Tema 1 - Metode numerice Clustering. Algoritmul K-Means

Descriere generala

Scopul acestei teme este familiarizarea studentilor cu limbajul de programare GNU Octave prin implementarea unui algoritm de clustering. Tema consta in 5 subpuncte dependente intre ele care valoreaza, in total, 115 puncte.

Ce inseamna clustering?

Algoritmii de clustering sunt folositi pentru a grupa anumite obiecte in multimi separate astfel incat fiecare obiect sa aiba mai multe caracteristici in comun cu obiectele din multimea careia apartine decat cu alte obiecte. In cadrul temei de fata se cere implementarea unui astfel de algoritm, K-Means.

Fisiere de intrare

- .param – contine numarul de clustere (grupuri) ce trebuie formate, numar notat pe parcursul temei cu NC.
- .points – contine multimea de puncte pe care va fi aplicat algoritmul K-Means; aceasta este reprezentata printr-o matrice, fiecare punct pe cate o linie, fiecare din coordonatele x, y, z pe cate o coloana.

Testare

Tema va fi testata folosind functia implementata in fisierul “test.m”, inclus in arhiva. Sunteti incurajati sa testati tema inainte sa o trimiteti. Scriptul de test va afisa punctajul final pentru taskurile 2 si 4, precum si cele doua grafice cerute in tema; acestea vor fi, insa, verificate manual.

Atentie! Nu implementati tema in urmatoarele fisiere, ele vor fi inlocuite la testare:

- graph.m
- reorder_by_x.m
- test_clustering_by_index.m
- test_cost_by_index.m
- test.m

Task 1 (Citirea datelor de intrare) 5p

Fisier .m: read_input_data.m
Descriere: Citirea datelor de intrare din cele doua tipuri de fisiere. Functia este folosita in fisierele clustering.m si compute_cost.m (pe care nu este nevoie sa le modificati).
Parametri: caile catre cele doua fisiere (.param si .points)
Intoarce: valoarea NC si matricea "points", citite din fieire.
Verificare: puteti vizualiza setul de puncte citit apeland "view_points(points)".
Sugestii: save/load, citiri C-like

Task 2 (Implementarea K-Means) 30p

Fisier .m: clustering_pc.m
Descriere: calculueaza coordonatele a NC puncte in spatiu (numite centroizi), reprezentand centrele de masa ale clusterelor.
Pasii algoritmului K-Means:

1. Se aleg cei NC centroizi din multimea de puncte, aleator, avand grija ca acestia sa nu se suprapuna.
2. Fiecare punct din multime este atribuit grupului reprezentat de cel mai apropiat centroid (folosind notiunea de distanta euclidiana)
3. Recalculam pozitiile centroizilor ca fiind centrele de masa ale punctelor atribuite fiecarui grup in parte.
4. Repetam pasii 2, 3 pana cand pozitiile centroizilor nu se mai modifica.

Parametri: multimea de puncte (in formatul descris mai sus) si NC, numarul de grupuri ce trebuie formate.
Intoarce: un set de NC puncte (pozitiile finale ale centroizilor).
Verificare: se poate utiliza functia test (vezi sectiunea "Testare").
Sugestii: norm, find, mean

Task 3 (Vizualizarea rezultatului K-Means) 20p

Fisier .m: view_clusters.m
Descriere: se doreste vizualizarea rezultatului algoritmului K-Means sub forma unui grafic similar celui generat de functia view_points, dar colorat. Toate punctele care apartin unui anumit grup trebuie sa aiba aceeasi culoare, diferita de culorile altor grupuri. Alegerea culorilor pentru grupuri este la latitudinea studentilor, dar acestea trebuie sa poata fi usor deosebite.
Parametri: multimea de puncte si multimea centroizilor calculati la taskul anterior (acestea sunt date, asa cum s-a mentionat mai sus, sub forma de matrice cu 3 coloane).
Verificare: se poate utiliza functia graph (vezi sectiunea "Testare"). Rezultatul ar trebui sa semene cu cel prezentat in acest enunt.
Sugestii: scatter3

Task 4 (Functie de cost) 20p

Fisier .m: compute_cost_pc.m
Descriere: definim costul unui cluster (grup) ca fiind suma distantelor de la centroid la fiecare din punctele ce apartin clusterului respectiv. Astfel, costul unei solutii este egal cu suma costurilor tuturor clusterelor.
Parametri: cele doua matrice cu care ati lucrat si la taskul 3 (points si centroids).
Intoarce: suma costurilor tuturor clusterelor
Verificare: se poate utiliza functia test (vezi sectiunea "Testare").
Sugestii: sum, norm

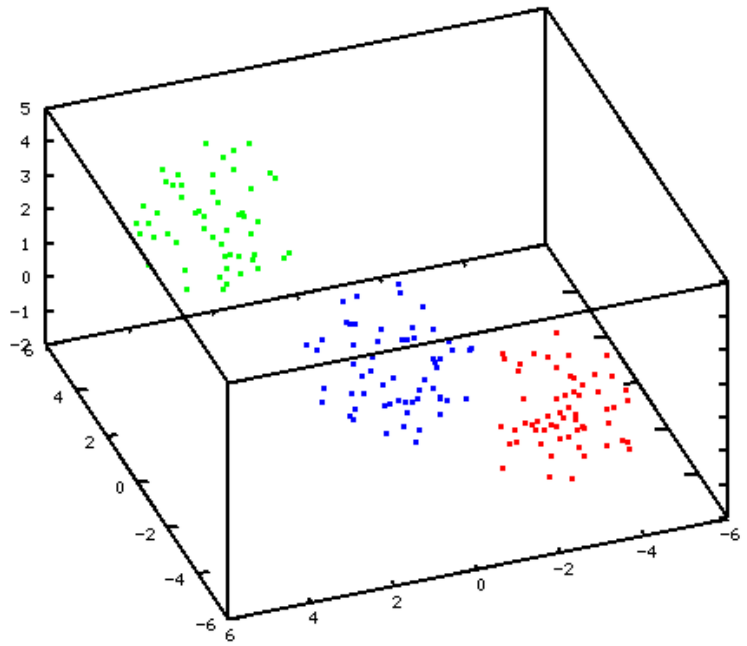
Task 5 (Determinarea numarului de grupuri) 30p

Fisier .m: view_cost_pc.m
Descriere: Numarul de clustere poate fi determinat automat in anumite situatii, iar acest task vizeaza implementarea uneia dintre metodele folosite in practica. Metoda consta in trasarea graficului costului total in functie de numarul de clustere si analiza acestuia: s-a observat ca marirea numarului de clustere peste o anumita valoare nu aduce scaderi semnificative ale costului total. Pentru a trasa graficul cerut, se vor inregistra costurile rezultatelor obtinute prin apelarea clustering_pc cu NC din multimea {1 .. 10}.
Parametri: calea catre un fisier ".points".
Verificare: se poate utiliza functia graph (vezi sectiunea "Testare"). Rezultatul ar trebui sa semene cu cel prezentat in acest enunt.
Sugestii: plot

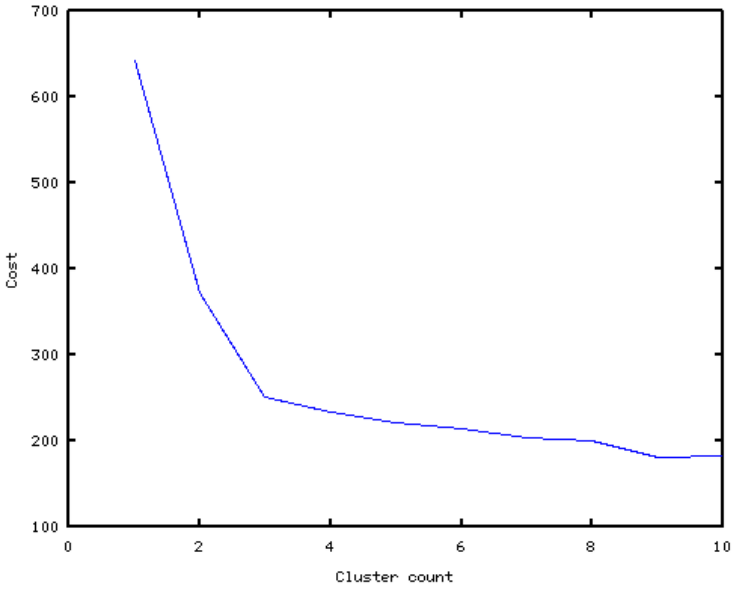
README 10p

Se acorda 10 puncte pentru README. Acesta trebuie sa contina detaliile implementarii, precum si eventuale probleme pe care le-ati intalnit pe parcursul rezolvarii temei, impreuna cu solutiile gasite. Feedbackul este, de asemenea, bine venit.

Grafice taskuri 3, 5



Task 3 - Vizualizarea clusterelor



Task 5 - Costul solutiei in functie de numarul de clustere