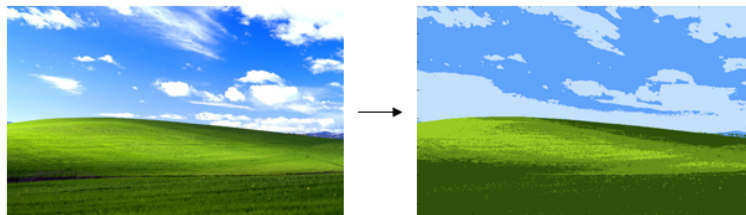

Úvod do Umelej Inteligencie

Cvičenie 11 - k -means

November 30, 2022

9. k -MEANS - POSTERIZÁCIA OBRÁZKU

Pomocou zhlukovacej metódy k -means budeme robiť posterizáciu obrázku, teda prevod do niekoľkých vybraných farieb.

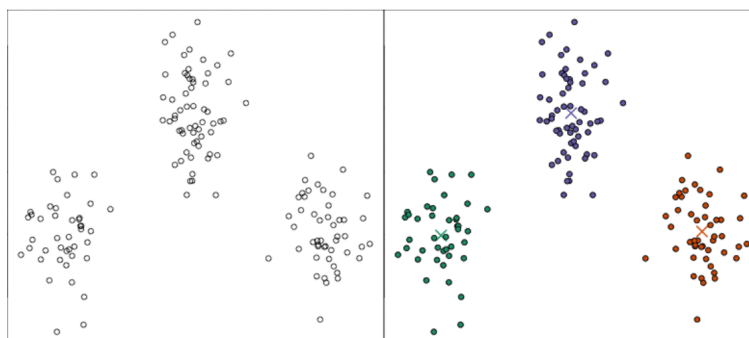


Dáta:

Vstupom pre náš program bude obrázok, t.j. množina pixelov, z ktorých každý má vlastnú farbu (RGB). Z týchto farieb pomocou k -means algoritmu vypočítame niekoľko (napr. 5) "hlavných" farieb. Obrázok potom upravíme tak, že bude obsahovať iba týchto 5 "hlavných" farieb (t.j. posterizácia). Vašou úlohou bude naprogramovať iba prostrednú časť tohto procesu, t.j. hľadanie hlavných farieb.

k -means zhlukovanie:

Princíp k -means spočíva v nájdení k centier, ktoré reprezentujú zhluky vo vstupných dátach. Príklad: na začiatku máme dvojrozmerné dáta, o ktorých nič viac nevieme (vľavo). k -means nájde tri centrá (krížiky na obrázku vpravo), ktoré reprezentujú zhluky v pôvodných dátach.



V našom prípade budú dáta trojrozmerné - R,G,B zložky farby. Výsledné centrá budú teda tiež trojrozmerné farby, a budú to spomínané "hlavné" farby, do ktorých obrázkov následne prevedieme. Usporiadanie pixelov (ich [x,y] súradnice) ignorujeme.

Algoritmus:

Existuje viacero alternatív k -means algoritmu, ktoré dávajú prakticky rovnaké výsledky.

1) Stochastická metóda: budeme sa vždy pozerat' iba na jeden bod (farbu pixelu), a opakovať nasledovnú úpravu centier pre každý bod počas niekoľkých iterácií:

- vezmi jeden bod x (farbu pixela)
- nájdi k nemu najbližšie centrum c^*
$$c^* = \operatorname{argmin}_{i=1..k} (||c_i - x||)$$
- "pritiahni" centrum c^* k bodu x
$$c^* := c^* + \alpha(x - c^*)$$

Podobne ako pri učení neurónových sietí, aj tu prechádzame všetky body x v náhodnom poradí, a to celé opakujeme niekoľko iterácií.

2) Batch metóda: pracujeme vždy so všetkými bodmi a centrá posúvame priamo do stredu zhhlukov:

- rozdeľ všetky body x_1, \dots, x_n do k zhhlukov C_1, \dots, C_k podľa toho, ktoré z centier c_i je k nim najbližšie
$$C_i = \{x | i = \operatorname{argmin}_{j=1..k} (||c_j - x||)\}$$
- nastav pozície centier c_1, \dots, c_k do stredov zhhlukov C_1, \dots, C_k
$$c_i = \operatorname{mean}(x \in C_i)$$

Pseudokód:

```
1: Initialize centers
2: while number_of_iterations < max_iterations do
3:   — Stochastic version —
4:   for each input  $x$  in random order do
5:      $c^* \leftarrow$  find center closest to  $x$ 
6:      $c^* \leftarrow c^* + \alpha.(x - c^*)$ 
7:   — Batch version —
8:    $C \leftarrow$  list of sets (clusters)
9:   for  $i$  in  $1, \dots, k$  do
10:     $C[i] \leftarrow$  set of inputs that center  $c_i$  is closest to
11:   for  $i$  in  $1, \dots, k$  do
12:     $c_i \leftarrow \operatorname{mean}(C[i])$ 
13:   if no center  $c_i$  was moved then
14:     found stable solution, quit
```

Úloha (1b): Do pripravenej kostry programu doprogramujte funkciu $k_means(...)$ prvým spôsobom (stochastická metóda).

Úloha (0.5b - bonus): Do pripravenej kostry programu doprogramujte funkciu $k_means(...)$ druhým spôsobom (batch metóda).

Pre prehľadnosť kódu môžete dorobiť a využiť navrhnuté funkcie $find_best_center(pixel_color)$ alebo $split_pixels_to_clusters(num_clusters)$.

V programe je pripravené:

- `self.pixels` : list všetkých pixelov obrázka, každý pixel je list [R, G, B] (napr. `self.pixels[42] == [255, 0, 0]`)
- `self.centers` : list k centrier, každé centrum je rovnako list [R, G, B] (napr. `self.centers[3] == [250, 10, 10]`)
- `distance(a, b)`: spočíta vzdialenosť dvoch farieb