2d-pistejoukon klusterointi k-keskiarvo -algoritmilla

Tietorakenteet ja algoritmit II
Ohjaaja: Tomi Suomi

Laatijat:
Kimmo Kettunen (68220) kmpket@utu.fi
Timi Suominen (505890) tijusuo@utu.fi

31.3.2014
Turku

Sisällys

1 Tehtävän kuvaus ja analysointi

Tehtävänä oli ryhmitellä 2d-pistejoukon pisteet merkityksellisiin luokkiin (myöhemmin klustereihin). Tarkasteltavaksi algoritmiksi valitsimme yleisesti käytetyn k-keskiarvot (k-means).

Toteutimme algoritmin kirjastofunktiona, jota kutsutaan syöttämällä sille halutut pisteet, klustereiden lukumäärä ja iteraatioiden maksimimäärä. Funktio palauttaa klustereiden keskipisteet ja liittää syötettyihin pisteisiin sen klusterin numeron, johon se kuuluu.

Ainoa varsinainen käytön rajoitus on se, että klustereiden määrä pitää olla pienempi, kun syötteenä annettujen pisteiden määrä. Jatkokehitys voisi olla esimerkiksi Piste-luokan muuttaminen rajapinta-luokaksi ja joidenkin toiminnallisuuksien siirtäminen siihen; kuten pisteiden välisen etäisyyden laskeminen ja pisteiden yhdistäminen. Näin algoritmi toimisi myös useampi ulotteisille pisteille ja/tai eri etäisyyden määrittely tavoille.

2 Ratkaisuperiaate

K-keskiarvo algoritmissa pisteet klusteroidaan k määrään klustereita pisteiden keskinäisen etäisyyden perusteella. Tämä tapahtuu kahdessa vaiheessa, joita toistetaan kunnes lopputulos ei enää muutu tai on suoritettu haluttu määrä iterointeja. Ensimmäisessä nk. sijoitusvaiheessa pisteet sijoitetaan kukin siihen klusteriin, jonka keskipiste on pistettä lähinnä. Toisessa nk. päivitysvaiheessa, klusterin keskipiste lasketaan uudelleen siihen liitettyjen pisteiden arvojen perusteella.

K-keskiarvot-algoritmin pseudokoodi:
Syöte: S (pisteet), k (klustereiden lukumäärä)
Tuloste: klusterit
1: Alusta k määrä klusterien keskipisteitä
2: **while** lopetusehto epätosi
3: Sijoita pisteet lähimpään klusteriin
4: Päivitä klustereiden keskipisteet sijoitusten perusteella
5: **end while**

Algoritmille annetaan syötetietoina klusteroitavat pisteet, haluttu klustereiden määrä ja iteraatioiden maksimimäärä. Loimme pisteille oman luokan Piste, jolla on attribuutteina x- ja y-koordinaatit sekä klusterin numero, johon se tulee kuulumaan.

Aluksi pisteiden joukosta valitaan satunnaisesti k kappaletta ensimmäisiksi klustereiden keskipisteiksi. Seuraavaksi näitä lähdetään tarkentamaan iteroimalla. Jokaiselle pisteelle lasketaan etäisyys eri klustereihin ja piste sijoitetaan sitä lähinnä olevaan klusteriin. Seuraavaksi lasketaan kuhunkin klusteriin kuuluvien pisteiden perusteella klusterin uusi keskipiste. Koska klusterien koordinaatit ovat (lähes) aina approksimaatioita eikä tarkkaan tulokseen siten päästä, on iteraatiokerrat rajoitettava johonkin mielekkääseen määrään. Iteraatiokerralla tässä tarkoitetaan sitä operaatiojoukkoa, joka tarkentaa klusteriden sijaintia 2d-pistejoukossa (pseudokoodin rivit 3-4). Iteraatioiden loputtua (eli rivit 2-5) palautetaan klustereiden keskipisteet ja parametrina annettuihin pisteisiin on tallennettu tieto siitä, mihin klusteriin kukin piste kuuluu.

3 Ohjelman ja sen osien kuvaaminen

3.1 Luokkien kuvaus

3.1.1 class Piste

Vakiot ja attribuutit:

`double X`

Pisteen x-koordinaatti.

`double Y`

Pisteen y-koordinaatti.

`int Group`

Klusterin numero

3.1.2 class Kmeans2d

Julkiset metodit:

`GetKmeans2d``(List<Piste> pisteet, int k, int maxIter)`

`List<Piste> pisteet`

Listaus kaikista pisteistä.

`int k`

Klustereiden ennalta valittu määrä.

`maxIter`
Ohjelmallinen rajoite iteraatio- eli laskentakierroksille.

Metodi palauttaa suorituksen päätyttyä keskipisteiden tarkennettuine sijaintineen.

Virhetilanteissa heitetään `exception` eli poikkeus tilanteissa, joissa pisteitä on vähemmän parametrina `k` annettu määrä.

Sivuvaikutuksena pisteet liitetään klustereihin.

Yksityiset metodit:

liitaLahimpaanKeskukseen(List<Piste> pisteet, List<Piste> keskipisteet)

List<Piste> pisteet

Listaus kaikista pisteistä

List<Piste> keskipisteet

Klustereiden keskipisteet

Metodi päivittää kunkin syötteenä saadun pisteen (pisteet) kuulumaan lähimpään keskipisteeseen sen etäisyyden perusteella eri keskipisteistä (keskipisteet). Lähin keskipiste etsitään LahinKeskus()-funktion avulla.

Palauttaa true, jos yksikin piste on vaihtanut klusteria, false muulloin.
Sivuvaikutuksena pisteen Group-attribuutti muuttuu, jos sen uusi klusteri on eri kuin edellinen.

lahinKeskus(Piste vertailupiste, List<Piste> keskipisteet)

Piste vertailupiste

Piste, jolle etsitään lähin keskus

List<Piste> keskipisteet

Keskipisteet, joihin vertailupisteen etäisyys lasketaan

Metodi palauttaa attribuuttina annetun pisteen (vertailupiste) lahinnä olevan klusterin numeron. Etäisyys lasketaan käyttäen apuna EtaisyydenNelio(Piste a, Piste b) -funktiota.

etaisyydenNelio(Piste a, Piste b)

Palauttaa kahden parametreina saadun pisteen (a ja b) välisen euklidisen etäisyyden.

paivitaKeskipisteet(int k, List<Piste> pisteet)

int k

Klusterien lukumäärä

List<Piste> pisteet

Klusteroitavat pisteet

Metodi laskee klustereiden uusien keskipisteiden sijainnit. Keskipisteen sijainti lasketaan laskemalla ensiksi klusteriin liitettyjen pisteiden kehyksen, eli kaikkien pisteiden koordinaattien yhteenlasketun leveyden ja korkeuden. Lopuksi tämän kehyksen keskipiste lasketaan ja siitä tulee uusi klusterin keskipiste.

3.1.3 class Testi

Testi-luokan tarkoitus on havainnollistaa graafisesti k-keskiarvoalgoritmin ja siten myös Kmeans2d-luokan toimintaa.

main (String[] args)

`String[] args`

Komentorivargumentit, joita ei käytetä ohjelmassa mihinkään.

Main-metodissa luodaan satunnaisesti kolmesta yhdeksään klusteria. Klusterit luodaan valitsemalla klusterien (eli keskipisteiden) määrä väliltä [3,9]. Keskipisteen sijainnin päättämisen jälkeen keskipisteen ympärille luodaan satunnaisesti pisteitä. Pisteitä luodaan 100-200 kappaletta maksimissaan sadan pisteen päähän keskipisteestä.

Tämän kaltainen lähestymistapa on graafisen esityksen havainnollistamiseksi tärkeä. Näin saadaan luotua visuaalisesti helposti erottuvia keskipisteitä ja algoritmin voidaan todeta silmämääräisesti toimivaksi. Toinen tapa luoda pisteet koordinaatistoon olisi arpoa ne sinne satunnaisesti. Tällöin tosin pisteet jakautuisivat suhteellisen tasaisesti koordinaatistoon, eikä algoritmin toiminta ole visuaalisesti mielenkiintoista.

Lopuksi metodissa kutsutaan pisteille Kmeans2d-luokan GetKemans2d –metodia. Metodin palautusarvo annetaan Testi-luokan kontstruktorille joka puolestaan plottaa pisteet näkyville.

`Testi(List<Piste> pisteet, List<Piste> kpt)`

`List<Piste> pisteet`

Kaikki pisteet.

`List<Piste> kpt`

Keskipisteet.

Testi-konstruktori ottaa vastaan kaikki pisteet, sekä keskipisteet ja asettaa nämä arvot luokkamuuttuujiin.

paintComponent(Graphics g)

`Graphics g`

Piirtoalue

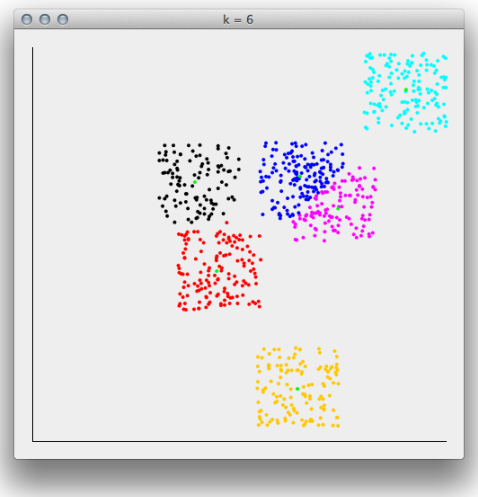
Funkito piirtää määrätyt pikselit grafiikka-alueelle g. Tässä tapauksessa funktio piirtää Pisteet 2d koordinaatistoon sekä kpt:t (keskipisteet) samaan koordinaatistoon. Jokainen klusteri, eli merkityksellinen pistejoukko saa yhden muista klustereista erottuvan yhtenäisen värin. Keskipisteet piirretään aina vihreällä värillä.

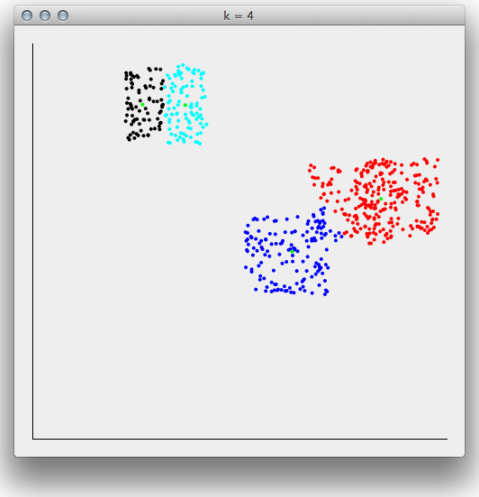
Lopputuloksena syntyy vaikutelma pistejoukoista, jotka on värjätty klustereittain ja klustereiden keskipisteet on havainnollistettu vihreällä pisteellä.

4 Testausjärjestely

Ohjelmiston toiminta verifioidaan visuaalisesti. Ohjelmisto käynnistetään, jolloin 2d graafin kaikki pisteet plotataan näkyville. Jokainen piste värjätään klusterin värin mukaiseksi. Esimerkiksi ensimmäiseen klusteriin kuuluvat sinisellä, toiseen punaisella ja kolmanteen oranssilla. Graafisen käyttöliittymän otsikkona ilmoitetaan ennalta valittu klustereiden määrä. Tällöin voidaan havinnoida, jos ilmoitettu klustereiden määrä poikkeaa plotatsta määrästä.

Alla on kuvaruutukaappaukset kahdesta eri testiajosta, jossa ensimmäisessä algoritmi onnistuu kohtuullisen hyvin vaikka klusterit ovatkin osittain päällekkäin. Toisessa puolestaan algroritmin alkuperäiset klustereiden keskipisteiden valinta ei ole onnistunut ja lopputulos ei vastaa odotettua tulosta, vaan yksi alkuperäisistä joukoista on jaettu kahdeksi klusteriksi (musta ja cyan) ja kaksi alkuperäistä joukkoa on yhdistetty yhdeksi klusteriksi (punainen).





5 Liitteet:

5.1 Alkuperäinen tehtävänanto

25. 2D-pistejoukon klusterointi: Pistejoukon samankaltaiset pisteet jaetaan erillisiin (merkityksellisiin luokkiin). Useita mahdollisia menetelmiä, (esim. k-keskiarvot, hierarkkinen ryvästäminen).