

# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

## Fakulta informačních technologií



## Dokumentácia k projektu SFC

### Demonštrácia činnosti Fuzzy K-means algoritmu

# 1 Úvod

Tento dokument popisuje krátky úvod do problematiky, implementáciu a výsledky demonštrácie algoritmu Fuzzy K-means implementovaného v jazyku Python. Algoritmus bol implementovaný pre demonštráciu 2 rôznych úloh a to:

- zhľukovanie bodov v 2D priestore
- segmentácia čierneho-bielich medicínskych snímok.

## 2 Fuzzy K-means

Princíp Fuzzy K-means je veľmi podobný základnému algoritmu K-means, hlavný rozdiel spočíva v príslušnosti jednotlivých dát (bodov) k zhľukom. Pri klasickom K-means algoritme prebieha "tvrdé" zhľukovanie a teda každý bod prislúcha práve jednému zhľuku. Naproti tomu pri použití Fuzzy K-means, jednotlivé body prislúchajú každému zhľuku s nejakou váhou príslušnosti. Pretože v každej iterácii sa počítajú príslušnosti bodov ku všetkým zhľukom, je tento algoritmus o niečo pomalší ako klasický K-means, ktorý počíta len vzdialenosť od stredu zhľuku. Avšak Fuzzy K-means odstraňuje nedostatok závislosti na správne zvolených počiatkových stredoch jednotlivých zhľukov. Fuzzy K-means je vhodný na riešenie problémov kde máme zašumené dáta, porozumenia vzorov v dátach či hľadania závislostí/asociačných pravidiel v dátach. Taktiež sa používa na segmentáciu obrázkov napr. medicínskych snímok pretože takto dokážeme získať viac informácií z daných snímok.

Princíp algoritmu spočíva v minimalizácii objektívnej funkcie:

$$\operatorname{argmin}_c \sum_{i=1}^N \sum_{j=1}^C m_{ij}^q \|\vec{x}_i - \vec{c}_j\|^2, \quad (1)$$

kde  $m_{ij}^q$  je príslušnosť  $i$ -teho bodu k  $j$ -temu zhľuku a  $q \in R, m \geq 1$  predstavuje fuzzy koeficient. Ak sa  $q = 1$  algoritmus sa správa ako klasický K-means.

Poloha stredov zhľukov sa upravuje podľa rovnice:

$$\vec{c}_j = \frac{\sum_{i=1}^N m_{ij}^q \cdot \vec{x}_i}{\sum_{i=1}^N m_{ij}^q} \quad (2)$$

Úprava príslušnosti k jednotlivým bodom je počítaná nasledovne:

$$m_{ij} = \frac{1}{\sum_{k=1}^C \left( \frac{\|\vec{x}_i - \vec{c}_j\|}{\|\vec{x}_i - \vec{c}_k\|} \right)^{\frac{2}{q-1}}} \quad (3)$$

Algoritmus potom pracuje nasledovne:

1. Náhodne sa inicializujú hodnoty príslušnosti  $m_{ij}$  a fuzzy koeficient  $q$ .
2. Spočítajú sa hodnoty stredov jednotlivých zhľukov podľa rovnice č.2.
3. Spočítajú sa nové hodnoty príslušnosti k zhľukom pre všetky body podľa rovnice č.3.
4. Ak sú splnené ukončovacie podmienky skonči, inak pokračuj bodom 2.

### 3 Implementácia a demonštrácia činnosti

Projekt bol implementovaný v jazyku Python s použitím knižníc na počítanie s maticami (numpy), vykresľovanie dát (matplotlib) a PyQt5 na vytvorenie aplikácie. Algoritmus Fuzzy K-means je implementovaný ako jedna trieda **FuzzyCMeans**, pomocné funkcie na vykreslenie výsledkov sú implementované v súbore **utils.py** a aplikácia je implementovaná v **app.py**.

#### Trieda FuzzyCMeans

Všetky potrebné dáta, koeficienty a funkcie implementujúce algoritmus sú zabalené v jednej triede. Pri vytváraní objektu je nutné špecifikovať hyper-parametre ako napr. počet epoch, fuzzy koeficient, cestu k dátam alebo úlohu ktorú bude algoritmus vykonávať. Na výber je zhlukovanie bodov v 2D priestore (argument task = "**points**") alebo segmentácie obrázku (argument task = "**img**").

#### FuzzyCMeans.\_\_init\_\_memberships

Na začiatku náhodne inicializuje príslušnosť dát k jednotlivým zhlukom tak, že každý bod má náhodne hodnotu príslušnosti k nejakému zhluku 1 a k ostatným 0 keďže na počiatočných hodnotách nezáleží.

#### Úprava stredov zhlukov

Vypočítanie nových stredov prebieha zavolaním funkcie **update\_centroids\_2D()** pre zhlukovanie bodov v 2D priestore a **update\_centroids\_img()** pre segmentáciu obrázka.

#### Úprava príslušnosti bodov

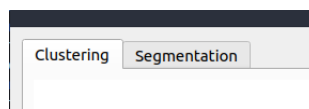
Nová príslušnosť je spočítaná vo funkciách **update\_membership\_2D()** pre zhlukovanie bodov v 2D priestore na základe ich euklidovskej vzdialenosti a **update\_membership\_img()** pre segmentáciu obrázka na základe hodnoty odtieňu šedi.

#### Beh algoritmu

Algoritmus sa spustí pomocou funkcií **run\_clustering\_2D()** alebo **run\_clustering\_img()** a beží po zadaný počet epoch. Je možné spraviť aj jeden krok algoritmu zavolaním funkcie **one\_step\_2D()** alebo **one\_step\_img()**.

#### Aplikácia na demonštráciu algoritmu

Aplikácia je implementovaná ako jedna trieda **Window** a je rozdelená na dve časti. V prvej je demonštrovaný priebeh zhlukovanie bodov v 2D priestore a v druhej je demonštrovaná segmentácia obrázka.



## Zhlukovanie bodov

Na obrázku 1 je predstavené zhlukovanie bodov s konfiguráciou 9 zhlukov a fuzzy koeficient je 2. Môžeme vidieť že už po 30 epochách celkom pekne algoritmus zkonvergoval nezávisle na pozícii počiatočných stredov zhlukov. Body su vykreslené do daného zhliku na základe najväčšej hodnoty príslušnosti a veľkosť vykresleného bodu závisí na sile príslušnosti do tohto zhliku (čím väčší bod, tým silnejšia príslušnosť). čím väčší je fuzzy koeficient  $q$  tým konverguje pomalšie pretože váhy príslušnosti sa menia pomalšie. Ako môžeme vidieť na obrázku 2 pre  $q = 10$  po rovnakom počte krokov nekonvergoval tento algoritmus tak rýchlo ako pre  $q = 2$ .

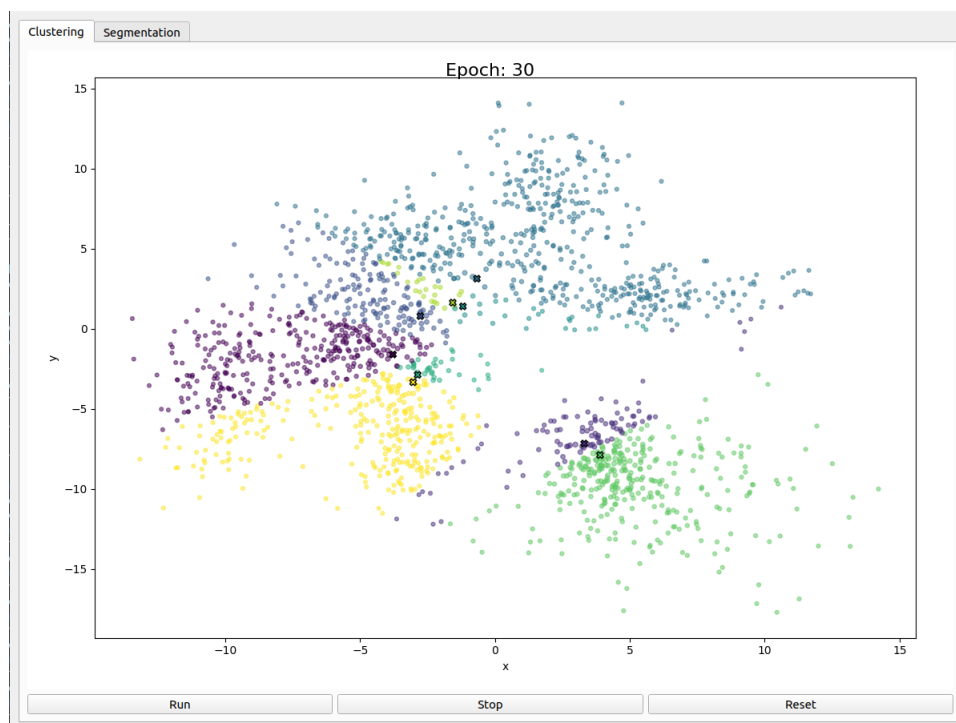


Obr. 1: Vytvorené zhluky po 30 epochách.

Tlačítkom **Run** pustíme algoritmus, **Stop** pozastaví aktuálny priebeh, tlačidlo **reset** resetuje hodnoty príslušnosti a stredov zhlukov do počiatočnej hodnoty.

## Pridanie nového bodu

Kliknutím do vnútra grafu je možné pridať nový bod, ktorý bude klasifikovaný v ďalšom kroku.



Obr. 2: Vytvorené zhľuky po 30 epochách s fuzzy koeficientom  $q=10$ .

## Segmentácia obrázka

Na obrázku 3 je predstavený algoritmus Fuzzy K-means pre segmentáciu snímkov CT pľúc. Kde jednotlivé segmenty, ktoré by potenciálne mohli odhaliť ďalšie informácie o stave pacienta sú oddelené rôznymi farbami. Každá farba predstavuje odlišný zhľuk algoritmu. Konfigurácia Fuzzy K-means pre tento snímok je: fuzzy koeficient 2, počet zhľukov je 11. Tlačítkom **Run** pustíme algoritmus, **Stop** pozastaví aktuálny priebeh, tlačidlo **Grey/Segmented** prepína medzi pôvodným čierno-bielym obrázkom a segmentovaným obrázkom a tlačidlo **reset** resetuje hodnoty príslušnosti a stredov zhľukov do počiatočnej hodnoty.

## 4 Spustenie projektu

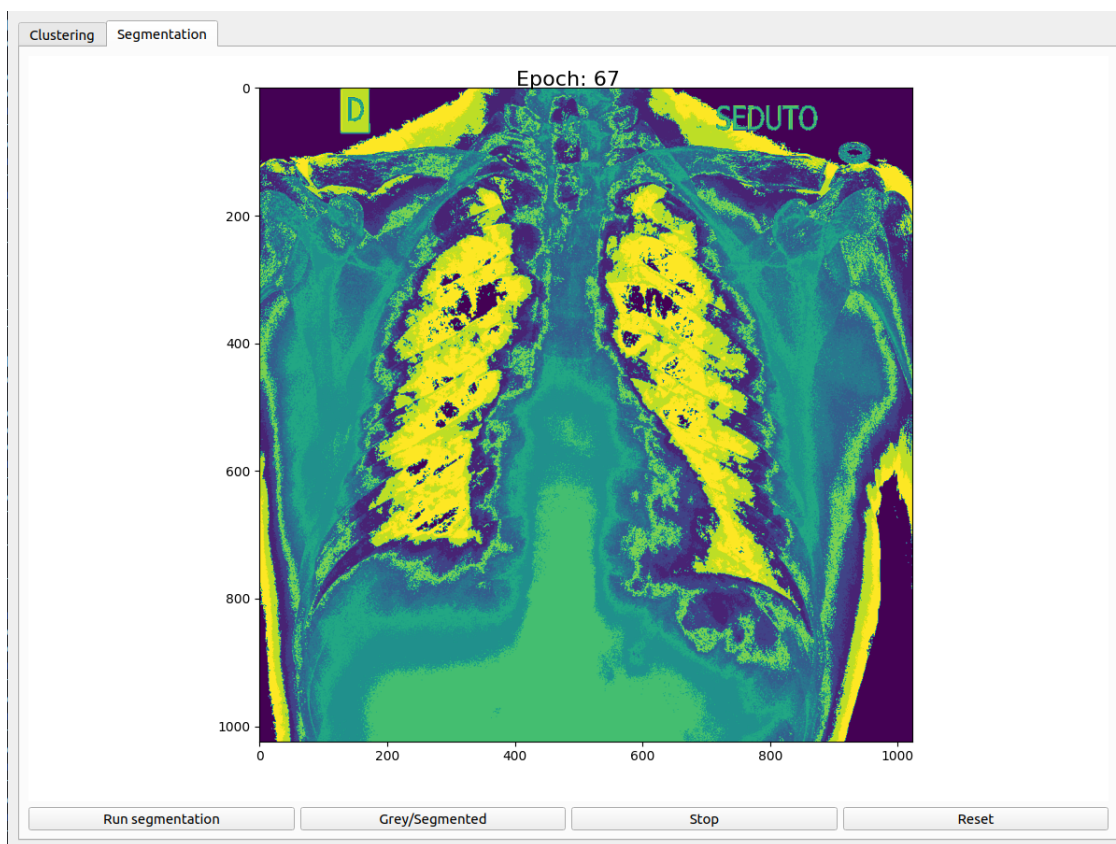
Projekt pozostáva z python skriptov, ktoré sa pre jednoduchosť spúšťajú pomocou **makefile**. Pred spustením je potrebné nainštalovať python prostredie (popis inštalácie nižšie).

### Inštalácia knižníc

Knižnice potrebné pre spustenie projektu sú uložené v **requirements.txt**. Predpokladá sa prítomnosť python3, venv a pip na referenčnom stroji. Inštalácia prostredia a knižníc sa spustí pomocou príkazu **make install**. Manuálne spustenie prostredia je možné pomocou príkazu **source path.sh**, avšak to nie je pre spustenie projektu potrebné (viz. nižšie).

### Spustenie skriptov a aplikácie

Pre spustenie skriptov a aplikácie sú predpripravené recepty v Makefile aj s príslušnými konfiguráciami, ktoré je možné zmeniť podľa potreby experimentov. Príkazy pre spustenie projektu:



Obr. 3: Segmentácia CT pľúc po prekonaní COVID-19

- `make points_exp1` spustí Fuzzy K-means algoritmus pre zhlukovanie bodov v 2D priestore, obrázok počiatočných bodov a výsledných zhlukov sú uložené do zložky skriptu (**src/**)
- `make covid` spustí Fuzzy K-means algoritmus pre segmentáciu CT snímku pľúc po prekonaní covid-19, výsledok segmentácie je uložený do zložky skriptu (**src/**)
- `make normal` spustí Fuzzy K-means algoritmus pre segmentáciu CT snímku zdravých pľúc, výsledok segmentácie je uložený do zložky skriptu (**src/**)
- `make pneum` spustí Fuzzy K-means algoritmus pre segmentáciu CT snímku pľúc po prekonaní ochorenia, výsledok segmentácie je uložený do zložky skriptu (**src/**)
- `make app` spustí interaktívnu aplikáciu pre demonštráciu činnosti Fuzzy K-means algoritmu

## Dáta

Dáta pre demonštráciu algoritmu sú uložené v **data/img** a **data/points**.

## 5 Záver

Podarilo sa mi implementovať základný Fuzzy K-means algoritmus pre dve rôzne úlohy a to zhlukovanie bodov a segmentácia snímkov. Spolu s jednoduchou aplikáciou na interaktívnu demonštráciu činnosti tohto algoritmu.