sme skúsili zväčšiť parametre K na hodnotu $K = 15$ a výsledok môžeme vidieť na obrázku (Obr. 3.16). V tomto prípade sa novo prichádzajúci bod (červený štvorček) nezaradil a preto je vyznačený v ďalšom vývoji bledo modrou farbou. Teda ak je váha K pri difúznom koeficiente veľká znamená to, že zmenšujeme vplyv vzdialenosti bodov od seba a vývoj bude pomalší, čo môže spôsobiť, že novo prichádzajúci bod bude nezaradený. V prípade ak zvolíme K príliš malé mohlo by sa stať, že aj body, ktoré sú priďaleko od klastrov sa rýchlo pritiahnu a nevhodne sa zaradia. Preto, ako neskôr uvidíme, je nesmierne dôležité nájsť správnu hodnotu váhy K, aby pri aplikácii na reálnych príkladoch neprišlo k nesprávnym zaradeniam a teda k zlým výsledkom.



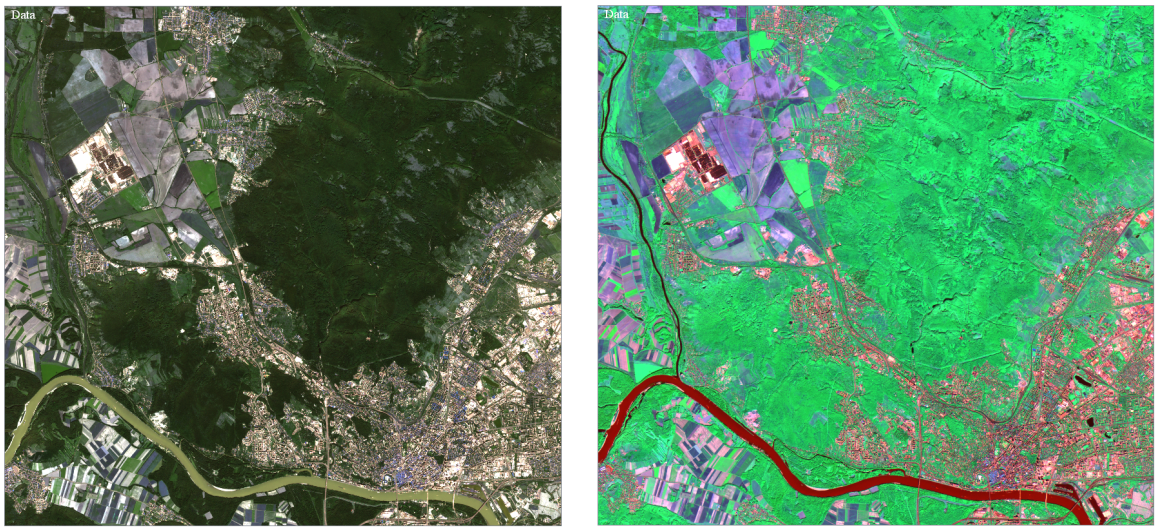
Obr. 3.16: Klasifikácia novo prichádzajúceho bodu (červený štvorček) s aplikáciou dopredno-spätnej difúzie s $K = 15$ v časových krokoch $n = 50, 100, 150, 200$.

Kapitola 4

Klasifikácia dát Natura 2000

Po ukončení implementačnej časti našej práce a po úspešnom testovaní algoritmu na umelo vytvorených dátach, sme algoritmus využili na klasifikáciu reálnych veľkorozmerných satelitných dát.

Európska vesmírna agentúra spustila program Copernicus v roku 2014, ktorý je zameraný na pozorovanie atmosféry, pevniny, morí a klímy na Zemi [2]. Hlavnými družicami programu Coprenicus sú družice Sentinel a dáta, ktoré budeme spracovávať, sú konkrétne z družíc Sentinel-2. Družice Sentinel-2 majú za cieľ, okrem iného, monitorovať poľnohospodárske plodiny, kvalitu vody a flóru na zemskom povrchu [19].



Obr. 4.1: Štandardný RGB obrázok získaný kombináciou B2, B3 a B4 kanálov (naľavo) a obrázok pre využitie v poľnohospodárstve z kombinácie B2, B8 a B11 kanálov (napravo).

Kanály	Rozlíšenie	Popis	Vlnová dĺžka
B1	60 m	Pobrežná aerosolová detekcia - *Coastal aerosol* (Ultra modré pásmo) - *(Ultra blue)*	443 nm
B2	10 m	Modré pásmo - *Blue*	490 nm
B3	10 m	Zelené pásmo - *Green*	560 nm
B4	10 m	Červené pásmo - *Red*	665 nm
B5	20 m	Klasifikácia vegetácie - *Vegetation classification* (Viditeľné a blízke infračervené pásmo) *(Visible and Near Infrared)*	705 nm
B6	20 m	Klasifikácia vegetácie - *Vegetation classification* (Viditeľné a blízke infračervené pásmo) *(Visible and Near Infrared)*	740 nm
B7	20 m	Klasifikácia vegetácie - *Vegetation classification* (Viditeľné a blízke infračervené pásmo) *(Visible and Near Infrared)*	783 nm
B8	10 m	Blízke infračervené pásmo - *Near Infrared*	842 nm
B8a	20 m	Klasifikácia vegetácie - *Vegetation classification* (Viditeľné a blízke infračervené pásmo) *(Visible and Near Infrared)*	865 nm
B9	60 m	Vyparovanie vody - *Water vapour* (Infračervené krátke vlny) - *(Short Wave Infrared)*	940 nm
B10	60 m	Rozlišovanie oblačnosti-ciry - *Cloud map* (Infračervené pásmo krátkych vĺn) *(Short Wave Infrared)*	1375 nm
B11	20 m	Rozlišovanie snehu/ľadu/oblačnosti *Snow/Ice/Cloud discrimination* (Infračervené pásmo krátkych vĺn) *(Short Wave Infrared)*	1610 nm
B12	20 m	Rozlišovanie snehu/ľadu/oblačnosti *Snow/Ice/Cloud discrimination* (Infračervené pásmo krátkych vĺn) *(Short Wave Infrared)*	2190 nm

Tabuľka 4.1: Tabuľka meraných kanálov z družíc Sentinel-2 s meraným rozlíšením, popisom každého kanálu a vlnovou dĺžkou kanálov [5].

Sentinel-2 tvoria dve sesterské družice Sentinel-2A a Sentinel-2B, ktoré sa nachádzajú na rovnakej obežnej dráhe s fázovým posunom 180°, čo umožňuje sledovať rovnaké

územie dvakrát častejšie. Družice sú vybavené prístrojmi na snímanie až dvanástich kanálov (Tab. 4.1), ktoré je možné rôzne kombinovať a získavať tak dáta na rozmanité účely. Každý z uvedených kanálov je nastavený na jednu z troch presností merania, buď je to 10 × 10 m na pixel, 20 × 20 m na pixel alebo 60 × 60 m na pixel. Kombináciou kanálov B2, B3 a B4 získame štandardný RGB obrázok povrchu Zeme (Obr. 4.1 (naľavo)). Kombináciou kanálov B2, B8 a B11 získame obrázok vhodný pre využitie v poľnohospodárstve, na ktorom môžeme monitorovať zdravie plodín (Obr. 4.1 (napravo)), napríklad vieme zvýrazniť hustú vegetáciu, ktorá sa javí ako sýto zelená farba [5].

Družica Sentinel-2 produkuje dva typy dát, ktoré sú už predspracované, teda urobené sú korekcie a odvodené sú ďalšie kanály, ktoré nie sú explicitne namerané snímačmi z družíc. Jeden typ je Level-1C, čo sú dáta monitorujúce vrchnú časť atmosféry, a typ Level-2A, čo sú dáta spodnej časti atmosféry [3]. V dátach typu Level-2A sú dopočítané štyri odvodené kanály, ktoré my použijeme, a uvedené sú v hornej časti tabuľky (Tab. 4.2 (hore)).

Odvodené kanály	Popis
Optická hrúbka aerosolov *(Aerosol Optical Thickness)*	Miera aerosolov (napr. častice dymu, púštny prach, ...)
Klasifikácia scenérie *(Scene Classification)*	Základná klasifikácia (napr. oblačnosť, pôdy/púšte, vody, ...)
Snehová mapa *(Snow map)*	Výskyt snehu
Mapa vyparovania - priemerné *(Scene-average Water Vapour map)*	Priemerné vyparovanie vody
Normalizovaný index rozdielnej vegetácie *(Normalized difference vegetation index)*	Ukazovateľ živej zelenej vegetácie

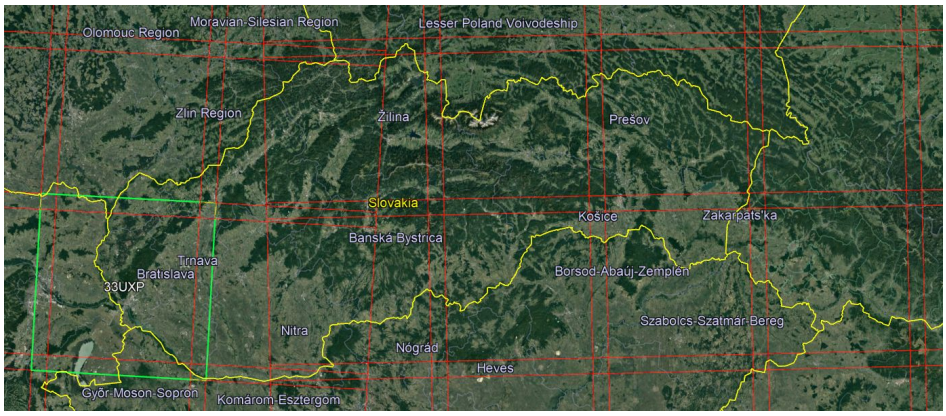
Tabuľka 4.2: Štyri odvodené kanály dopočítané Európskou vesmírnou agentúrou pre dáta získané z družíc Sentinel-2 (hore). Normalizovaný index rozdielnej vegetácie sme dopočítali my (dolu).

K odvodeným kanálom sme pridali ešte jeden odvodený kanál a to Normalizovaný

index rozdielnej vegetácie, ktorý kvantifikuje vegetáciu meraním rozdielu medzi takmer infračerveným žiarením, ktoré vegetácia silne odráža, a červeným svetlom, ktoré vegetácia absorbuje. Použili sme nasledovný vzťah

$$NDVI = \frac{B8 - B4}{B8 + B4},$$

v ktorom $NDVI$ je normalizovaný index rozdielnej vegetácie a B4 a B8 sú dva kanály družíc [16].



Obr. 4.2: Mapa Slovenska s vyznačenou oblasťou (zelený štvorček), z ktorej sú spracovávané dáta.

My pracujeme s dátami, ktoré mapujú územie Slovenska, konkrétne oblasť Západného Slovenska - Záhorie, Malé Karpaty a Podunajsko (Obr. 4.2). Na tejto časti boli semi-automaticky a automaticky vysegmentované oblasti chránených lesov v spolupráci s Botanickým ústavom Slovenskej akadémie vied [11, 10]. Na základe smernice o biotopoch sústavy Natura 2000 a posudku odborníkov zo Slovenskej akadémie vied, boli vysegmentované konkrétne štyri druhy dobre rozlíšiteľných biotopov na tomto území (Tab. 4.3).

Kód biotopu	Názov biotopu	Farba znázornenia
91E0	Lužné vŕbovo-topoľové a jelšové lesy	červená
91F0	Lužné dubovo-brestovo-jaseňové lesy okolo nížinných riek	modrá
91G0	Karpatské a panónske dubovo-hrabové lesy	žltá
9110	Kyslomilné bukové lesy	fialová

Tabuľka 4.3: Štyri segmentované biotopy zo sústavy Natura 2000 [15].

Na celej oblasti Západného Slovenska bolo vysegmentovaných stodvadsaťštyri chránených oblastí týchto štyroch biotopov zo snímky z dňa 10.09.2018.. Ukážku segmentácie uvedených biotopov môžeme vidieť na obrázku (Obr. 4.3), kde je znázornená priblížená menšia oblasť z územia Západného Slovenska a na nej sú vidno všetky štyri uvedené biotopy v podobe uzavretých kriviek okolo chránenej oblasti. Každá z označených kriviek je uložená vo formáte .kml a vieme si ju vždy nanovo otvoriť bez toho, aby sme museli znovu vyhľadávať a segmentovať chránenú oblasť.

Obr. 4.3: Priblížená malá oblasť Západného Slovenska s vyznačenými chránenými oblasťami biotopov podľa tabuľky (4.3). V pravom dolnom rohu je znázornená celá oblasť Západného Slovenska a prekrývajúcimi sa štvorčekmi (červený a modrý štvorček) je vyznačená zväčšená oblasť.

Dáta, ktoré budeme spracovávať, sú vytvorené zo všetkých vysegmentovaných chránených oblastí biotopov a so všetkých kanálov, ktoré sme vyššie popísali, a to tak, že sme počítali z každej vysegmentovanej chránenej oblasti biotopu a každého kanálu, strednú hodnotu, smerodajnú odchylku, minimálnu a maximálnu hodnotu (Obr. 4.4) a tak sme vytvorili priestor čŕt.

	91E0_bodiky_sever1	S2A_MSIL2A_20...		
	Area [m^2]	71559.054		
	Perimeter [m]	1380.418		
	IPR (Isoperimetric ratio)	2.119		
	Mean	Std	Min	Max
AOT-Aerosol ...	76	0	76	76
B01-Aerosol ...	246.238	41.9886	194	349
B02-Blue	236.299	46.1598	125	523
B03-Green	392.609	82.8258	177	689
B04-Red	245.444	51.5881	99	442
B05-Vegetation ...	616.227	132.46	270	1069
B06-Vegetation ...	1811.13	455.253	562	2973
B07-Vegetation ...	2183.23	547.322	554	3377
B08-Near ...	2269	682.702	181	3961
B09-Water ...	2145.71	507.135	1003	3012

Obr. 4.4: Jedna vysegmentovaná chránená oblasť biotopu s plochou v m^2, obvod v m a izoperimetrickým pomerom. Ďalej vidíme pár čŕt daného biotopu a ich stredné hodnoty, smerodajné odchýlky, minimálnu a maximálnu hodnotu.

Následne sme pre každú chránenú oblasť biotopu usporiadali črty tak, že za sebou boli najprv všetky stredné hodnoty, potom smerodajné odchýlky, minimálne hodnoty a nakoniec maximálne hodnoty (Obr. 4.5 (riadky v tabuľke)). Vytvorený súbor sme uložili vo formáte .csv. Takto nám vznikol pracovný priestor veľkosti 124×72, keďže vysegmentovaných chránených oblastí biotopov bolo stodvadsaťštyri, kanálov pre každý biotop bolo osemnásť a pre každý kanál sme počítali štyri hodnoty.

	A	B	C	D	E	F	G	H	I	J
1	124	72	4	23	24	38	39			
2	76	246.238	236.299	392.609	245.444	616.227	1811.13	2183.23	2269	2145.71
3	72.8933	254.864	249.033	423.445	274.693	697.712	2171.54	2644.61	2771.72	2732.64
4	71.4154	243.357	250.081	414.71	255.57	665.037	2028.84	2444.52	2542.19	2575.5
5	71	203.077	213.696	344.868	219.646	567.541	1773.11	2177.21	2228.07	2487.85
6	71	232	247.295	419.41	243.498	658.892	1991.4	2440.74	2614.24	2554.57
7	71	226	249.627	411.879	266.613	692.907	1973.99	2372.42	2509.33	2732.79
8	71	235.095	260.828	429.85	283.165	709.722	1865.8	2208.87	2320.07	2447
9	76	217.588	230.046	387.544	249.206	670.794	2034.53	2480.55	2591.31	2721.92
10	76	230.638	230.466	376.043	234.512	615.833	1950.42	2391.74	2521.43	2514.55
11	76	219.143	241.476	401.786	265.297	676.867	1904.53	2305.86	2421.14	2549
12	76	229.889	240.483	413.55	256.733	675.654	2010.19	2426.99	2559.6	2653.89
13	76	310.667	317.209	517.856	360.496	752.686	1710.26	2012.83	2221.67	1540.33
14	76	278.5	313.629	521.486	345.129	762.5	1805.85	2124.1	2433.6	1894.5
15	75.6819	218.19	233.178	407.037	266.414	695.509	2017.49	2437.8	2546.79	2623.55

Obr. 4.5: Pár hodnôt zo spracovávaných dát zobrazených v Exceli. Prvý riadok predstavuje hlavičku, v ktorej sa nachádza počet riadkov, počet stĺpcov, množstvo klastrov a početnosti v jednotlivých klastroch.

Keď si chceme spraviť porovnanie s umelými príkladmi popisovanými v kapitole 3, vysegmentované chránené oblasti biotopov predstavujú body a súradnice týchto bodov

sú jednotlivé črty, teda stredná hodnota, smerodajná odchýlka, minimálna a maximálna hodnota vo všetkých kanáloch. Tým pádom získavame sedemdesiatdva rozmerný priestor. Predstaviť si takto veľký priestor je nemožné a počítať v ňom je obtiažne a príliš zdĺhavé a preto sme na tento veľkorozmerný priestor aplikovali analýzu hlavných komponentov, čím sme zredukovali dimenziu a získali sme dvojrozmerný priestor. Pracovný priestor sa zmenšil na 124×2 a s takýmto priestorom sa omnoho rýchlejšie a ľahšie pracovalo, už len preto, že dvojrozmerný priestor si vieme predstaviť a veľmi jednoducho aj vizualizovať.

4.1 Učenie a validácia siete

Segmentácia chránených oblastí biotopov a výpočet ich čŕt bola vykonaná v softvéri NaturaSat, ktorý je vyvíjaný v spolupráci s Európskou vesmírnou agentúrou ESA. Keďže náš algoritmus bude nadstavbou softvéru NaturaSat, celý algoritmus sme prepísali do programovacieho jazyku C++, pričom sme sa sústredili na optimalizáciu celého kódu. Aby sme dosiahli čo najväčšiu možnú rýchlosť, kód sme sparalelnili pomocou OpenMP.

Vysegmentovaných bolo stodvadsaťštyri chránených oblastí biotopov a z toho boli dvadsaťtri Lužné vŕbovo-topoľové lesy (biotop 91E0), dvadsaťštyri Lužné dubovo-brestovo-jeseňové lesy (biotop 91F0), tridsaťosem Dubovo-hrabové lesy (biotop 91G0) a tridsaťdeväť Kyslomilné bukové lesy (biotop 9110). Biotopy uvedené v tabuľke (Tab. 4.3), predstavujú štyri klastre a každý bod, teda vysegmentovaná chránená oblasť biotopov, má priradený jeden konkrétny klaster, teda biotop, ku ktorému patrí. Vytvorili sme päť datasetov (Tab. 4.4), na ktorých sme spustili najprv učiacu fázu. Výber konkrétnych chránených oblastí biotopov do datasetov nebol náhodný. Hľadali sme práve také chránené oblasti biotopov, ktoré boli dobre odlíšiteľné od ostatných, aby sme algoritmus čo najlepšie naučili rozpoznávať charakteristiky uvedených biotopov.

Názov datasetu	Rozmer	Výber lesov zo štyroch uvedených biotopu
Dataset16	16×72	vybrali sme z každého typu biotopu štyri lesy
Dataset32	32×72	vybrali sme z každého typu biotopu osem lesov
Dataset48	48×72	vybrali sme z každého typu biotopu dvanásť lesov
Dataset64	64×72	vybrali sme z každého typu biotopu šestnásť lesov
Dataset80	80×72	vybrali sme z každého typu biotopu dvadsať lesov

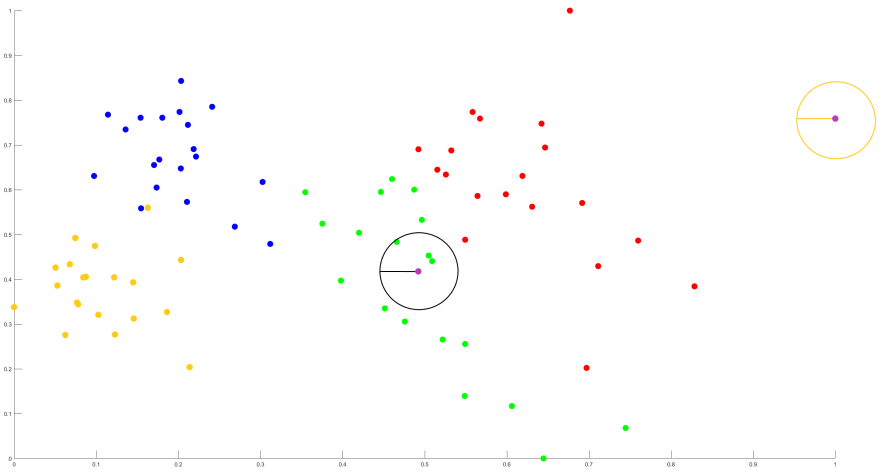
Tabuľka 4.4: Päť vytvorených datasetov, na ktorých sme trénovali algoritmus.

Keďže cieľom je, aby náš algoritmus vedel pracovať s akýmkoľvek typom dát, rozhodli sme sa vstupné dáta preškálovať do intervalu $[0, 1]$, čím zabezpečíme univerzálnosť algoritmu. Preškálovaním dosiahneme, že pri akomkoľvek type dát budú platiť rovnaké rozpätia vstupných parametrov. Napríklad pri histograme si vždy po preškálovaní vstupných dát do intervalu $[0, 1]$ a pri $spacing = 0.01$ budeme istý, že interval je rozdelený na 100 rovnakých podintervalov.

Spracovávané dáta, ako sme už spomínali, sú veľkorozmerné dáta a obsahujú až sedemdesiatdva dimenzií. Aby sme nemuseli pracovať s takýmto veľkým priestorom, na vstupné dáta aplikujeme analýzu hlavných komponentov (Podčasť 2.2.4), pomocou ktorej zmenšíme rozmer, a budeme pracovať s dátami, ktoré sú dvojrozmerné. Riešiť budeme dva systémy rovníc, teda jeden pre každú súradnicu.

V prípade reálnych dát sme si všimli, že v dôsledku potenciálnej nerovnomernej početnosti bodov v klastroch, môže vzniknúť situácia kedy novo prichádzajúci bod, ktorého počiatočná poloha je na rozmedzí viacerých klastrov, je automaticky priťahovaný tým klastrom, ktorého početnosť bodov je najvyššia. Tento problém sme vyriešili lokalizáciou vplyvu doprednej difúzie na novo prichádzajúci bod tak, že sme do algoritmu pridali podmienku, že na novo prichádzajúci bod budú pôsobiť iba body, pre ktoré je hodnota difúzneho koeficientu väčšia ako hodnota parametru δ. Po tejto úprave vznikne pomyselné okolie okolo novo prichádzajúceho bodu, nazývajme ho δ-okolie, v ktorom je novo prichádzajúci bod priťahovaný iba bodmi z tohto okolia a vplyv ostatných bodov je nulový (názorná ukážka okolia vplyvu na Obr. 4.6). Vhodnou voľbou parametra δ vieme zabezpečiť aj to, že ak je bod príliš vzdialený od všetkých klastrov, teda v jeho okolí sa nevyskytujú žiadne body, čo zapríčiní, že žiaden bod nemá hod-

noty difúzneho koeficientu väčšiu ako hodnota δ, a tak tento bod zostane nezaradený. Naskytuje sa nám otázka, ako zvoliť hodnotu δ, aby sme dostali optimálne výsledky.



Obr. 4.6: Vstupné dáta, na ktorých sme vyznačili okolie vplyvu doprednej difúzie pri dvoch bodoch. Pri bode v strede (okolie znázornené čiernym krúžkom) vidíme, že jeho ovplyvňujú tri zelené body. Pri bode napravo (znázornené žltým krúžkom) vidíme, že na neho nevplýva žiaden bod a tento bod zostane nezaradený.

V učiacej fáze si nájdeme optimálnu hodnotu parametra δ, pre vhodné δ-okolie, ako aj optimálnu hodnotu parametra difúzneho koeficienta K. Keďže pracujeme s vopred označenými bodmi, vieme presne, ku ktorému klastru patrí ktorý bod a znovu budeme postupne po jednom označovať body ako novo prichádzajúce a budeme sledovať či sa zaradia do správneho klastru. Zadávať budeme hodnoty parametrov v rozpätí pre parameter difúzneho koeficienta $K = [100, 20000]$ s krokom $Kstep = 100$, a rozpätie parametra δ, pre δ-okolie, v rozpätí $\delta = [0.001, 0.1]$ s krokom $\delta step = 0.001$. Na vstupné dáta aplikujeme $n = 200$ časových krokov nelineárnej dopredno-spätnej difúzie a budeme vyhodnocovať, že pri ktorej kombinácii parametrov získame najväčšiu úspešnosť. Pre numerické riešenie systému rovníc použijeme, ako aj v minulých prípadoch, SOR iteračnú metódu s relaxačným parametrom $\omega = 1.25$, ktorý sme znovu získali experimentálne, a toleranciou konvergencie $tol = 10^{-6}$.

Pri takomto priestore parametrov a uvažujúc, že každý bod označíme ako novo prichádzajúci nám vznikne veľa iterácii a celkový proces učenia siete bude časovo a výpočtovo náročný. Preto je potrebné paralelniť algoritmus a to tak, že každému jadru procesoru priradíme všetky hodnoty parametra K a parameter δ rozdistribuujeme medzi jednotlivé jadrá. Musíme však dávať pozor, ktoré premenné sú zdieľané,

teda definované pred blokom kódu, ktorý paralelníme, a ktoré nezdieľané, teda jednotlivé jadrá si ich dopočítajú sami.

Výsledky učiacej fázy sú uvedené v tabuľke (Tab. 4.5), kde si môžeme všimnúť nárast úspešnosti so zväčšovaním počtu dát v datasete. Najlepší výsledok sme dosiahli pri Datasete80, až 92.5% úspešne zaradených a preto pri validačnej fáze budeme vychádzať z tohto datasetu. Takýto výsledok sa nám poradilo získať pre pätnásť sád parametrov difúzneho koeficienta K a parameter δ-okolia δ.

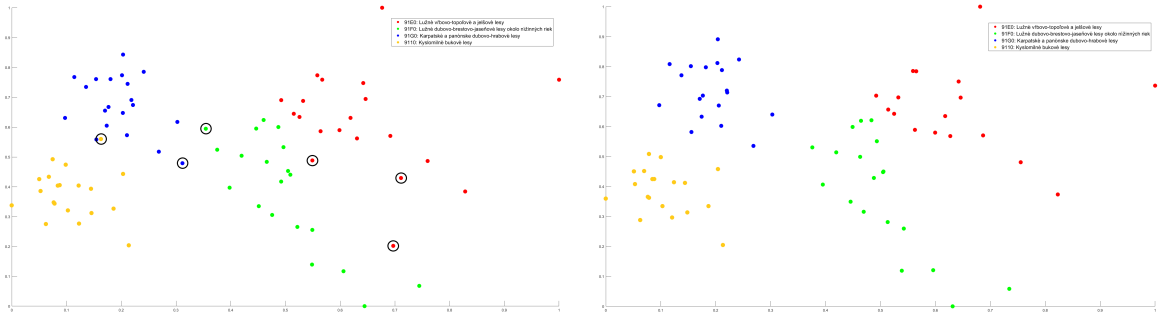
Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset16	9	4	3	56.25%
Dataset32	24	7	1	75%
Dataset48	38	7	3	79.17%
Dataset64	55	9	0	85.93%
Dataset80	74	6	0	92.5%

Tabuľka 4.5: Výsledky učiacej fázy na reálnych dátach, v prípade piatich datasetov so zaznamenaným precentuálnym podielom úspešnosti.

4.1.1 Prvá verzia validácie

Pri dosiahnutí úspešnosti 92.5% v Datasete80 sa vyskytlo šesť bodov, ktoré sa zle zaradili. Aby nám tieto zle zaradené body nekazili výsledky validácie, rozhodli sme sa ich odstrániť z datasetu (Obr. 4.7 (naľavo)). Naša úvaha bola, že keď sa týchto šesť bodov nezaradí správne do klastru, ku ktorému patrí, ich črty sú na pomedzí a mohli, by spôsobiť, že novo nameraná hodnota by sa priradila k nesprávnemu klastru iba preto, že hodnoty na pomedzí ju ovplyvňovali. Zistili sme, že z týchto šesť hodnôt tri patria do červeného klastru (biotop 91E0) a po jednom do zvyšných troch klastrov. Po odstránení zle zaradených bodov nám vznikne nepomer počtu bodov v jednotlivých klastrov, ale vďaka rozšíreniu algoritmu o lokalizáciu vplyvu môžeme tento nepomer zanedbať. Vytvorili sme nový Dataset74, ktorý bol ochudobnený o šesť zle zaradených bodov (Obr. 4.7 (napravo)). Body v novom datasete sme rovnako ako pôvodné body preškálovali do intervalu $[0, 1]$ a aplikovali sme na ne analýzu hlavných komponentov,

aby sme veľkorozmerný priestor zmenšili na dvojrozmerný priestor.



Obr. 4.7: Dataset80 po preškálovaní do $[0, 1]$ s označenými šiestimi odstránenými bodmi (naľavo) a vytvorený nový Dataset74 po preškálovaní do $[0, 1]$ (napravo).

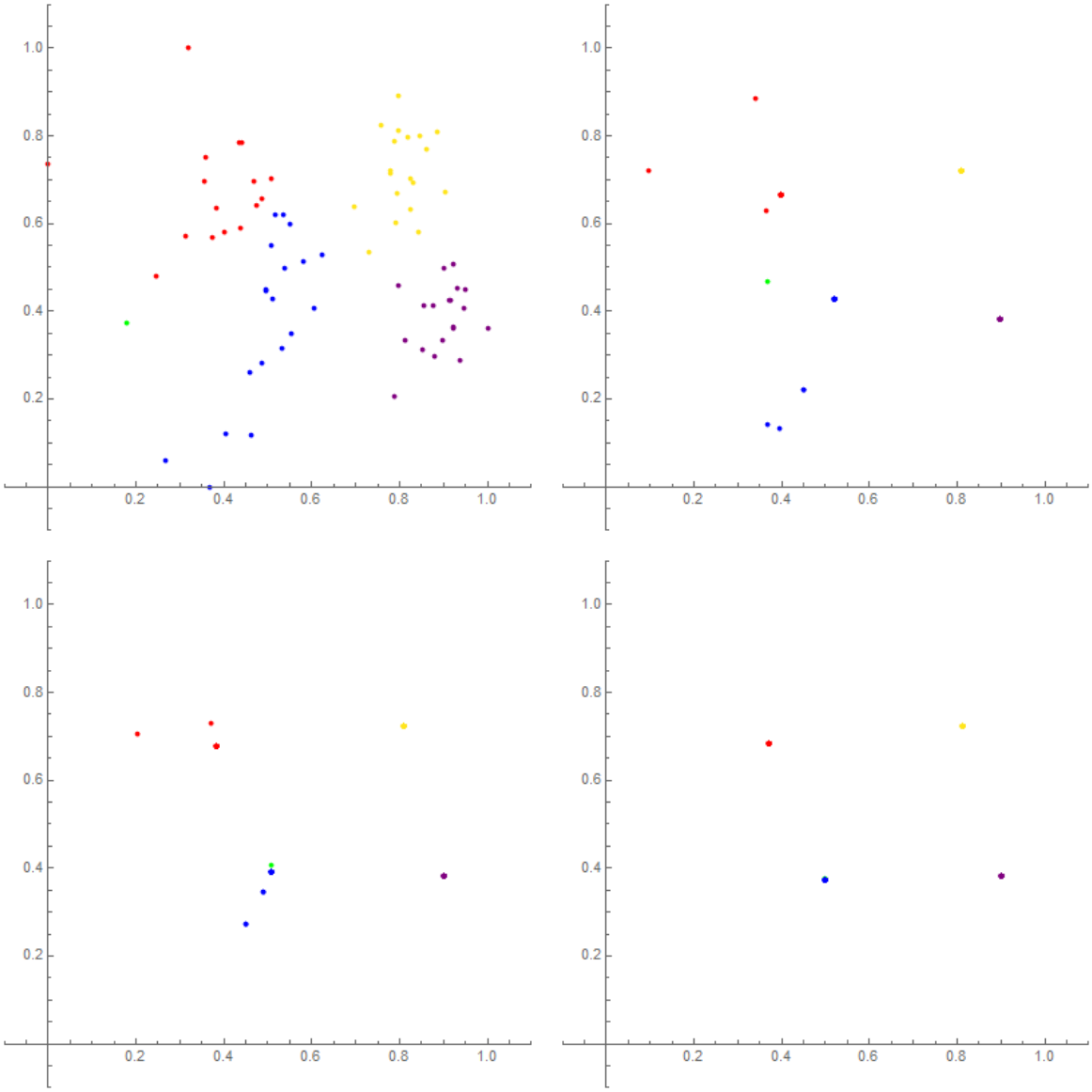
Keďže vznikol nový dataset, spustili sme učiacu fázu s rozpätím parametra difúzneho koeficienta $K = [100, 20000]$ s krokom $Kstep = 100$ a s rozpätím parametra δ-okolia $\delta = [0.001, 0.1]$ s krokom $\delta step = 0.001$. Očakávali sme, že po vytvorení nového datasetu, ktorý neobsahuje zle zaradené body výsledok by mal mať 100% úspešnosť, teda zaradených 74/74 bodov.

Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset74	72	1	1	97.29%

Tabuľka 4.6: Výsledky učiacej fázy na Datasete74.

Pri učiacej fáze Datasetu74 sa nám podarilo nájsť dvadsaťosem sád optimálnych parametrov, kedy sa úspešne zaradilo iba 72/74 bodov (Tab. 4.6).

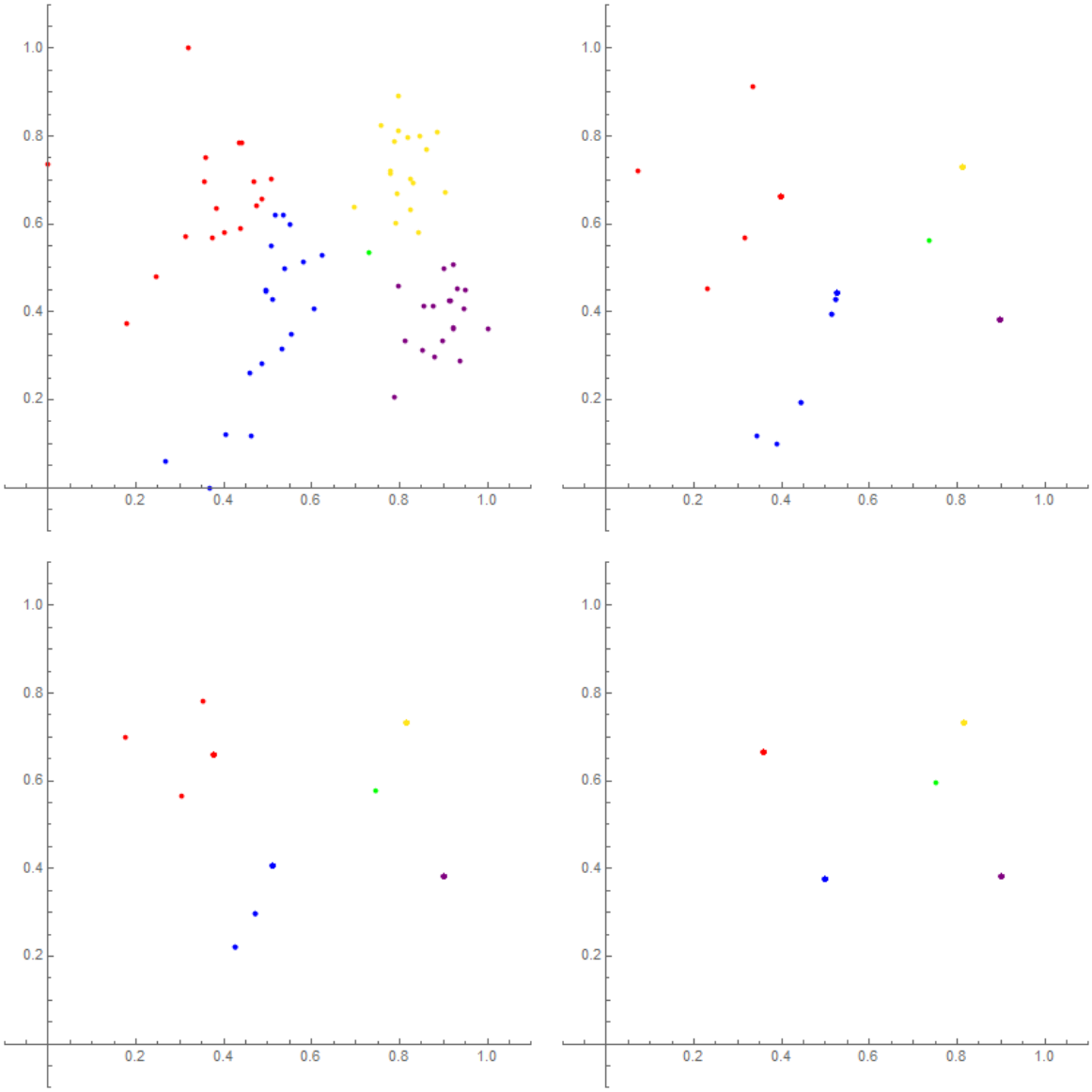
Nesprávne sa zaradil jeden bod a keď sa pozrieme na obrázok (Obr. 4.8) všimneme si, ako sa novo prichádzajúci bod správa. Najskôr sa bod očakávane pohybuje smerom do stredu medzi červený a modrý klaster, ale následne prevládol vplyv modrého klastra, ktorý ho nakoniec pritiahne k sebe. Preto sa naskytuje otázka, či je správne odstrániť hodnoty, ktoré sa v učiacej fáze na Datasete80 zle zaradili.



Obr. 4.8: Klasifikácia Datasetu74 s jedným bodom, pôvodne z červeného klastra, označeným ako novo prichádzajúci bod (zelený bod) v časových krokoch $n = 0, 13, 23, 33$. Použité optimálne parametre $K = 5100$ a $\delta = 0.001$, pri ktorých sa novo prichádzajúci bod zle zaradí.

Pri učiacej fáze na Datasete74 sa jeden bod nedokázal zaradiť (Obr. 4.9). V tomto prípade, keď sa pozrieme na vývoj vidíme, že novo prichádzajúci bod, pôvodne zo žltého klastra, sa nachádza medzi tromi klastrami. To znamená, že v priamočiarom pohybe k svojmu klastru mu bránia body modrého klastra, ktoré sa ho snažia pritiahnuť k sebe, a tiež aj body fialového klastra. Je vidieť, že bod sa na začiatku pohybuje minimálne lebo naňho pôsobia všetky okolité klastre. Až po viacerých časových krokoch nelineárnej difúzie, keď sa ostatné body zoskupia, novo prichádzajúci bod sa začne pohybovať

smerom k žltému klastru. V poslednom $n = 30$ časovom kroku sa celý proces zastaví, pretože je splnená podmienka histogramového zastavovacieho kritéria a algoritmus vyhodnotí zaraďovaciu podmienku. Keďže v blízkom okolí 0.05 novo prichádzajúceho bodu sa nenachádza žiaden z klastrov, tento bod bude označený ako nezaradený.



Obr. 4.9: Klasifikácia Datasetu74 s jedným bodom, pôvodne zo žltého klastra, označeným ako novo prichádzajúci bod (zelený bod) v časových krokoch $n = 0, 10, 20, 30$. Použité optimálne parametre $K = 5100$ a $\delta = 0.001$, pri ktorých sa novo prichádzajúci bod nezaradí.

Tu sa naskytuje ďalšia úvaha, ktorú bude potrebné riešiť v ďalšom výskume. V prípade učiacej fázy vieme kde sa má bod zaradiť a je možné upraviť zastavovacie kritérium tak, že ak má novo prichádzajúci bod tendenciu ísť smerom ku klastru, necháme systém

vyvíjať až kým sa bod nezaradí. Ale ak by sme mali úplne nove pozorovanie, pri ktorom vopred nevieme kam bod patrí a poznáme iba jeho črty, môžeme touto úpravou spôsobiť, že bod nasilu zaradíme do nesprávneho klastra.

Aj napriek nie 100% úspešnosti, sme algoritmus podrobili validácii a to tak, že sme mu ako novo prichádzajúce body dali zvyšné body z pôvodných dát 124×72, ktoré neboli obsiahnuté v Datasete80. Mohli sme si to dovoliť, keďže algoritmus tieto dáta nevidel a pre neho to boli úplne nové pozorovania. My sme ale presne vedeli kam sa majú tieto body zaradiť, teda vedeli sme vyhodnotiť úspešnosť siete. Vytvorili sme dva nové datasety novo prichádzajúcich dát (Tab. 4.7). Dataset12 sme vytvorili preto, aby sme algoritmus testovali na rovnakom počte (náhodne) vybraných nových bodov z každého biotopu a Dataset44 pozostáva zo všetkých zvyšných bodov, ktoré netvoria Dataset80, a teda Dataset44 má výrazne rôznu početnosť bodov z rôznych biotopov.

Názov datasetu	Rozmer	Výber oblastí zo štyroch uvedených biotopu
Dataset12	12×72	vybrali sme z každého typu biotopu tri oblasti, ktoré sa nenachádzajú v Datasete80
Dataset44	44×72	vybrali sme všetky zvyšné oblasti, ktoré sa nenachádzajú v Datasete80

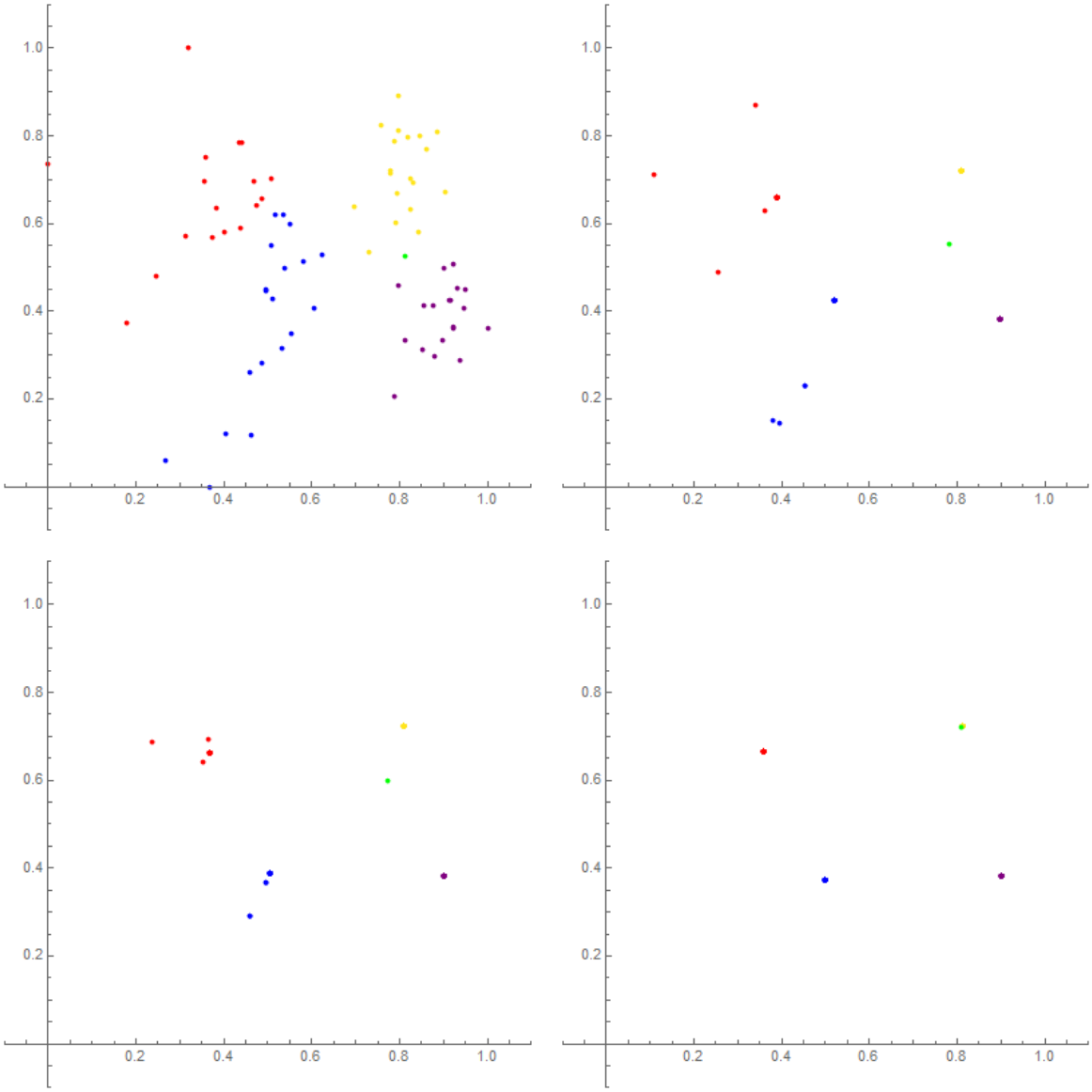
Tabuľka 4.7: Dva datasety novo prichádzajúcich bodov, na ktorých sme validovali algoritmus.

Proces validácie prebiehal nasledovne, algoritmus dostal na vstup jednu sadu optimálnych parametrov, ktoré sme našli v trénovacej fáze, $K = 5100$ a $\delta = 0.001$, na vstup dostal tiež Dataset74, na ktorom bol trénovaný. Dáta z Datasetu74 preškáloval do intervalu $[0, 1]$, pričom si zapamätá škálovacie parametre, a aplikoval analýzu hlavných komponentov. Z analýzy hlavných komponentov si zapamätal transformačnú maticu. Na dataset novo prichádzajúcich bodov sa najprv aplikuje škálovanie so škálovacím faktorom z Datasetu74 a potom sa aplikuje analýza hlavný komponentov s transformačnou maticou z Datasetu74. Následne sa z datasetu novo prichádzajúcich bodov vyberá po jednom bode a aplikuje sa kontrolovaná klasifikácia, kedy sa snažíme zaradiť novo prichádzajúcu hodnotu k už existujúcim klastrom. Zhodnotenie výsledkov je dané v tabuľke (Tab. 4.8).

Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset12	9	2	1	75%
Dataset44	31	11	2	70.45%

Tabuľka 4.8: Výsledky validačnej fázy na Datasete74 pri novo prichádzajúcich bodoch z Datasetu12 a Datasetu44.

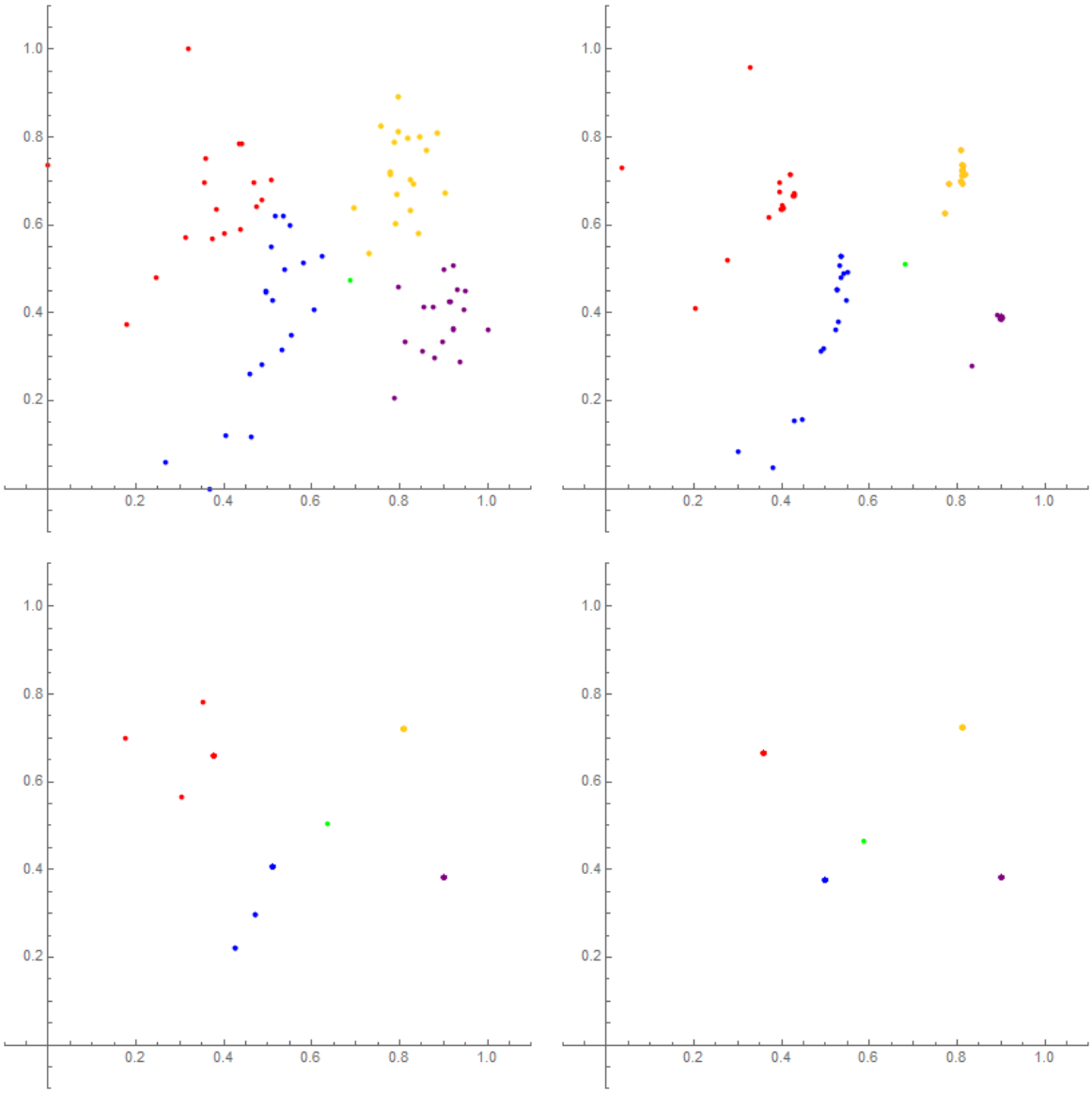
Keď sme si pozreli ako prebiehal vývoj každého z novo prichádzajúcich bodov z Datasetu12, zistili sme, že jeden nezaradený bod sa nezaradil, kvôli rovnakému dôvodu ako na obrázku (Obr. 4.9). V prípade zle zaradených novo prichádzajúcich bodov je dôvod to, že sa nachádzajú presne na rozhraní medzi dvoma klastrami. Ukážku zle zaradeného bodu z Datasetu12 môžeme vidieť na obrázku (Obr. 4.10). Novo prichádzajúci bod, pôvodne z fialového klastra, sa nachádza na rozhraní žltého a fialového klastra. Pri ďalšom vývoji sa pomaly začne pohybovať k žltému klastru a je to spôsobené tým, že na začiatku ma okolo seba viac žltých bodov ako fialových. Žlté body sú bližšie a teda ho silnejšie priťahujú, čo spôsobí, že sa tento bod zle zaradí. Ako si neskôr povieme, takto umiestnený bod bude mať veľmi malú relevanciu správneho zaradenia.



Obr. 4.10: Klasifikácia Datasetu74 s novo prichádzajúcimi bodmi z Datasetu12 v časových krokoch $n = 0, 14, 24, 34$. Vyznačený jeden novo prichádzajúci bod (zelený bod), ktorý pôvodne patril fialovému klastru, ale po vývoji sa zle zaradí.

Pri Datasete44 sa správne zaradilo 31 zo 44 bodov. Po detailnom rozanalyzovaní spôsobu zaraďovania sme zistili, že väčšina bodov, ktoré sa zaradili zle sa nachádzala na rozhraní medzi dvomi klastrami, podobne ako na obrázku (Obr. 4.10). Vyskytli sa aj dva nezaradené body, z ktorých sa jeden nezaradil kvôli rovnakému dôvodu ako na obrázku (Obr. 4.9), a vývoj druhého nezaradeného bodu je znázornený na obrázku (Obr. 4.11). Rozloženie bodov na začiatku sa veľmi podobá minulému príkladu nezaradeného bodu, znovu je novo prichádzajúci bod v strede medzi tromi klastrami. Po pár krokoch, ako môžeme vidieť sa veľmi nepohol z miesta, pretože ho na začiatku všetky

tri klastre priťahujú rovnako. Následne ale môžeme vidieť ako sa bod rozhodne ísť k modrému klastru, no nestačí prísť natoľko blízko, aby po zastavení vývoja bol priradený k modrému klastru.

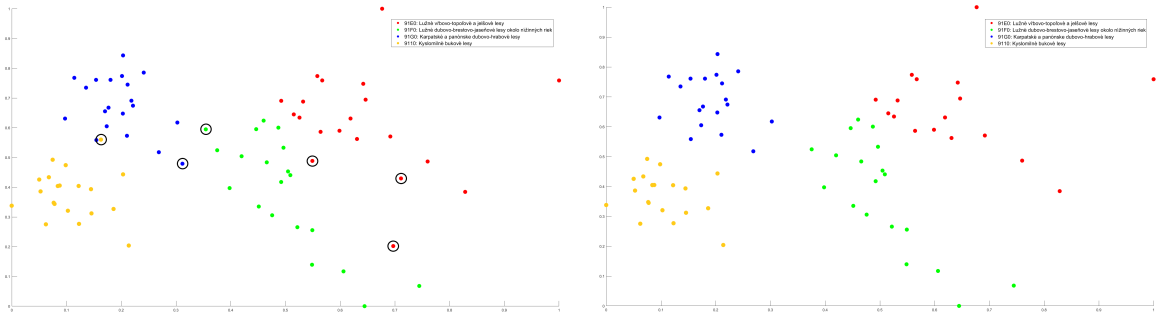


Obr. 4.11: Klasifikácia Datasetu74 s novo prichádzajúcimi bodmi z Datasetu44 v časových krokoch $n = 0, 5, 20, 30$. Vyznačený jeden novo prichádzajúci bod (zelený bod), ktorý pôvodne patril žltému klastru, ale po vývoji zostane nezaradený.

Tento konkrétny bod pôvodne patrí žltému klastru, teda, ak by sa priradil k modrému klastru radili by sme ho do zle zaradených bodov. V tomto prípade bolo správne, že bod ostal ako nezaradený. Z tohto uhla pohľadu úvaha, ktorú sme mali o ponechaní procesu ešte dlhšie bežať sa teraz ukazuje ako nesprávna. Je to naozaj otvorená otázka, ktorej sa budeme venovať v ďalšom výskume.

4.1.2 Druhá verzia validácie

V druhej verzii sme pokračovali v myšlienke, že z Datasetu80 odstránime šesť zle zaradených bodov, ale preškálovanie a analýzu hlavných komponentov budeme robiť na základe Datasetu80. Najprv preškálujeme dáta z Datasetu80, aplikujeme na nich analýzu hlavných komponentov, pričom zmenšíme dimenziu dát na dvojrozmernú. Z takto upraveného Datasetu80 odstránime šesť zle zaradených hodnôt a vznikne Dataset74A. Takýto spôsob tvorby nového datasetu by mohol byť vhodnejší, pretože iba odstránime zle zaradené body, pričom nezmeníme polohu zvyšných bodov (Obr. 4.12).



Obr. 4.12: Dataset80 po preškálovaní do $[0, 1]$ s označenými šiestimi odstránenými bodmi (naľavo) a vytvorený nový Dataset74A (napravo).

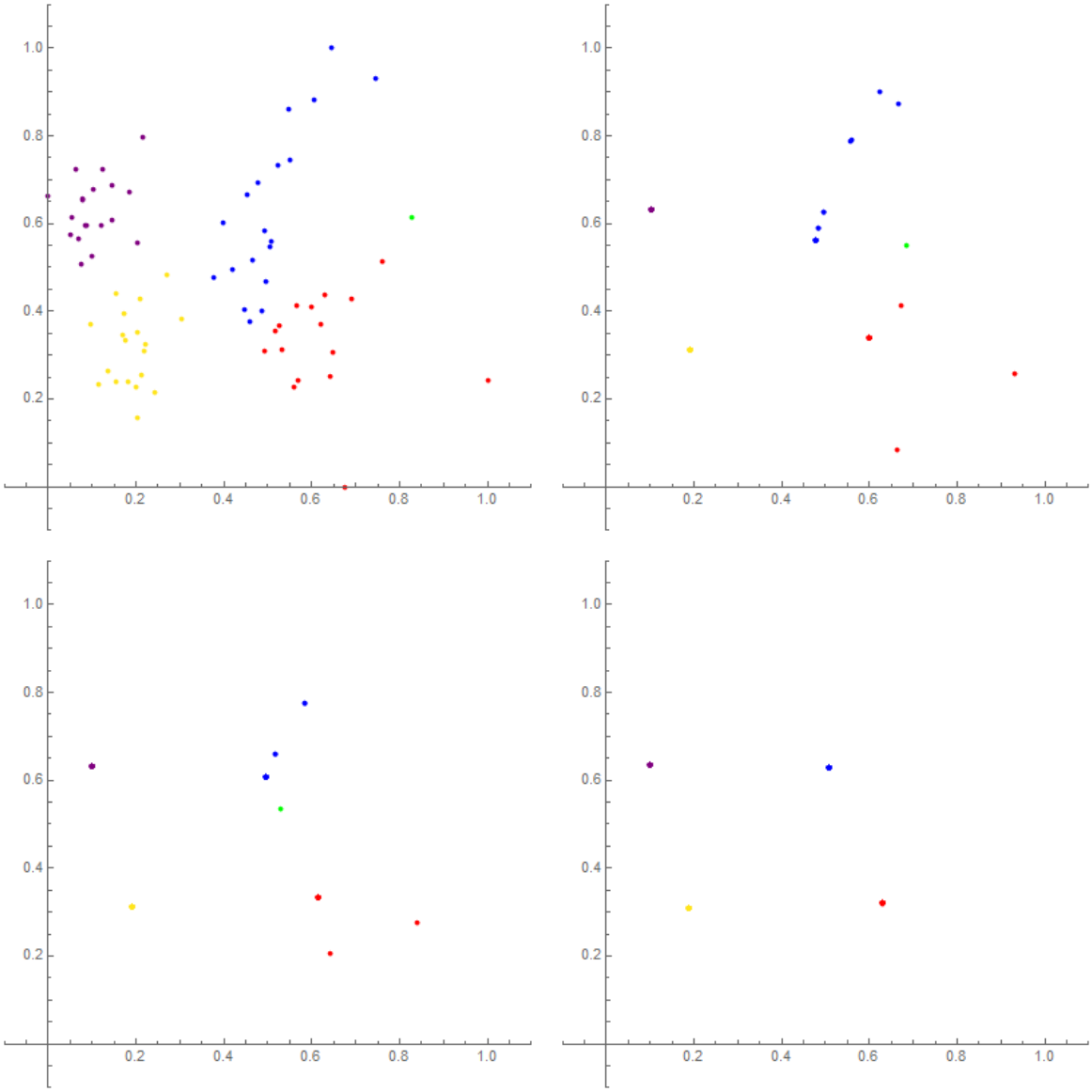
Zopakujeme rovnaký postup ako pri prvej verzii, teda spustíme učiacu fázu s rozpätím parametra difúzneho koeficienta $K = [100, 20000]$ s krokom $Kstep = 100$ a s rozpätím parametra δ-okolia $\delta = [0.001, 0.1]$ s krokom $\delta step = 0.001$. Pri zmene, ktorú sme urobili, keď sme iba odstránili zle zaradené body a zvyšné body nezmenili svoje miesto, budeme očakávať, že úspešnosť nového datasetu bude 100%.

Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset74A	73	1	0	98.65%

Tabuľka 4.9: Výsledky učiacej fázy na Datasete74A.

V prípade Datasetu74A sa nám podarilo nájsť tridsať optimálnych parametrov, pri ktorých sa správne zaradili takmer všetky body, až na jeden, ktorý sa zaradil zle.

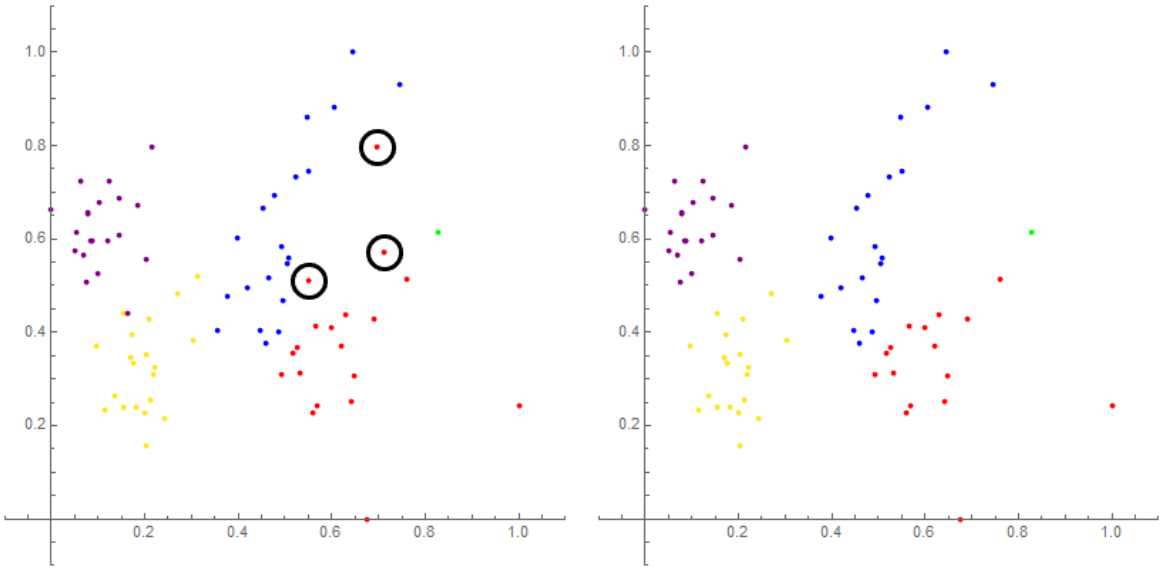
Vývoj zle zaradeného bodu je zvizualizovaný na obrázku (Obr. 4.13), kde si môžeme všimnúť, že na začiatku sa bod nachádza medzi červeným a modrým klastrom, aj keď má bližšie k červenému klastru, ku ktorému pôvodne patrí. Ako sledujeme vývoj, novo prichádzajúci bod sa pohybuje smerom do stredu, až kým ho nepritiahnu k sebe body modrého klastra.



Obr. 4.13: Klasifikácia Datasetu74A s jedným novo prichádzajúcim bodom (zelený bod), pôvodne z červeného klastra, v časových krokoch $n = 0, 10, 20, 33$. Použité optimálne parametre $K = 5100$ a $\delta = 0.001$, pri ktorých sa novo prichádzajúci bod zle zaradí.

Zaujímalo nás prečo nastala takáto situácia a čo ju spôsobilo. Zistili sme, že práve tých šesť zle zaradených bodov z Datasetu80, ktoré sme odstránili, zapríčinili, že sa tento jeden bod zle zaradí. Keď sa pozrieme na obrázok (Obr. 4.14 (naľavo)) vidíme, že

v Datasete80 novo prichádzajúci bod je obklopený červenými bodmi, a keďže aj novo prichádzajúci bod patrí červenému klastru predpokladáme, že by sa zaradil správne. Ibaže keď sa pozrime na Dataset74A (Obr. 4.14 (napravo)), v ktorom sme odstránili okolité body novo prichádzajúceho bodu, tento bod je už nie tak silno ovplyvnený bodmi červeného klastra a vo výsledku sa priradí do zlého klastra.



Obr. 4.14: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod) a vyznačenými tromi bodmi (červené krúžky), ktoré spôsobia, že sa po ich odstránení, novo prichádzajúci bod zle zaradí (naľavo). Dataset74A s jedným novo prichádzajúcim bodom (zelený bod), po odstránení troch spomínaných bodov (napravo).

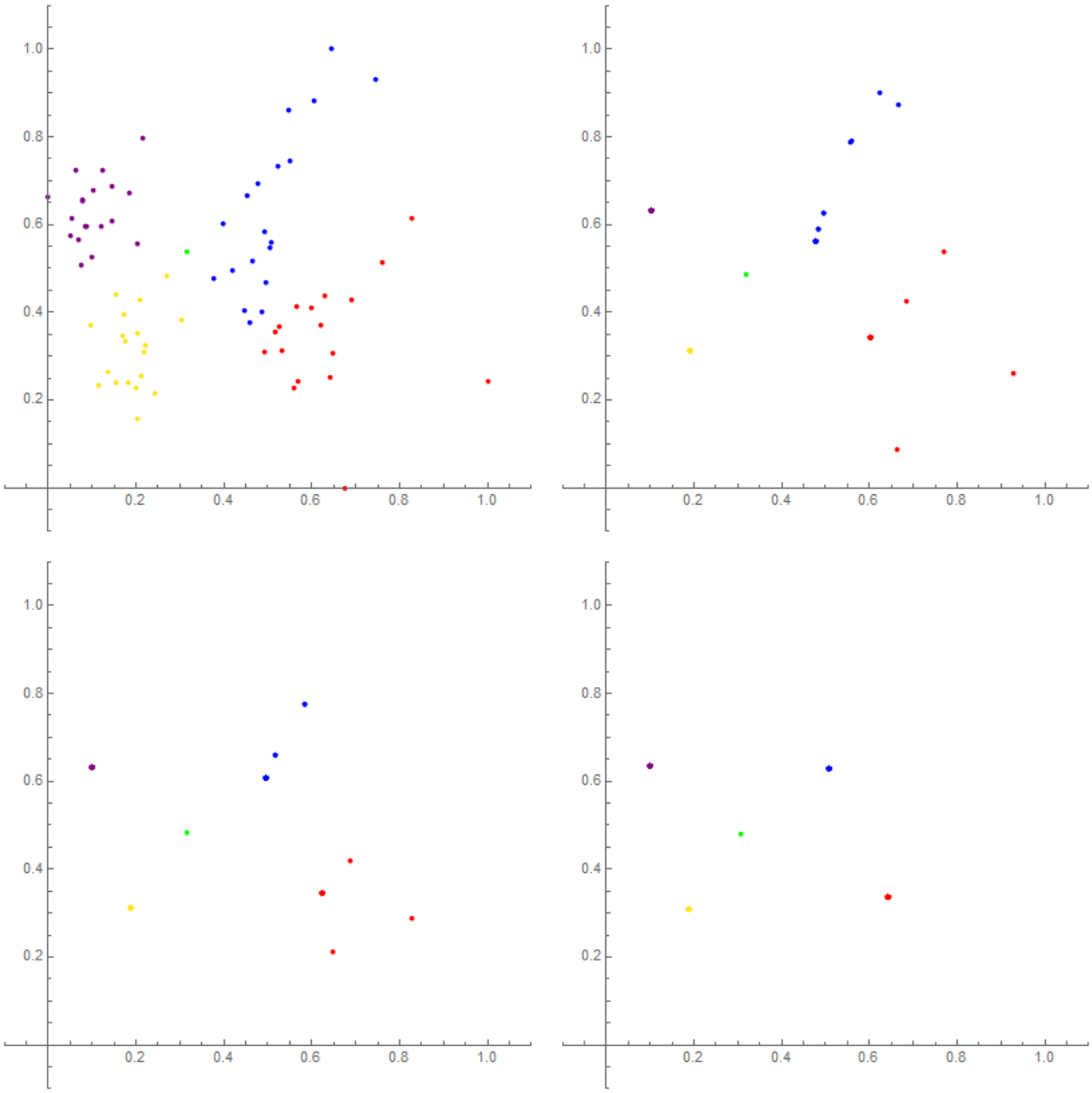
Vo validačnej časti budeme klasifikovať body z Datasetu12 a Datasetu44 (Tab. 4.7). Zvolíme jednu sadu optimálnych parametrov, konkrétne $K = 5100$ a $\delta = 0.001$, a zistíme, či Dataset74A bude vhodnejší na klasifikáciu.

Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset12	10	2	0	83.34%
Dataset44	32	10	2	72.73%

Tabuľka 4.10: Výsledky validačnej fázy na Datasete74A pri novo prichádzajúcich bodoch z Datasetu12 a Datasetu44.

Z tabuľky (Tab. 4.10) vidíme, že Dataset74A je úspešnejší, aj keď v oboch prí-

padoch iba o jednu hodnotu. V prípade novo prichádzajúcich bodov z Datasetu12 sa dva body zle zaradia, pretože sa nachádzajú na rozhraní dvoch klastrov. V prípade novo prichádzajúcich bodov z Datasetu44 väčšina zle zaradených bodov sa zle zaradila, kvôli rovnakým dôvodom a to, že boli za začiatku umiestnené medzi dvomi klastrami. Pri Datasete44 máme dva nezaradené body a jeden z nich si ukážeme (Obr. 4.15), pretože je zaujímavé, čo sa s novo prichádzajúcim bodom dialo.



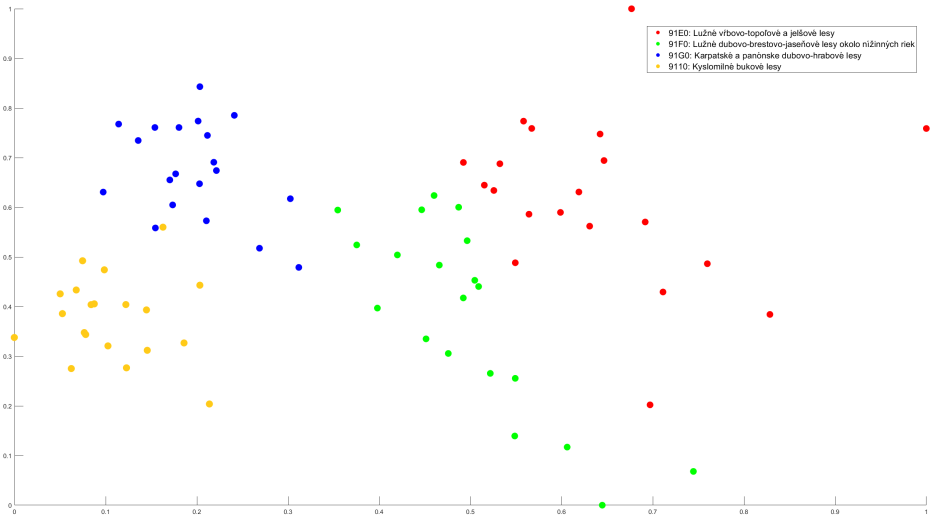
Obr. 4.15: Klasifikácia Datasetu74A s jedným novo prichádzajúcim bod (zelený bod) z Datasetu44 v časových krokoch $n = 0, 10, 20, 30$. Novo prichádzajúci bod pôvodne patril žltému klastru, no zostal nezaradený.

Môžeme si všimnúť, že na obrázku (Obr. 4.15) novo prichádzajúci bod sa nachádza v strede medzi tromi klastrami a ako sa body vyvíjajú novo prichádzajúci bod sa veľmi

nehýbe, až nakoniec zostane jednoznačne nezaradený. Je to spôsobené tým, že keďže je bod v strede, na neho pôsobia všetky body približne rovnakou silou a bod nevie kam sa má pohnúť. Potom nastane taká situácia, že novo prichádzajúci bod zostane sám, teda v jeho δ-okolí sa nenachádza žiaden iný bod, ktorý by ho pritiahol k jednému z klastrov a tento bod sa právom označí ako nezaradený.

4.1.3 Tretia verzia validácie

Pri tretej verzii si ukážeme ako sa budú správať novo prichádzajúce body pri klasifikácii s Datasetom80 v pôvodnej podobe (Obr. 4.16). Myšlienka bola taká, že v Datasete80 každý bod má presne priradený klaster a aj keď sa v učiacej fáze niektoré body zle zaradili, pri validačnej fáze sa určite zaradia správne, keďže už vopred poznajú klaster, ku ktorému patria. Body, ktoré sa pri učiacej fáze zle zaradili sme v tomto prípade ponechali v datasete z toho dôvodu, že ak by novo prichádzajúci bod bol vedľa jedného zo zle zaradených bodov, tento bod by ho ovplyvnil. Skúsili sme uvažovať tak, že body z Datasetu80 majú určite správny klaster a nezáleží nám na tom ako sa zaradili v učiacej fáze, ale budeme sledovať iba zaradenie novo prichádzajúcich bodov.



Obr. 4.16: Rozloženie bodov pôvodného Datasetu80.

Pripomeňme si ako Dataset80 vyzerá. Vytvorený je z pôvodných dát 124×72, kedy sme vybrali z každého typu biotopu dvadsať oblasti a vznikol nám dataset s rozmerom 80×72. Na začiatku tejto časti sme hovorili o učiacej fáze na Datasete80, kedy sme dosiahli úspešnosť 92.5%, teda správne sa zaradilo 74/80 bodov.

Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset80	74	6	0	92.5%

Tabuľka 4.11: Výsledky učiacej fáze na Datasete80.

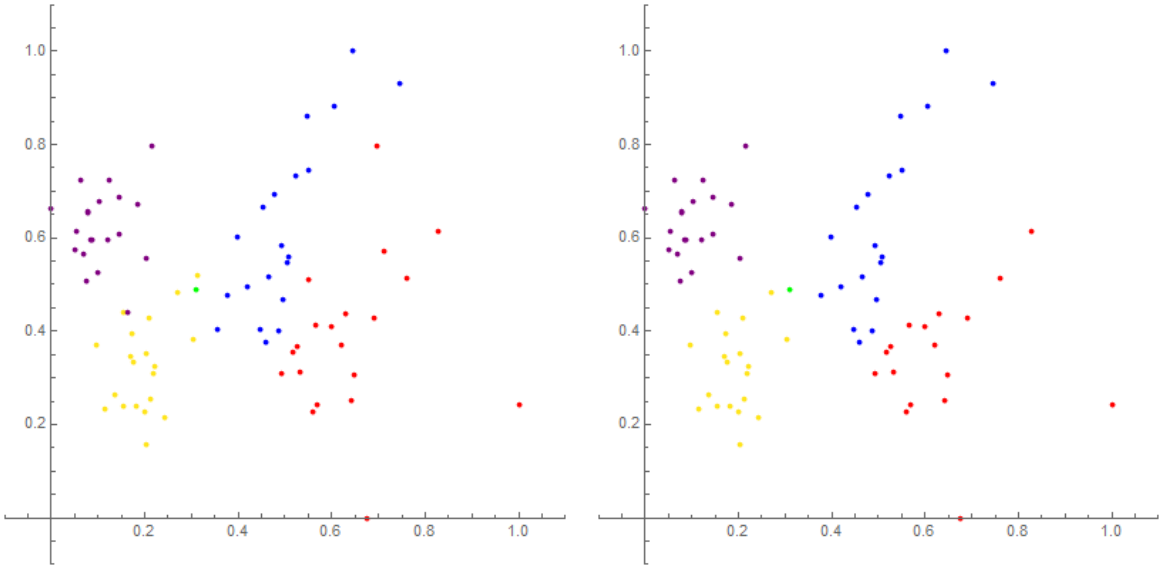
Výsledok 74/80 správne zaradených bodov sa nám poradilo dostať pre pätnásť sád parametrov, parametra difúzneho koeficienta K a parametra δ-okolia δ. Šesť zle zaradených bodov sa zle zaradilo, kvôli tomu, že sa nachádzali medzi dvomi klastrami alebo dokonca medzi tromi klastrami, teda ich črty neboli jednoznačne definované.

Validačnú fázu algoritmu na Datasete80 s novo prichádzajúcimi bodmi z Datasetu12 a Datasetu44 sme znovu o trošku vylepšili prístupom neodstrániť žiaden zle zaradený bod. Klasifikáciu sme spustili s parametrami $K = 5100$ a $\delta = 0.001$.

Názov datasetu	Počet správne zaradených	Počet nesprávne zaradených	Počet nezaradených	Percentuálny podiel úspešnosti
Dataset12	10	2	0	83.34%
Dataset44	33	11	0	75%

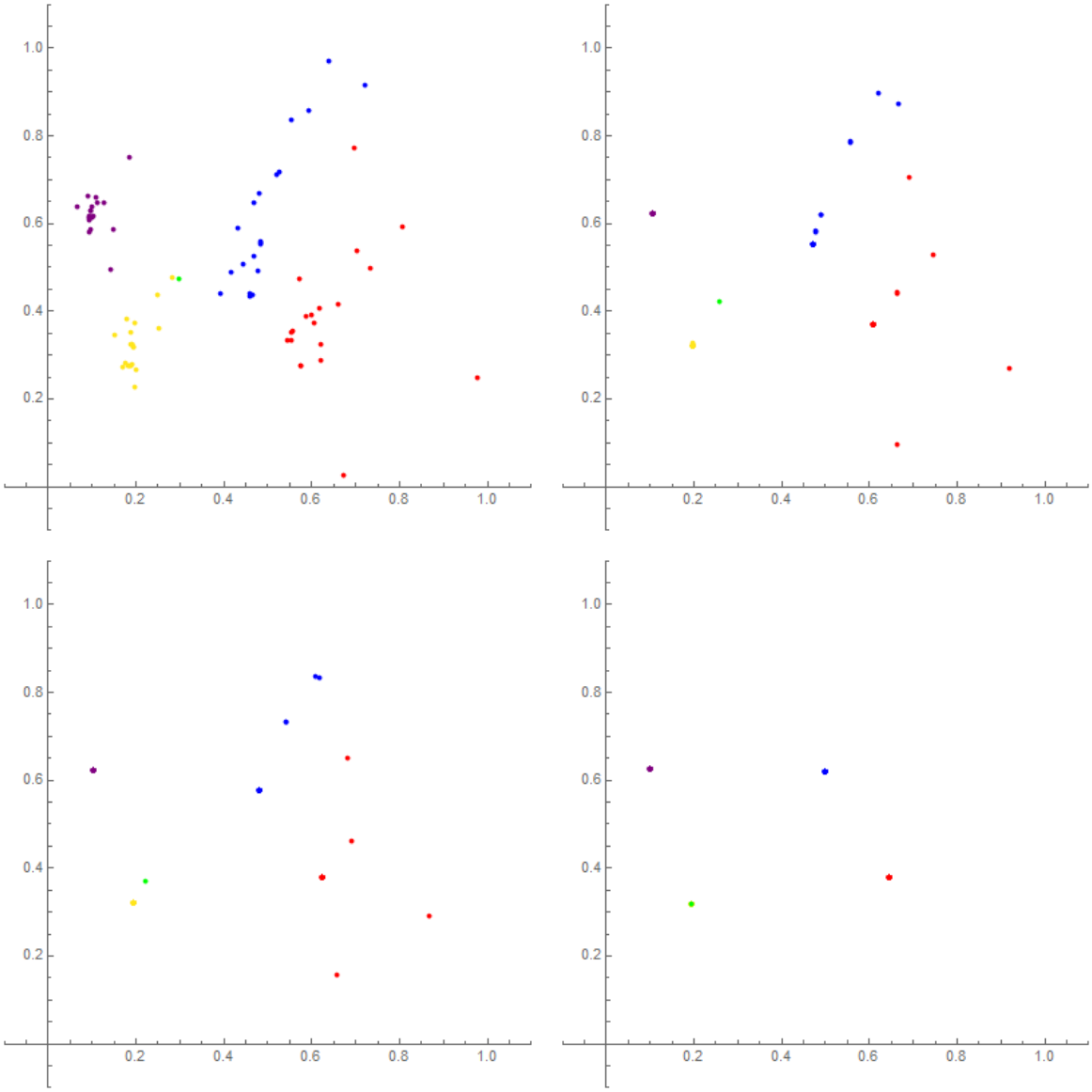
Tabuľka 4.12: Výsledky validačnej fázy na Datasete80 pri novo prichádzajúcich bodoch z Datasetu12 a Datasetu44.

Konkrétne pri novo prichádzajúcich bodoch z Datasetu12 sa nezmenilo nič, teda rovnako dva body sa zle zaradili a príčina je ich umiestnenie na rozhraní dvoch klastrov. V prípade novo prichádzajúcich bodov z Datasetu44 sa, na rozdiel od minulých verzií, nevyskytol nezaradený bod. Jeden bod, ktorý bol pri Datasete74A nezaradený sa pri Datasete80 dobre zaradil (Obr. 4.18). Je to kvôli tomu, že pri Datasete74A, ktorý vznikol po odstránení zle zaradených bodov z Datasetu80 v učiacej fáze, sa v δ-okolí novo prichádzajúceho bodu nachádzalo málo bodov (Obr. 4.17 (napravo)), ktoré mohli vplývať na jeho pohyb. Novo prichádzajúci bod nevedel kam má ísť, preto ostal nezaradený. V prípade Datasetu80, v δ-okolí novo prichádzajúceho bodu je viac bodov (Obr. 4.17 (naľavo)), ktoré ovplyvňujú jeho pohyb a preto sa v tomto prípade zaradí správne.



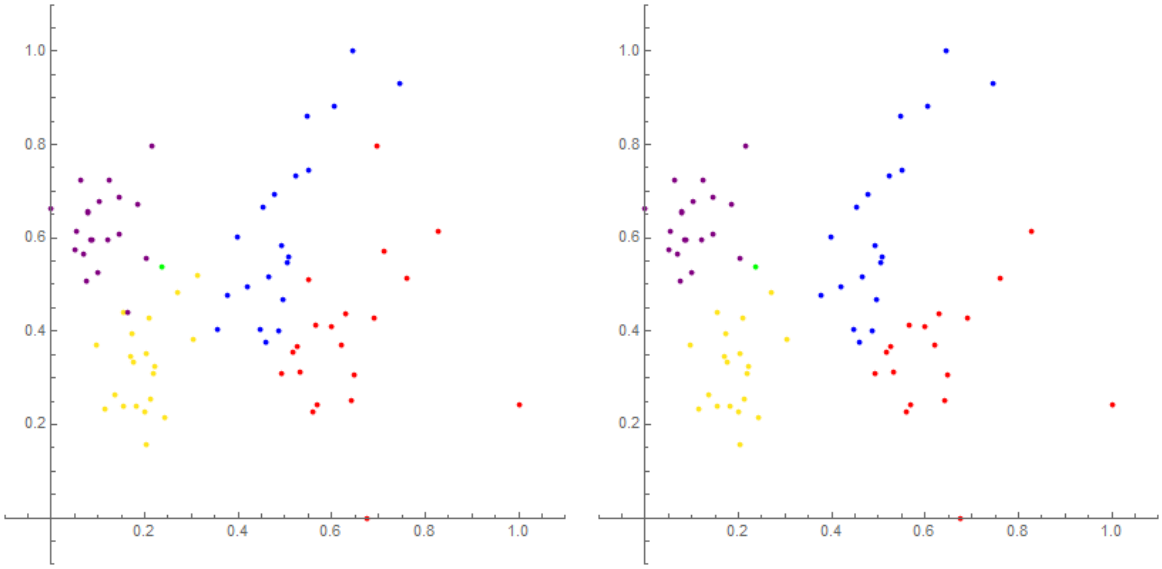
Obr. 4.17: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod) z Datasteu44 (naľavo) a Dataset74A s rovnakým novo prichádzajúcim bodom (zelený bod) z Datasetu44 (napravo). Oba prípady sú v $n = 0$.

Pri klasifikácii novo prichádzajúceho bodu na obrázku (Obr. 4.18), si môžeme všimnúť, že novo prichádzajúci bod bol pritiahnutý bodmi žltého klastra, preto sa rozhodol ísť smerom k tomuto klastru, a nakoniec sa k žltému klastru aj priradil. V tomto prípade je to správny výsledok, pretože novo prichádzajúci bod patril pôvodne žltému klastru a ako vidíme je podľa jeho čŕt umiestnený medzi dva body zo žltého klastra. Čiže celková situácia v Datasete80 s pridaným novo prichádzajúcim bodom z Datasetu44 poukazuje na fakt, že v tomto prípade sme dostali lepšie výsledky, keď sme zlé zaradené body v učiacej fáze z datasetu neodstraňovali.



Obr. 4.18: Klasifikácia Datasetu80 s jedným novo prichádzajúcim bod (zelený bod) z Datasetu44 v časových krokoch $n = 3, 10, 15, 31$. Novo prichádzajúci bod pôvodne patril žltému klastru a aj sa zaradil k žltému klastru.

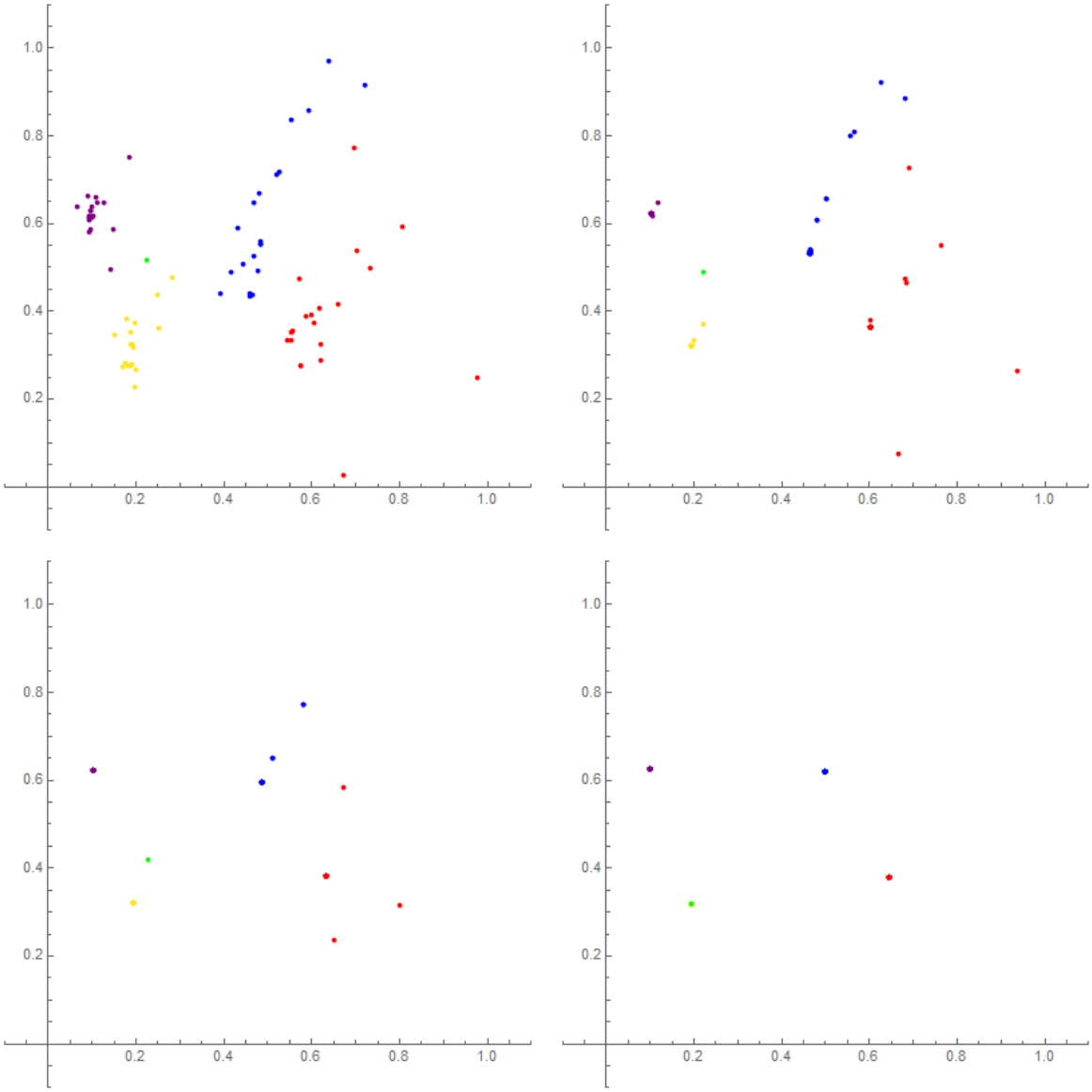
Pozrime sa aj na druhý novo prichádzajúci bod, ktorý sa pri Datasete74A nezaradil a v tomto prípade sa zaradil zle. Vidíme na obrázku (Obr. 4.19 (napravo)) rozloženie bodov v Datasete74A s vyznačeným novo prichádzajúcim bodom. Novo prichádzajúci bod sa nachádza na pomedzí, ale má tendenciu byť bližšie k fialovému klastru, ku ktorému aj skutočne patrí. Na prvý pohľad by sme asi povedali, že tento bod aj naozaj skončí v klastri fialových, ale musíme uvažovať, že novo prichádzajúci bod priťahujú aj body iných klastrov a preto po vývoji tento bod ostal nezaradený.



Obr. 4.19: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod) z Datasteu44 (naľavo) a Dataset74A s rovnakým novo prichádzajúcim bodom (zelený bod) z Datasetu44 (napravo). Oba prípady sú v $n = 0$.

Keď si všimneme rozloženie bodov v Datasete80 s vyznačeným novo prichádzajúcim bodom (Obr. 4.19 (naľavo)), na prvý pohľadom už zaradenie nie je také jasné. Vidíme, že novo prichádzajúci bod má okolo seba viac bodov zo žltého klastra a aj keď je bližšie k fialovému klastru, môže sa stať, že tento bod viac ovplyvnia body žltého klastra. Na obrázku (Obr. 4.20) môžeme sledovať vývoj klasifikácie spomínaného novo prichádzajúceho bodu. Deje sa presne opačný scenár ako v minulom prípade. Teraz body, ktoré sa zle zaradili a ktoré sme ponechali v Datasete80, ovplyvnia novo prichádzajúci bod. Fialový bod, ktorý bol na začiatku bližšie k novo prichádzajúcemu bodu, sa od bodu vzdiali a ďalšie okolité body pritiahnu novo prichádzajúci bod k sebe a on sa začne pohybovať smerom k žltému klastru. Nakoniec sa k žltému klastru zaradí, aj keď pôvodne patril fialovému klastru.

Teda toto je druhá strana úvahy, kedy ponechané zle zaradené body spôsobili to, že sa novo prichádzajúci bod zle zaradil. Preto bude potrebné v ďalšom výskume ešte hlbšie preskúmať všetky tri verzie tvorby datasetov.



Obr. 4.20: Klasifikácia Datasetu80 s jedným novo prichádzajúcim bod (zelený bod) z Datasetu44 v časových krokoch $n = 3, 8, 20, 31$. Novo prichádzajúci bod pôvodne patril fialovému klastru, ale zaradil sa zle.

4.1.4 Problémy s biotopom 9110

Skúsili sme porovnať výsledky klasifikácie hodnôt z Datasetu44 pri všetkých troch verziách validácie a zistenie, ku ktorému sme dospeli, je tiež jednou z tém na ďalšiu diskusiu. Dataset44 obsahuje tri chránené oblasti biotopu 91E0, štyri chránené oblasti biotopu 91F0, osemnásť chránených oblastí biotopu 91G0 a devätnásť chránených oblastí biotopu 9110. Pozreli sme sa na každú chránenú oblasť a na to ako prebiehala jej klasifikácia vo všetkých troch verziách (Tab. 4.13). Zisťujeme, že hodnoty z biotopu

91E0 a z biotopu 91F0 sa pri každej verzii vždy zaradili správne. Zaradenosť hodnôt z biotopu 91G0 sa mení pri rôznom prístupe tvorby datasetov pre klasifikáciu, pričom najlepšie výsledky boli dosiahnuté v tretej verzii, kedy boli do klasifikácie ponechané aj zle zaradené body z učiacej fázy. Pri biotope 9110, ktorý predstavuje Kyslomilné bukové lesy, práve naopak, početnosť správne zaradených bodov klesá. Po detailnejšej analýze sme zistili, že dôvod prečo sa pri tomto biotope tak málo hodnôt správne zaradí je ten, že vo väčšine prípadov sú body na rozmedzí medzi dvomi klastrami. To znamená, že črty chránených oblastí biotopu 9110 sú buď málo výrazné alebo veľmi podobné iným biotopom. Určite v ďalšom výskume bude potrebné konzultovať toto zistenie s odborníkmi z Botanického ústavu Slovenskej akadémie vied, aby sme zistili, čo môže spôsobovať nesprávne zaradenie oblastí biotopu 9110 a ich podobnosť s vysegmentovanými oblasťami z iných klastrov.

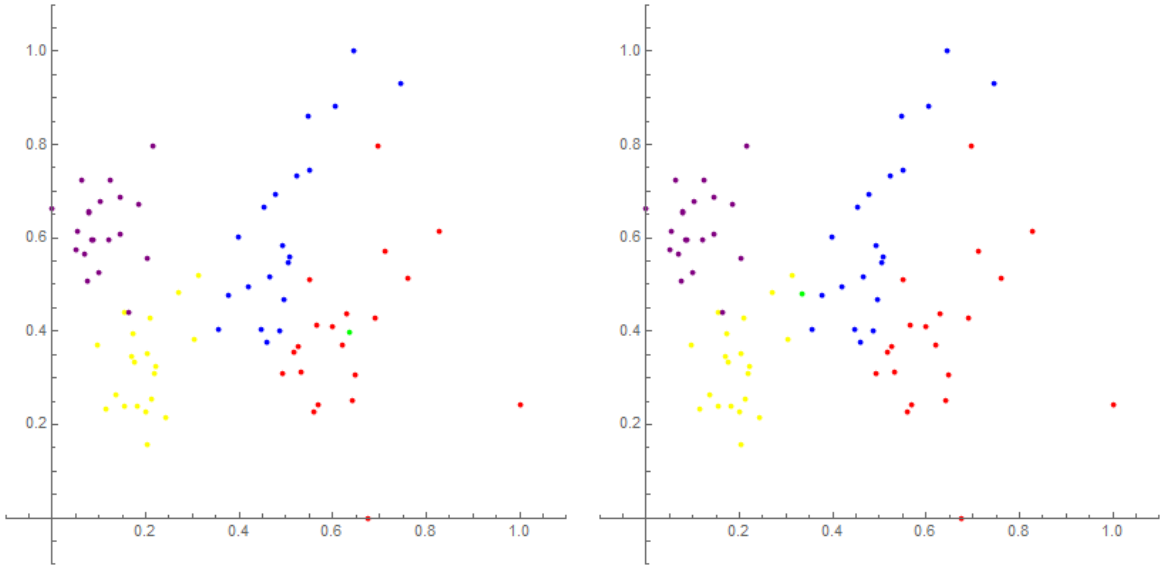
Úspešne zaradený:	Biotop 91E0	Biotop 91F0	Biotop 91G0	Biotop 9110
Prvá verzia (Dataset74)	3/3	4/4	14/18	10/19
Druhá verzia (Dataset74A)	3/3	4/4	15/18	10/19
Tretia verzia (Dataset80)	3/3	4/4	17/18	9/19

Tabuľka 4.13: Výsledky úspešne zaradených bodov vo validačnej fáze pri troch verziách datasetov s novo prichádzajúcimi bodmi z Datasetu44, rozpísané jednotlivo po biotopoch.

4.2 Relevantnosť úspešného zaradenia

Pri pozorovaniach trajektórií novo prichádzajúcich bodov počas klasifikácie sme si všimli, že v niektorých prípadoch je výsledné zaradenie do klastra jednoznačne určené už v prvých krokoch vývoja systému a inokedy je na výsledné zaradenie potrebných viac časových krokov. Pri prvom prípade sú polohy novo prichádzajúcich bodov v okolí ťažísk klastrov a pri druhom prípade sú novo prichádzajúce body na rozhraní viacerých klastrov alebo ďaleko od ich ťažísk (Obr. 4.21). Tieto pozorovania sme kvantifikovali

pomocou koeficientu relevantnosti, ktorý nám bude určovať relevantnosť výsledného zaradenia. Výsledný koeficient získame spriemerovaním dvoch čiastkových koeficientov $relevancy_1$ a $relevancy_2$.



Obr. 4.21: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod), pri ktorom je evidentné kam sa priradí (naľavo) a s jedným novo prichádzajúcim bodom (zelený bod), pri ktorom nevieme vopred povedať kam sa zaradí (napravo).

Koeficient relevantnosti $relevancy_1$ je definovaný vzdialenosťami novo prichádzajúceho bodu od ťažísk klastrov. Na začiatku klasifikácie novo prichádzajúceho bodu máme danú jeho polohu, následne po klasifikácii vieme, do ktorého klastra sa priradil a vieme si vypočítať ťažiská vzniknutých klastrov.

Označme si x ako novo prichádzajúci bod, c ako index klastra, ku ktorému sa priradil novo prichádzajúci bod. Ťažiská vzniknutých klastrov budú tvoriť vektor $centC$ o veľkosti N.

Najprv si vypočítame vzdialenosť novo prichádzajúceho bodu od ťažiska klastra, ku ktorému sa priradí,

$$dist(x) =\mid x - centC_c \mid,$$

kde $centC_c$ je ťažisko klastra, ku ktorému sa priradil novo prichádzajúci bod x.

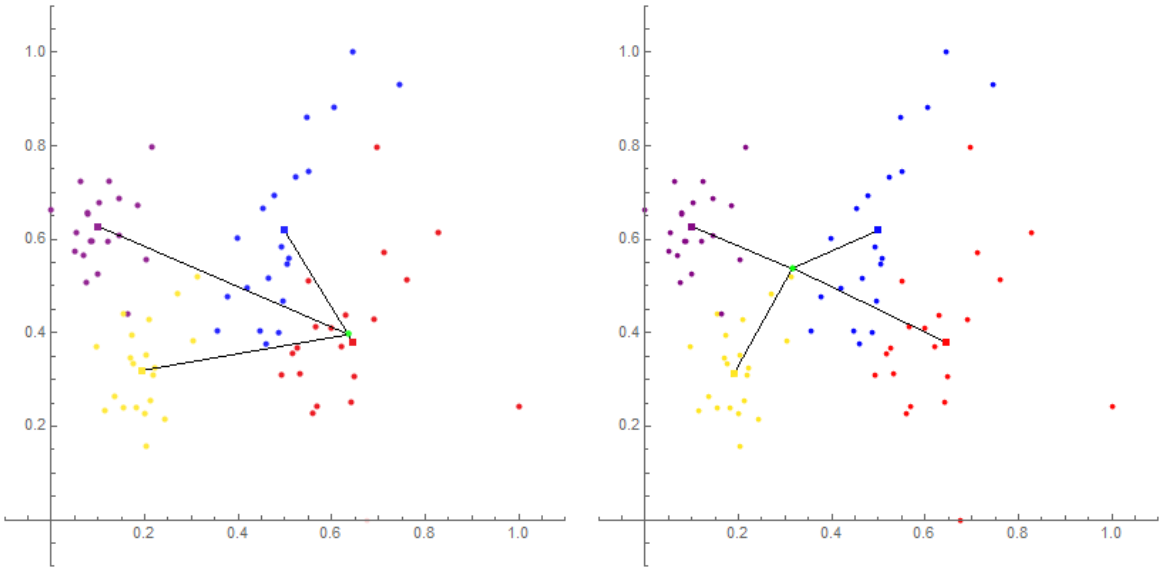
Následne spočítame priemernú vzdialenosť novo prichádzajúceho bodu od všetkých ostatných klastrov nasledovne

$$avgDist(x) = \frac{1}{N-1} \sum_{i,i\neq c}^{N} \mid x - centC_i(x) \mid,$$

kde v našom prípade $N = 4$.

Nakoniec spočítame hodnotu prvého čiastkového koeficientu relevantnosti, ako

$$relevancy_1(x) = 1 - \frac{dist(x)}{dist(x) + avgDist(x)}. \tag{4.1}$$



Obr. 4.22: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod), vyznačené ťažiská vzniknutých klastrov (farebné štvorčeky) a vyznačené vzdialenosti od ťažísk vzniknutých klastrov (čierne čiary). Novo prichádzajúci bod s čiastkovým koeficientom relevantnosti $relevancy_1 = 0.968$ (naľavo). Novo prichádzajúci bod s čiastkovým koeficientom relevantnosti $relevancy_1 = 0.518$ (napravo).

Koeficient relevantnosti $relevancy_1$ bude nadobúdať hodnoty z intervalu $[0, 1]$. V prípade ak $relevancy_1$ nadobudne hodnotu blízku 1 znamená to, že poloha novo prichádzajúceho bodu bola blízko ťažiska klastra, do ktorého sa bod zaradil (Obr. 4.22 (naľavo)). V tomto prípade je relevantnosť výsledného zaradenia vysoká. V opačnom prípade, ak je hodnota $relevancy_1$ blízka $1/2$ alebo dokonca menšia, znamená to, že vzdialenosť bodu od ťažiska klastra, do ktorého sa výsledne zaradil, je väčšia ako vzdialenosť od niektorého z ostatných ťažísk okolitých klastrov, čo znižuje výslednú relevantnosť zaradenia (Obr. 4.22 (napravo)).

Druhý čiastkový koeficient relevantnosti $relevancy_2$ skúma okolie novo prichádzajúceho bodu.

Označme si index novo prichádzajúceho bodu ako nc, klaster, ku ktorému sa priradí novo prichádzajúci bod, označme ako c, vstupné body ako X a vektor vzdialeností novo

prichádzajúceho bodu od ostatných bodov ako $dist$.

Pri výpočte tohto koeficientu relevantnosti sa najprv vypočíta vzdialenosť novo prichádzajúceho bodu od všetkých bodov

$$dist_i(nc) = \mid X_{nc} - X_i \mid, \qquad i = 1, \ldots, N_V.$$

Následne nájdeme iba tie body, ktoré sú od novo prichádzajúceho bodu vzdialené menej ako 0.1 a vyhodnotíme, ku ktorým klastrom patria tieto blízke body. V prípade ak blízky bod patrí do klastra c navýšime premennú $countC$ o jeden. V prípade ak blízky bod nepatrí do klastra c navýšime premennú $countA$ o jeden.

Ak sa premenná $countA = 0$, znamená to, že v okolí novo prichádzajúceho bodu sa nenachádza žiaden bod, ktorý by nebol z klastra c. V tomto prípade je čiastkový koeficient relevantnosti

$$relevancy_2 = 1, \tag{4.2}$$

keďže v okolí novo prichádzajúceho bodu sú iba také body, ktoré patria do klastra, do ktorého sa novo prichádzajúci bod zaradil.

Ak je premenná $countA \neq 0$, môžu nastať dve situácie. V prípade ak $countA > countC$, vtedy nastavíme čiastkový koeficient relevantnosti na

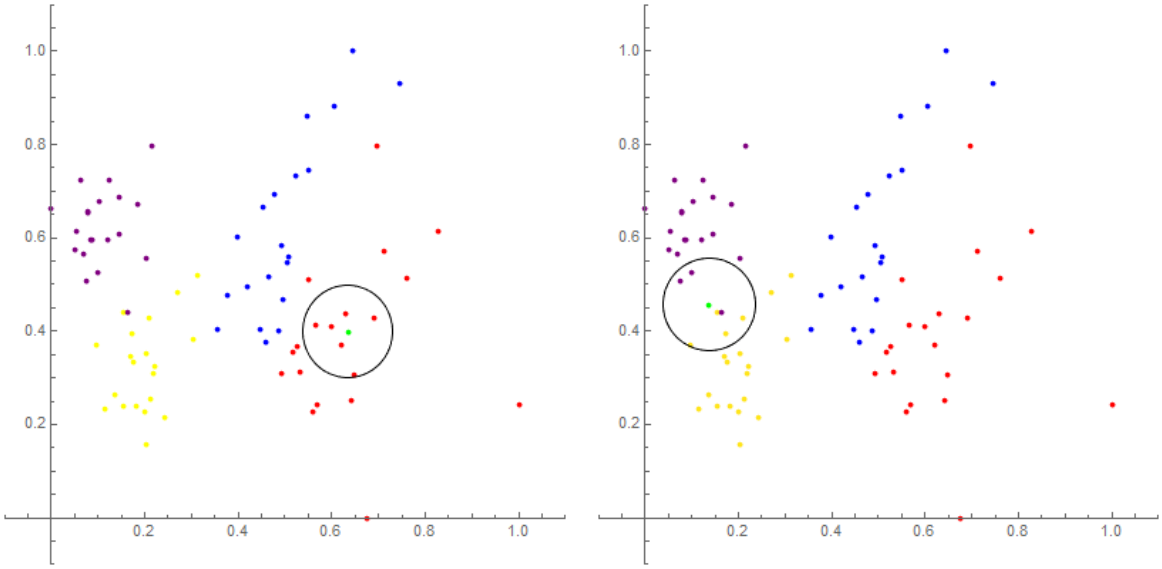
$$relevancy_2 = 0. \tag{4.3}$$

V tomto prípade je v okolí novo prichádzajúceho bodu viac bodov z klastra, do ktorých sa bod nezaradil, čo znižuje výslednú relevantnosť zaradenia.

V prípade ak $countA \leq countC$, koeficient relevantnosti sa vypočíta nasledovne

$$relevancy_2 = 1 - \frac{countA}{countC}. \tag{4.4}$$

Rovnako aj koeficient relevantnosti $relevancy_2$ nadobúda hodnoty z intervalu $[0, 1]$. Keď je koeficient $relevancy_2$ blízky 1, znamená to, že v okolí novo prichádzajúceho bodu sa nachádza väčší počet bodov, ktoré sú z klastra, do ktorého sa priradil aj novo prichádzajúci bod (Obr. 4.23 (naľavo)). V prípade ak koeficient $relevancy_2$ je blízky k 0, hovorí nám to, že v okolí novo prichádzajúceho bodu sa nachádza viac bodov z klastru, do ktorého sa novo prichádzajúci bod nepriradil (Obr. 4.23 (napravo)).



Obr. 4.23: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod), ktorý pôvodne patril fialovému klastru, ale zaradil sa k žltému, a s vyznačeným skúmaným okolím novo prichádzajúceho bodu (čierne krúžky). Novo prichádzajúci bod s čiastkovým koeficientom relevantnosti $relevancy_2 = 1$ (naľavo). Novo prichádzajúci bod s čiastkovým koeficientom relevantnosti $relevancy_2 = 0.25$ (napravo).

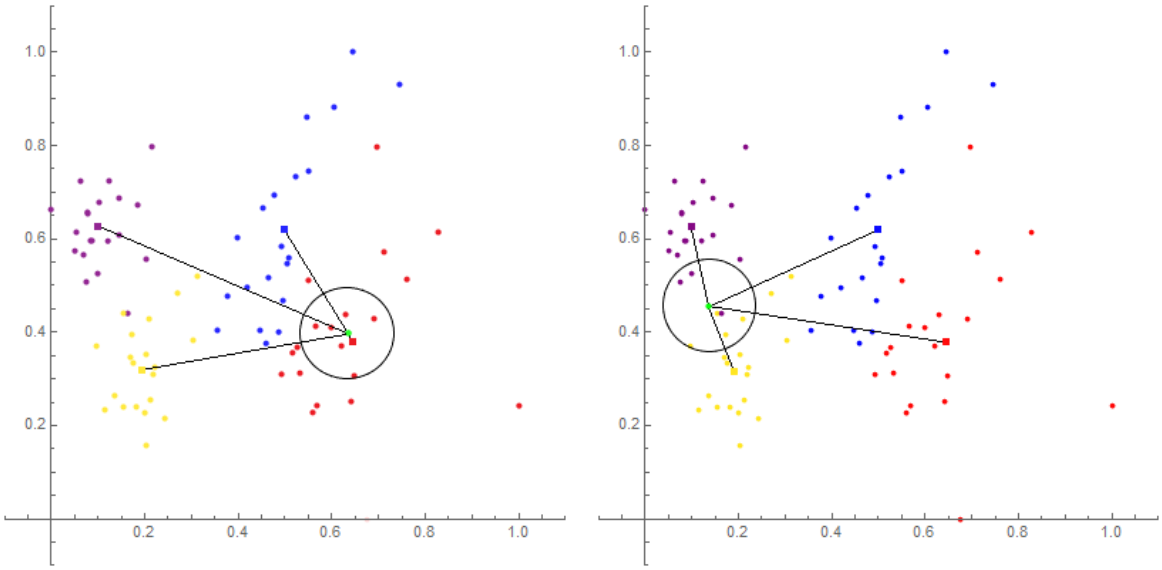
Celkový koeficient relevantnosti úspešne zaradeného novo prichádzajúceho bodu potom dostaneme ako priemernú hodnotu z oboch definovaných čiastkový koeficientov

$$relevancy = \frac{relevancy_1 + relevancy_2}{2}. \tag{4.5}$$

Ukážku fungovania koeficientu relevantnosti môžeme vidieť na obrázku (Obr. 4.24). Pri prvom prípade (Obr. 4.24 (naľavo)), novo prichádzajúci bod je pôvodne z červeného klastra a po klasifikácii sa zaradil do červeného klastra. Je vidieť, že novo prichádzajúci bod je takmer v ťažisku červeného klastra a jeho koeficient $relevancy_1 = 0.968$. Keď sa pozrieme na okolie novo prichádzajúceho bodu, vidíme, že v jeho okolí sa nachádzajú iba body červeného klastra, teda $relevancy_2 = 1$. Výsledný koeficient relevantnosti, že sa novo prichádzajúci bod zaradil správne do červeného klastra, je $relevancy = 0.984$.

V druhom prípade (Obr. 4.24 (napravo)), novo prichádzajúci bod je z fialového klastra, ale po klasifikácii sa zaradil k žltému klastru. Keď sa pozrieme na vzdialenosť novo prichádzajúceho bodu od ťažísk klastrov a vypočítame čiastkový koeficient relevantnosti $relevancy_1$ zistíme, že relevantnosť priradenia k žltému klastru je $relevancy_1 = 0.713$. Potom si vypočítame koeficient $relevancy_2 = 0.25$ a z neho zisťu-

jeme, že v okolí novo prichádzajúceho bodu je síce väčšina bodov z klastra, do ktorého sa priradil novo prichádzajúci bod, ale v okolí sa nachádzajú aj body z iných klastrov. Výsledný koeficient relevantnosti $relevancy = 0.481$ a hovorí nám o tom, že relevantnosť, že daný bod sa správne zaradil k žltému klastru je rovná 0.481. Poukazuje to správne na fakt, že novo prichádzajúci bod sa zaradil do zlého klastra, keďže vieme, že bod pôvodne patrí k fialovému klastru.



Obr. 4.24: Dataset80 s jedným novo prichádzajúcim bodom (zelený bod), s vyznačenými ťažiskami klastrov (farebné štvorčeky), s vyznačenými vzdialenosťami od ťažísk vzniknutých klastrov (čierne čiary) a s vyznačeným skúmaným okolím novo prichádzajúceho bodu (čierne krúžky). Novo prichádzajúci bod s koeficientom relevantnosti $relevancy = 0.984$ (naľavo) a koeficientom relevantnosti $relevancy = 0.481$.

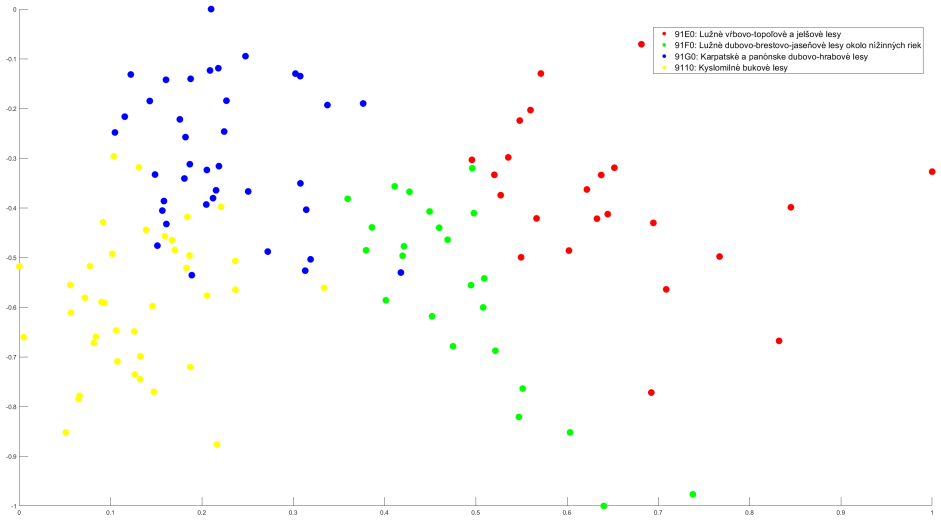
Kapitola 5

Záver

Úspešne sa nám podarilo navrhnúť a implementovať prirodzenú hlbokú sieť na báze dopredno-spätnej difúzie pre klasifikáciu environmentálnych dát. Numericky sme odvodili model difúzie na neorientovanom úplnom grafe pričom sme úspešne kombinovali doprednú a spätnú nelineárnu difúziu, aby sme dosiahli optimálne výsledky pri automatickom zaraďovaní nových pozorovaní do vopred určených klastrov, iba na základe ich čŕt. Pracovali sme s dátami zo systému Natura 2000, ktoré mapujú chránené oblasti na území Európskej únie. Odborní pracovníci z Botanického ústavu Slovenskej akadémie vied vysegmentovali chránené oblasti na území Západného Slovenska a nami vytvorený algoritmus kontrolovaného hlbokého učenia klasifikoval dáta podľa získaných charakteristík z družíc Sentinel-2. Pomocou softvéru NaturaSat, ktorý je vyvíjaný v spolupráci s Európskou vesmírnou agentúrou ESA, sme vypočítali dôležité črty z dát, ktoré predstavovali vstup do nášho algoritmu. Najprv sme vytvorený algoritmus učili, ako sa má rozhodovať pri zaraďovaní trénovacích dát tak, aby sme dosiahli čo najväčšiu úspešnosť správneho zaradenia. Trénovacie dáta sme vytvárali tak, aby početnosť v klastroch bola rovnomerná, a aby nám z pôvodnej vzorky zostali dáta na overenie funkčnosti algoritmu. Nasledovala validácia algoritmu, kedy sme mu dodali dáta, ktoré doposiaľ nevidel a analyzovali sme na koľko správne sa vedel rozhodnúť pri klasifikácii nových pozorovaní. Keďže sme zistili, že správne zaradenie dát nie vždy má rovnakú váhu, teda nie vždy je jednoznačné, že zaradenie je správne, do algoritmu sme implementovali ohodnocovanie relevantnosti správneho zaradenia dát. Pri vývoji algoritmu vzniklo viacero

otvorených otázok, ktoré budeme konzultovať a analyzovať s odborníkmi z botaniky pri našej ďalšej práci.

Pri ďalšom výskume sa budeme venovať aj samotnej klasifikácii na pôvodných dátach, ktoré pozostávajú zo stodvadsaťštyroch chránených oblastí biotopov (Obr. 5.1). Zatiaľ sme na tých dátach spustili iba učiacu fázu, kedy sme dosiahli úspešne zaradených 105 zo 124 bodov. Takto natrénovaná sieť má potenciál sa ďalej vyvíjať, jej úspešnosť sa môže zväčšovať postupným pridávaním nových hodnôt po úspešnej klasifikácii nových pozorovaní. Zväčšovali by sme učiacu vzorku dát tým, že by sme do nej pridávali nové pozorovania, ktoré budú mať vysokú relevantnosť správneho zaradenia, a tým by sme dosiahli aj väčšiu úspešnosť v učiacej a validačnej fáze.



Obr. 5.1: Rozloženie bodov pôvodne vysegmentovaných chránených oblastí biotopov.

Literatúra

[1] Chang B., Meng L., Haber E., Ruthotto L., Begert D., Holtman E., *Reversible Architectures for Arbitrarily Deep Residual Neural Networks*, Association for the Advancement of Artificial Intelligence, (2018).

[2] Copernicus, *Copernicus: Europe's eyes on Earth*, https://www.copernicus.eu/sk/o-programe-copernicus/program-copernicus-v-skratke, (2020).

[3] ESA, *Sentinel 2*, https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products, (2020).

[4] Friedman J., Tillich J-P., *Calculus on Graphs*, Archiv, (2008).

[5] GISGeography, *Sentinel 2 Bands and Combinations*, https://gisgeography.com/sentinel-2-bands-combinations/, (2019).

[6] Haber E., Ruthotto L., *Stable architectures for deep neural networks*, Inverse Problems 34(1), (2017).

[7] He K., Zhang X., Ren S., Sun J., *Deep residual learning for image recognition*, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 770–778, (2016).

[8] Knor M., *Teória grafov*, Slovenská technická univerzita v Bratislave, Bratislava, (2008).

[9] Mikula K. and Ramarosy N., *Semi-implicit finite volume scheme for solving nonlinear diffusion equations in image processing*, Numerische Mathematik, vol. 89, pp. 561–590, (2001).

[10] Mikula K., Urbán J., Kollár M., Ambroz M., Jarolímek I., Šibík J., Šibíková M., *An automated segmentation of Natura 2000 habitats from Sentinel-2 optical data*, Discrete & Continuous Dynamical Systems - S, (2019).

[11] Mikula K., Urbán J., Kollár M., Ambroz M., Jarolímek I., Šibík J., Šibíková M., *Semi-automatic segmentation of NATURA 2000 habitats in Sentinel-2 satellite images by evolving open curves*, Discrete & Continuous Dynamical Systems - S, (2018).

[12] Ng A., *What data scientists should know about deep-learning*, https://www.slideshare.net/ExtractConf/andrew-ng-chief-scientist-at-baidu, (2015).

[13] Perona P. and Malik J., *Scale-space and edge detection using anisotropic diffusion*, IEEE Transactionson Pattern Analysis and Machine Intelligence, vol. 12, pp. 629–639, (1990).

[14] Rencher A. C., *Methods of Multivariate Analysis*, Wiley-Interscience, New York, USA (2002).

[15] Štátna ochrana prírody SR, *Natura 2000*, http://www.sopsr.sk/natura/index1.php?p=3&lang=sk, (2020).

[16] Wikipedia, *Normalized difference vegetation index*, https://en.wikipedia.org/wiki/Normalized_difference_vegetation_index, (2020).

[17] Wikipedia, *Principal component analysis*, https://en.wikipedia.org/wiki/Principal_component_analysis, (2020).

[18] Wikipedia, *Rayleigh quotient*, https://en.wikipedia.org/wiki/Rayleigh_quotient, (2020).

[19] Wikipedia, *Sentinel-2*, https://en.wikipedia.org/wiki/Sentinel-2, (2020).