Ľubomír Dlhý, ID: 102897

KLASIFIKÁCIA

Ľubomír Dlhý, ID: 102897

ZADANIE

Máme 2D priestor, ktorý má rozmery X a Y, v intervaloch od −5000 do +5000. V tomto priestore sa môžu nachádzať body, pričom každý bod má určenú polohu pomocou súradníc X a Y. Každý bod má unikátne súradnice (t.j. nemalo by byť viacej bodov na presne tom istom mieste). Každý bod patrí do jednej zo 4 tried, pričom tieto triedy sú: red (R), green (G), blue (B) a purple (P). Na začiatku sa v priestore nachádza 5 bodov pre každú triedu (dokopy teda 20 bodov).

Vašou úlohou je naprogramovať klasifikátor pre nové body – v podobe funkcie classify(int X, int Y, int k), ktorá klasifikuje nový bod so súradnicami X a Y, pridá tento bod do nášho 2D priestoru a vráti triedu, ktorú pridelila pre tento bod. Na klasifikáciu použite k-NN algoritmus, pričom k môže byť 1, 3, 7 alebo 15.

RIEŠENIE

Na vyriešenie zadania bol použitý kNN algoritmus, teda algoritmus hľadaním k najbližších susedov. **Princíp fungovania algoritmu** je nasledovný:

1. Na začiatku prebieha trénovacia fáza – zoberú sa všetky vektory a priradia sa k nim prislúchajúce triedy (v našom prípade je to inicializácia 20 bodov – z každej farby po 5 bodov s presne danými súradnicami)
2. Po trénovacej fáze nastáva klasifikačná fáza. Body sa klasifikujú takou triedou, aká ja najčastejšia spomedzi *k* najbližších susedov. *K* je používateľom definová konštanta. Najbližší susedia sa určia pomocou euklidovej vzdialenosti medzi 2 bodmi – klasifikujúcim sa bodom a porovnávaným bodom

OPIS IMPLEMENTÁCIE

GENEROVANIE NOVÝCH BODOV

O generovanie nových bodov s jedinečnými súradnicami a rovnakým počtom bodov pre každú triedu sa stará funkcia

```
generateNewPoints(classifications, numOfEachClassification)
```

s parametrami *classifications* – pole tried a *numOfEachClassification* – počet bodov pre jednu triedu. Funkcia vygeneruje pre každú triedu potrebný počet bodov a vloží do poľa bodov, ktoré funkcia vráti. O jedinečnosť súradníc sa stará hash tabuľka (*pythonovský set()*), do ktorej sa pridávajú jednotlivé súradnice a pri generovaní sa overuje ich existencia v hash tabuľke. Potrebný počet bodov pre každú je implementovaný pomocou vnoreného cyklu, kde vonkajší cyklus sú 4 triedy a vnorený cyklus je potrebný počet bodov.

KLASIFIKÁCIA BODU

Klasifikácia bodu prebieha vo funkcii `classify(X,Y,k)`, ktorej parametre sú súradnice X a Y bodu a konštanta *k*. Vo funkcii sa postupne prepočítajú euklidove vzdialenosti všetkých už klasifikovaných bodov k nášmu bodu a *k* najbližších bodov sa ukladá. V implementácii sa najlepšie body ukladajú do dátovej

štruktúry dictionary (slovník), kde kľúčom sú súradnice bodu a hodnotou je vzdialenosť, súradnice bodu a jeho trieda. Ukladanie iba k najbližších bodov je spravené pomocou udržiavania si maximálnej hodnoty zatiaľ uložených vzdialeností. Pokiaľ novovypočítaná vzdialenosť bude menšia ako momentálne maximum z k najbližších susedov, tento najhorší sused sa vyberie a pridá sa nový bod. Po akejkoľvek zmene v slovníku k najbližších susedov treba nájsť opäť nové maximum a uložiť ho.

Po porovnaní všetkých už klasifikovaných bodov máme v slovníku k najbližších susedov. V ďalšom kroku je potrebné nájsť spomedzi najlepších bodov takú triedu, ktorá sa opakuje najviac (teda má najväčšiu incidenciu). V programe je to implementované tak, že triedam zvyšujem početnosť ako ich prechádzam v k najbližších susedoch. Funkcia vráti takú triedu, ktorá má najvyššiu početnosť.

VYKONANIE EXPERIMENTU

Vykonávanie experimentov – teda klasifikovanie všetkých vygenerovaných bodov pre rozdielnu konštantu k sa uskutočňuje vo funkcii main. Spočiatku sa prekopírujú všetky začiatočné body a k nim sa pridávajú klasifikované. Z tohto poľa sa počítajú aj vzdialenosti do už klasifikovaných bodov. Vygenerované body sa nachádzajú v samostatnom poli a postupne sa klasifikujú a pridávajú do spomínaného poľa už klasifikovaných bodov. Pre každý experiment sa ráta čas a úspešnosť. Pri klasifikácii rovnakou triedou akú mal daný bod sa zväčší premenná, ktorá značí počet úspešne klasifikovaných bodov. Úspešnosť celého experimentu sa potom vyráta ako podiel počtu úspešne klasifikovaných bodov a počet všetkých klasifikovaných bodov.

VYKRESĽOVANIE BODOV

Vykresľovanie bodov do grafu je realizované pomocou knižnice matplotlib. Po skončení jedného experimentu – teda klasifikovaní všetkých vygenerovaných bodov s konštantou *k* sa všetky body nachádzajú v jednom poli. Potom sú všetky body vykreslené s prislúchajúcou farbou triedy a graf je buď zobrazený v pop-up okne alebo je uložený, to závisí od nastavení používateľa.

UŽÍVATEĽSKÉ PROSTREDIE

```
Vyberte moznost:
(1) - spustenie s predvolenymi parametrami [#16000 bodov, 1 pokus(ov), Zobrazit grafy]
(2) - vlastne parametre
```

Po spustení programu sa zobrazí interaktívne menu, kde používateľ má jednu z dvoch možností – buď spustiť program s už prednastavenými parametrami, ktoré budú zobrazené na obrazovke alebo vybrať si vlastné parametre.

```
Zadajte pocet bodov pre jednu triedu:
10000
Zadajte pocet pokusov pre jeden experiment:
1
Zadajte 1 pre zobrazenie grafov a 0 pre ulozenie grafov na disk:
1
```

Po zvolení možnosti s vlastným nastavením hodnôt si program vyžiada potrebné údaje – počet generovaných bodov pre jednu triedu, počet pokusov pre jeden experiment a či chce užívateľ grafy zobraziť vo vyskakovacom okne alebo ich chce uložiť na disk.

Ľubomír Dlhý, ID: 102897

IMPLEMENTAČNÉ PROSTREDIE

Program bol implementovaný v jazyku Python vo verzii 3.8.5.
Boli použité knižnice: math, random, time (vstavané knižnice)
matplotlib – vykresľovanie grafov

TESTOVANIE

Program bol testovaný na **počte bodov 20 000, 40 000** a **100 000** pre hodnoty **k** = **1**, **3**, **7** a **15**.

Pre 20 000 a 40 000 bodov som urobil vždy 5 pokusov pre každý experiment, aby bolo meranie viac objektívne a na väčšej vzorke.

Pocet bodov	K	Uspesnost [%]	Time [sek]
20 000	1	73.95	39.21
20 000	1	71.46	43.92
20 000	1	72.32	44.93
20 000	1	71.95	49.95
20 000	1	74.30	43.71
20 000	1	72.80	44.344
20 000	3	74.44	42.85
20 000	3	76.45	41.06
20 000	3	75.44	42.96
20 000	3	73.81	45.74
20 000	3	75.24	50.85
200 000	3	75.08	44.692
20 000	7	74.19	46.28
20 000	7	73.84	44.59
20 000	7	75.22	41.57
20 000	7	74.30	45.56
20 000	7	76.84	52.4
20 000	7	74.88	46.08
20 000	15	59.29	47.99
20 000	15	56.42	46.37
20 000	15	53.84	46.86
20 000	15	46.88	50.14
20 000	15	57.56	62.15
20 000	15	54.80	50.70

Pocet bodov	K	Uspesnost [%]	Time [sek]
40 000	1	72.75	150.4
40 000	1	75.56	196.01
40 000	1	75.11	177.62
40 000	1	75.77	165.54
40 000	1	70.94	171.92
40 000	1	74.03	172.298
40 000	3	73.44	159.95
40 000	3	75.17	175.03
40 000	3	75.35	161.11
40 000	3	75.55	167.9
40 000	3	75.32	168.82
40 000	3	74.97	166.562
40 000	7	75.60	161.9
40 000	7	75.55	177.68
40 000	7	73.93	166.39
40 000	7	70.34	173.55
40 000	7	73.12	170.96
40 000	7	73.71	170.096
40 000	15	58.69	203.92
40 000	15	51.90	185.06
40 000	15	47.26	178.52
40 000	15	52.09	179.27
40 000	15	61.73	175.59
40 000	15	54.33	184.47

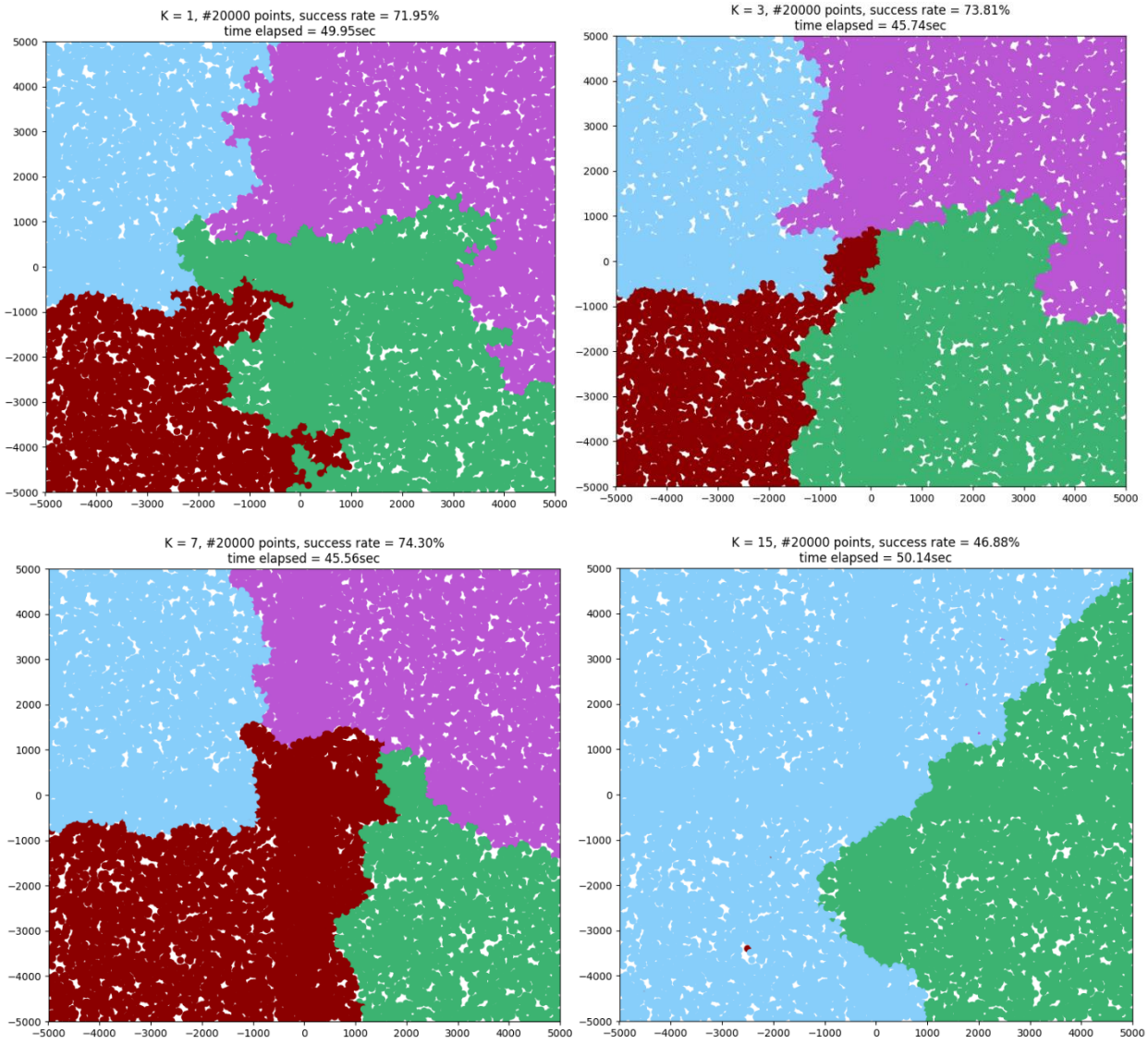
Pocet bodov	K	Uspesnost [%]	Time [sek]
100 000	1	73.35	941.88
100 000	3	75.97	1074.83
100 000	7	74.16	1049.64
100 000	15	58.11	1068.35

Legenda:
- Priemerná hodnota

Vo všetkých vykonaných experimentoch som zistil, že klasifikácia pre ***k* = 1, 3 a 7** má približne rovnakú úspešnosť – približne **74%** (odchýlka +- 2%). Pre ***k* = 15** bola priemerná úspešnosť **54%**, no pre 100 000 prvkov to bolo 58%. Spomedzi všetkých hodnôt *k* vychádzala najlepšie hodnota *k* = 3, potom *k* = 7, *k* = 1 a nakoniec s veľkým rozostupom *k* = 15.

Časovo vychádzalo **20 000** bodov v priemere na **46 sekúnd**, **40 000** bodov na **2 minúty 53 sekúnd** a **100 000** bodov na **17 minút a 14 sekúnd**.

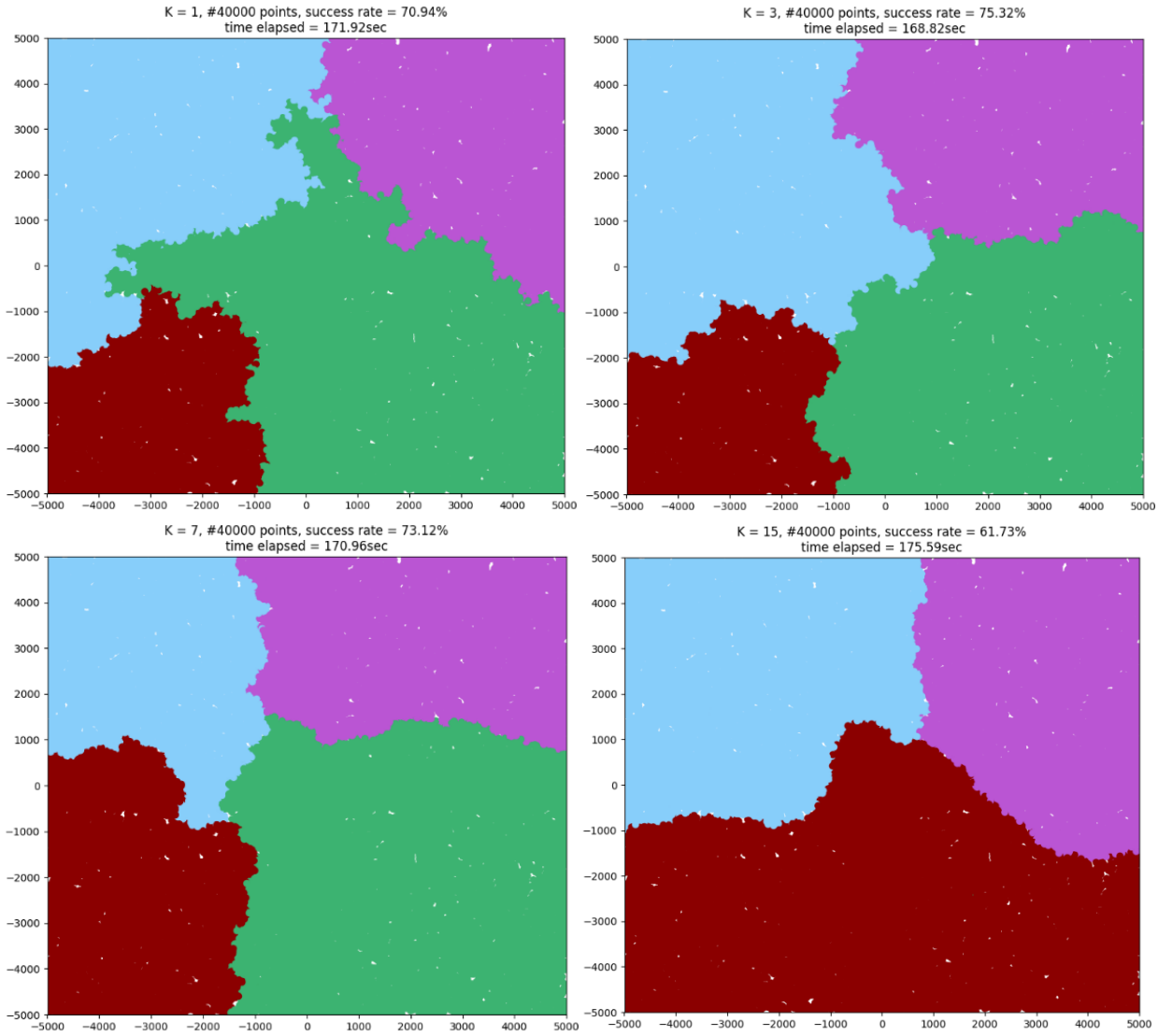
Ľubomír Dlhý, ID: 102897



Test pri **20 000 bodoch**.

Najlepšia bola hodnota k = 7 – úspešnosť 74,3%, potom nasledovalo k = 3, k=1 a nakoniec s veľkým rozdielom k = 15 s efektivitou iba 46,88%. Posledný graf má iba 2 triedy a to z toho dôvodu, že pri veľkej hodnote k sa hranice tried posúvajú a dokonca v našom prípade modrá predominovala a pohltila červenú farbu a spolu so zelenou tiež pohltili fialovú farbu.

Ľubomír Dlhý, ID: 102897



Test pri **40 000 bodoch**.

Najlepšia bola hodnota k = 3 – úspešnosť 75,32%, potom nasledovalo k = 7 (73,12%), k=1 (70,94%) a nakoniec s veľkým rozdielom k = 15 s efektivitou iba 61,73%. Z posledného grafu zmizla zelená trieda, ktorá bola pohltená červenou farbou. To vzniká z toho dôvodu, že pri veľkých hodnotách k sa posúvajú hranice a niektoré triedy môžu úplne zaniknúť. V prvom grafe (k=1) sú niektoré body chaoticky usporiadané a hranice medzi triedami nie sú priamočiare, ale majú rôzne výbežky. So zväčšujúcou sa hodnotou k hranice medzi triedami sú oveľa zreteľnejšie a priamočiare (nemajú toľko výbežkov).

Ľubomír Dlhý, ID: 102897

ZHODNOTENIE

Vytvorili sme klasifikátor s použitím kNN algoritmu, teda algoritmu hľadaním *k* najbližších susedov. Klasifikácia prebehla na vzorke 20 000, 40 000 a taktiež 100 000 bodov pri hodnotách k = 1, 3, 7 a 15. Vykonalo sa 5 pokusov pri každej hodnote k, aby boli výsledky o čosi objektívnejšie a vhodnejšie na porovnanie. Vo všetkých experimentoch bola úspešnosť klasifikácie pri hodnotách k = 1, 3 a 7 približne 74% ± 2%. Pri k = 15 bola efektivita oveľa nižšia – priemerne približne 55%. To znamená, že so zväčšujúcou sa konštantou k sa významne posúvajú hranice medzi jednotlivými triedami, dokonca niektoré triedy môžu úplne zaniknúť (sú pohltené inou triedou). Taktiež sme pozorovaním z grafov zistili, že so zväčšujúcou konštantou sa hranice stávajú viac priame a menej chaotické a roztrhnuté. Najlepšie vychádzala hodnota k = 3, potom k = 7, k = 1 a s veľkým rozdielom k = 15. Prvé 3 hodnoty v jednotlivých experimentoch mohli meniť víťazné poradie, keďže do výsledkov zasahovala náhodnosť generovaných prvkov.

Program bol testovaný na laptope s procesorom Ryzen 5 2500U @ 2.00 GHz (4 CPU 8 vlákien), časovo trvala klasifikácia v priemere 46 sekúnd pre 20 tis. bodov, 2 minúty 53 sekúnd pre 40 tis. bodov a 17 minút 14 sekúnd pre 100 tis. bodov.

Program klasifikuje 40 000 bodov do 3 minút (teda 4 experimenty trvajú ~12 minút), čo považujem za celkom vhodné a efektívne. Samozrejme, program sa dá ešte vylepšiť napr. rozdelením bodov do blokov. Experimentmi sme zistili že pri klasifikácii pomocou kNN je nesmierne dôležité vybrať vhodnú konštantu k tak, aby hodnota nebola ani príliš nízka (čo by ovplyvnilo graf tým, že hranice by boli trochu chaotické, nerovnomerné a nepriame) a súčasne aby hodnota konštanty nebola príliš vysoká, lebo by hranice tried sa mohli rozšíriť a pohltiť tak inú triedu.