Gruparea pe baza similarității a datelor de antrenament

Generalităţii

Problema de grupare automată a datelor, întâlnită în literatură sub numele de *clustering,* reprezintă procesul de identificare automată a exemplelor de antrenament care au caracteristici similare (atribute similare) și gruparea acestora împreună. Grupurile astfel rezultate poartă denumirea de *clustere*. Un cluster este o colecție de elemente din setul de date de antrenament care sunt similare unul față de altul și sunt nesimilare (diferite) față de elementele care nu fac parte din acel cluster. Un cluster trebuie să satisfacă două condiții: fiecare cluster trebuie să conțină cel puțin un element și fiecare element trebuie să fie inclus doar într-un singur cluster.
În literatură o categorie mare de algoritmi de *clustering* încearcă, pe cât posibil, pe baza unor idei relativ simple să rezolve problema de grupare a datelor de intrare în funcție de similaritate. După realizarea clusterelor acestea sunt identificate/reprezentate print-un element unic numit ”centrul clusterului” sau ”centroidul clusterului”, acesta fiind elementul care va caracteriza ulterior toate celelalte elemente care aparțin clusterului.
Pentru acest laborator prezentăm algoritmul *k-Means* care este unul dintre cei mai reprezentativi algoritmi de clustering din categoria algoritmilor partiționali. Ideea de bază a acestui algoritm este folosită de foarte mulți alți algoritmi de clustering. Acest algoritm este un algoritm partițional care are la bază ideea de a împărți datele în funcție de similaritatea lor și de a reprezenta clusterul rezultat print-un singur element reprezentativ al clusterului numit centroid.

Algoritmul *k*-Means

Algoritmul *k*-Means este cel mai popular și mai utilizat algoritm în aplicații științifice și industriale. Numele acestui algoritm provine de la reprezentarea a k clusteri C_i ($i \in \overline{1,k}$) a căror valoare medie (mean) este c_i calculată ca și centru de greutate al tuturor exemplelor din C_i (al clusterului). Valoarea parametrului k se stabilește la începutul parcurgerii algoritmului și reprezintă numărul de centroizi care se doresc a se obține. Similaritatea elementelor dintr-un cluster C_i se calculează raportat la centrul de greutate al clusterului. Algoritmul de clustering se bazează pe ideea că se poziționează în spațiul de reprezentarea al datelor de antrenament un număr de centroizi, iar aceștia în procesul de învățare vor încerca să-și găsească pozițiile optime care vor caracteriza cel mai bine toate datele de antrenament. Acest algoritm face parte din categoria algoritmilor de învățare nesupervizată deoarece se specifică ca și date de intrare doar exemplele de antrenament fără a se specifica și categoria (centrul) de care acestea aparțin. La această categorie de algoritmi specificăm doar datele pe care vrem să le învețe fără a specifica exact ce vrem să învețe.
Scopul acestui algoritm este de a obține o similaritate mare între elementele din cadrul unui cluster comparativ cu o similaritate mică între elementele din clusteri diferiți.

Pașii algoritmului:

1. Se alege aleator un numar de „centre de clase” sau ”centroizi” în funcție de câte categorii (clusteri) vrem să obținem în final. Acesta este reprezentat de valoarea lui k din numele algoritmului. Fiecare centroid va fi reprezentat într-un spațiu care va avea aceeași dimensiune cu spațiul de reprezentare al datelor de antrenament.

2. Pentru fiecare centru se generează aleator toate atributele acestuia. Numărul de atribute va fi reprezentat de dimensiunea spațiului de reprezentare al datelor de antrenament. Acest pas coincide cu stabilirea aleatoare în spațiul de reprezentare al datelor de antrenament a poziției centrelor claselor (centroizilor). Algoritmul poate porni și cu centroizi inițializați altfel. Se pot propune si alte metode de inițializare a centroizilor.
3. Pentru fiecare exemplu de antrenament din setul de date de antrenament se calculează similaritatea între acesta şi toți centroizii generați la pasul 2 (toate centrele). Exemplul respectiv se va încadra la centroidul față de care este cel mai similar (grupare pe baza similarității). Există mai multe metode pentru calculul similarităţi. În paragraful următor vom prezentat 3 astfel de metode. Se pot propune și alte metode.
4. După gruparea tuturor exemplelor din setul de date de antrenament considerăm că toate exemplele grupate la un centroid formează un cluster. Pentru ca un centroid să caracterizeze cât mai bine datele respective el ar trebui să stea în mijlocul lor. Pentru aceasta se calculează centru de greutate al tuturor exemplelor grupate într-un cluster. Astfel pentru fiecare cluster vom avea câte un nou centru de greutate.
5. Se mută „centroidul clasei" în centru de greutate calculat la punctul 4.
6. Se reia algoritmul de la pasul 3 atâta timp cât funcția de convergență se modifică. Funcția de convergență notată *E* se calculează ca fiind suma tuturor distanțelor de la fiecare exemplu dintr-un cluster la centroidul de care aparține. Dacă funcția de convergență nu se mai modifică putem spune că algoritmul s-a încheiat.
   Ecuația funcției de convergență este:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} d(p, c_i)$$

unde c_i reprezintă centroidul clasei, *p* este un exemplu din setul de date de antrenament iar $d(p,c_i)$ reprezintă distanta dintre exemplu și centroid.

Temă

Pornind de la fișierul cu datele de intrare generat de algoritmul prezentat în Laboratorul 1 să se realizeze o aplicație care folosind algoritmul *k*-Means va grupa automat datele de intrare pe baza similarității (poziției) acestora. Aplicația realizată va avea două etape:

- o etapă de antrenare în care pe baza datelor de intrare va încerca să găsească poziția optimă a centroizilor folosind algoritmul descris mai sus;
- o etapa de testare în care va reprezenta corespunzător toate punctele din spațiul de reprezentare al datelor pe baza similarității acestora cu centroizii găsiți.

În etapa de antrenare din fișierul de intrare se va prelua de pe fiecare linie doar coordonatele punctelor fără a se extrage sau păstra și informația legată de centrul pentru care a fost generat punctul. Acestea vor forma setul de date de antrenament pentru acest algoritm. În setul datelor de intrare vom avea pentru fiecare punct coordonata pe *x* şi pe *y* a acestuia. Se va antrena algoritmul *k-Mean*s pe baza acestor date. Se recomandă alegerea unui număr între 2 şi 10 centroizi iar pentru o mai bună vizualizare fiecărui centroid i se va atașa și o culoare corespunzătoare.

În etapa de testare se va lua pe rând fiecare punct din spațiul de reprezentare al datelor și se va calcula similaritatea dintre acel punct si toți centroizii. Punctul va fi colorat corespunzător culorii centroidului care este cel mai similar cu el. Se recomanda în etapa de testare alegerea unei nuanțe mai deschise a aceleiași culori pentru fiecare centroid.

Obs:

1. Se recomandă ca inițial toate punctele să fie de aceeași culoare iar ulterior să se coloreze acestea corespunzător după fiecare execuție a pasului 3 pentru o mai bună vizualizare a evoluţiei algoritmului.
2. Nu este obligatoriu generarea aleatoare a centrelor. Se pot folosi şi alte metode de alegere a centrelor. Propuneţi astfel de metode.
3. Centrul de greutate al unei clase poate reprezenta, de exemplu, media aritmetică a tuturor elementelor conținute în acel cluster.
4. Problema majora a acestui algoritm este că poziţionarea centrelor pe parcursul algoritmului nu se face în funcţie de informaţiile care vin din mediul de intrare ci, mai degrabă, pe idea de a acapara cât mai multe puncte.
5. O altă problemă majoră este poziţionarea iniţială a centrelor şi alegerea numărului acestora.

Metode de calcul a similarităţii

Există mai multe metode de calcul a similarității între doi vectori. Fiecare metodă se alege în funcţie de domeniu aplicabilităţii metodei respective. Printre cele mai cunoscute și folosite metode sunt calculul distanței Euclidiene sau a cosinusului unghiului dintre doi vectori.

Prezentăm în acest laborator doar trei metode:

1. Calculul similarităţii folosind distanta Euclidiană:

$d(\vec{x},\vec{x}') = \sqrt{\sum_{i=0}^{n}(x_i - x'_i)^2}$, unde *n* reprezintă nr. de trăsături caracteristice iar *x* şi *x'* reprezintă cei doi vectori de intrare.

2. Calculul similarităţii folosind distanta Manhattan:

$$d(\vec{x},\vec{x}') = \sum_{i=0}^{n}(|x_i - x'_i|)$$

3. Calculul similarității folosind cosinusul unghiului între 2 vectori:

$$d(\vec{x},\vec{x}') = \frac{<\vec{x},\vec{x}'>}{|\vec{x}|\cdot|\vec{x}'|}, \quad unde <\vec{x},\vec{x}'> = \sum_{i=0}^{n} x_i \cdot x_i' \quad si \quad |\vec{x}| = \sum_{i=0}^{n} x_i \cdot x_i$$

Întrebări

1. Reușește acest algoritm să funcţioneze bine pentru datele de intrare avute la dispoziţie?
2. Daţi exemple de cazuri în care acesta nu funcţionează bine?
3. Încercaţi şi alte metode de inițializare a centrelor claselor.
4. Propuneţi alte metode de calcul a similarităţii.
5. Propuneţi alte metode de calcul a centrului de greutate al unei clase.