DOB valmennus

Data-analyysi

Koneoppiminen

30.08.2017

6Aika

Uudenmaan liitto
Nylands förbund

Euroopan unioni
Euroopan aluekehitysrahasto

Vipuvoimaa EU:lta 2014–2020

Datasta oivalluksia ja bisnestä

Data-analytiikan menetelmien valmennusmateriaali

1. Johdatus tiedonlouhinnassa käytettäviin laskennallisiin menetelmiin

Henry Joutsijoki

Sisältö

Sisältö

- Tiedonlouhinta ja koneoppiminen
- Oppimisen eri muodot
- Klusterointimenetelmät:
  1. K-means-algoritmi
  2. Hierarkkinen klusterointi
- Mallin valitseminen ja arviointi, ristiinvalidointi ja sekaannusmatriisi
- K:n lähimmän naapurin menetelmä

Sisältö

Alustusta

- Elämme yhteiskunnassa, missä käsitteet tehokkuus, nopeus, tarkkuus, maksimointi ja minimointi korostuvat yhä useammin eri asiayhteyksissä.
- Nykyinen taloudellinen tilanne johtaa siihen, että olemassaolevia resursseja pitää käyttää yhä paremmin, monipuolisemmin ja tehokkaammin.
- Tästä seuraa myös se, että uusia aluevaltauksia ja innovaatioita täytyy kehittää perinteisten alojen rinnalle.
- Kilpailu on kuitenkin erittäin kovaa joka sektorilla, sillä samankaltaisia ideoita syntyy ympäri maailmaa ja kilpajuoksu kärkipaikan saavuttamiseen on alati käynnissä.

Alustusta

- Viimeisen vuosikymmenen aikana maailmalla kuin myös Suomessa on syntynyt kasvava kiinnostus erilaisten aineistojen/datojen analysointiin.
- Digitaalisessa muodossa olevan tiedon määrä on jo nykyään valtava ja kasvaa jatkuvasti.
- Teknologian kehittymisen myötä dataa kertyy lukuisista eri lähteistä ja kerätyt aineistot voivat muodostaa esimerkiksi hyvin suuria tietokantoja.
- Viime vuosina “hittituotteeksi” ja uusimmaksi trendiksi maailmalla onkin tullut ns. Big Data.

Mistä saamme dataa?

- Esimerkkejä datalähteistä:
  1. Pankkien maksutapahtumat
  2. Kännykkäkamera ym. muut kamerat
  3. Kauppojen/kauppaketjujen ostotapahtumatiedot
  4. Terveydenhuollon potilaiden mittaukset/testit (esim. verikoe)
  5. Liikenne
  6. Ilmasto
  7. Sensorit yleisesti ottaen
  8. Teollisuusprosessit
  9. Sosiaalinen media ja muut web-aineistot
  10. Hakukoneiden keräämä data

Datan luonne

- Data voidaan jakaa karkeasti kahteen eri tyyppiin riippuen sen lähteestä:
  1. Järjestetty data (esim. kyselylomake) rajoittuu tietylle toimialalle ja data on luonteeltaan diskreettiä. Lisäksi dataa säilytetään esim. Excel-tiedostossa tai relaatiotietokannassa.
  2. Järjestämätön data puolestaan on sellaista, jota ei voida suoraviivaisesti kategorisoida johonkin luokkaan. Järjestämätöntä dataa ovat esimerkiksi kuvat, videot, PDF-tiedostot, sähköpostit ja sosiaalisen median tuottamat aineistot (Twitter, Facebook, ...).

Miksi keräämme dataa?

- Datan sisältämässä informaatiossa nähdään potentiaalia.
- Uudet ammatit ja liiketoiminta liittyvät datan sisältämän informaation käyttöön.
- Datan käsittelymenetelmien kehittäminen (mm. Big Datan yhteydessä) luo uusia työmahdollisuuksia. Suurten tietomassojen käsitteleminen ja tallettaminen vaativat uudenlaisia teknisiä toteutustapoja.
- Datasta voidaan löytää uusia yhteyksiä eri asioiden välille tai vahvistaa olemassaolevia ennakkokäsityksiä.

Sisältö

Yhteys tiedonlouhintaan

- Kerätyn “raakatiedon” suuren määrän vuoksi on pitänyt kehittää tietojenkäsittelymenetelmiä (erityisesti tiedonlouhinnan menetelmiä), jotta oleellinen informaatio saadaan esille datasta.
- Tiedonlouhinta voidaan määritellä seuraavasti:

  Tiedonlouhinta on (usein laajojen) havaintotietojen, datajoukkojen tietojen välisten suhteiden etsimistä ja datan ominaisuuksien yhteenvedon tuottamista uusin tavoin, jotka ovat sekä ymmärrettäviä että hyödyllisiä käyttäjän näkökulmasta.

Tiedonlouhinta

- Tiedonlouhinnalla saatuja ominaisuuksia ja yhteenvetoja kutsutaan yleensä *malleiksi* tai *hahmoiksi*. Esimerkkejä malleista ovat mm. erilaiset yhtälöt, assosiaatiosäännöt, klusterit, graafit ja puurakenteet.
- Tiedonlouhinta käsittelee monesti dataa, joka on kerätty muuta tarkoitusta varten kuin itse tiedonlouhintaa varten (esim. pankin tapahtumatiedot).
- Tiedonlouhinta on siis eri asia kuin tilastotiede, jossa aineistoa pääasiassa kerätään vastaamaan tiettyihin kysymyksiin (tutkimuskysymykset/hypoteesit).

Tiedonlouhinta ja data-analyysi

- Tiedonlouhinta on data-analyysin yksi osa-alue, joka on keskittynyt mallintamiseen ja tietämyksen muodostamiseen erityisesti ennustamisen näkökulmasta.
- Tiedonlouhinta ei suinkaan ole ainoa data-analyysin osa-alue, vaan data-analyysi on monipuolinen kokonaisuus, joka sisältää monia osa-alueita lähtien datan keräämisestä ja siivoamisesta aina datan visualisointiin ja tulosten analysointiin.

Tiedonlouhinnan tehtäviä

- Tiedonlouhinnan osa-alueet voidaan ryhmitellä viiteen ryhmään:
  1. Tutkiva data-analyysi (exploratory data analysis)
  2. Deskriptiivinen eli kuvaava mallintaminen (descriptive modelling)
  3. Ennustava mallintaminen (predictive modelling)
  4. Hahmojen ja sääntöjen etsiminen (discovering patterns and rules)
  5. Haku sisällön perusteella (retrieval by content)
- Tässä valmennuksessa keskitymme kuvaavaan ja ennustavaan mallintamiseen.

Kuvaava mallintaminen

- Tarkoituksena on kuvata koko data (tai sen generoiva prosessi).
- Esimerkkejä kuvaavasta mallintamisesta ovat todennäköisyysjakaumamallit, klusterointi (kutsutaan myös segmentoinniksi) ja muuttujien välisten riippuvuuksien mallintaminen.
- Esimerkiksi klusteroinnissa tavoitteena on löytää luonnolliset ryhmät datasta (esim. samanlaisten asiakkaiden ryhmittely samaan ryhmään/klusteriin).

Ennustava mallintaminen

- Tavoitteena muodostaa malli, jolla pystytään ennustamaan yhden muuttujan luokka tai kvantitatiivinen arvo muiden muuttujien avulla.
- Luokittelussa ennustettava muuttuja on luokkatyyppinen (kategorinen), kun taas regressiossa se on kvantitatiivinen (jatkuva-arvoinen).
- On kuitenkin huomattava, että “ennustaminen” tässä yhteydessä ei ole aikariippuvainen käsite, vaan ennustamisella voidaan tarkoittaa esimerkiksi potilaan diagnoosin ennustamista tai roskapostin tunnistamista.
- Ennustamista varten on olemassa laaja kokoelma tilastollisia menetelmiä sekä koneoppimismenetelmiä.

Ennustava ja kuvaava mallintaminen

- Ennustavan ja kuvaavan mallintamisen tarkempi jaottelu
- Tiedonlouhinta:
  1. Ennustava mallintaminen:
     - 1.1 Luokittelu
     - 1.2 Regressio
     - 1.3 Aikasarja-analyysi
  2. Kuvaava mallintaminen:
     - 2.1 Klusterointi
     - 2.2 Koostaminen
     - 2.3 Assosiaatiosäännöt
     - 2.4 Sekvenssien löytäminen
- Keskitymme luokitteluun ja klusterointiin tässä valmennuksessa.

Luokittelu

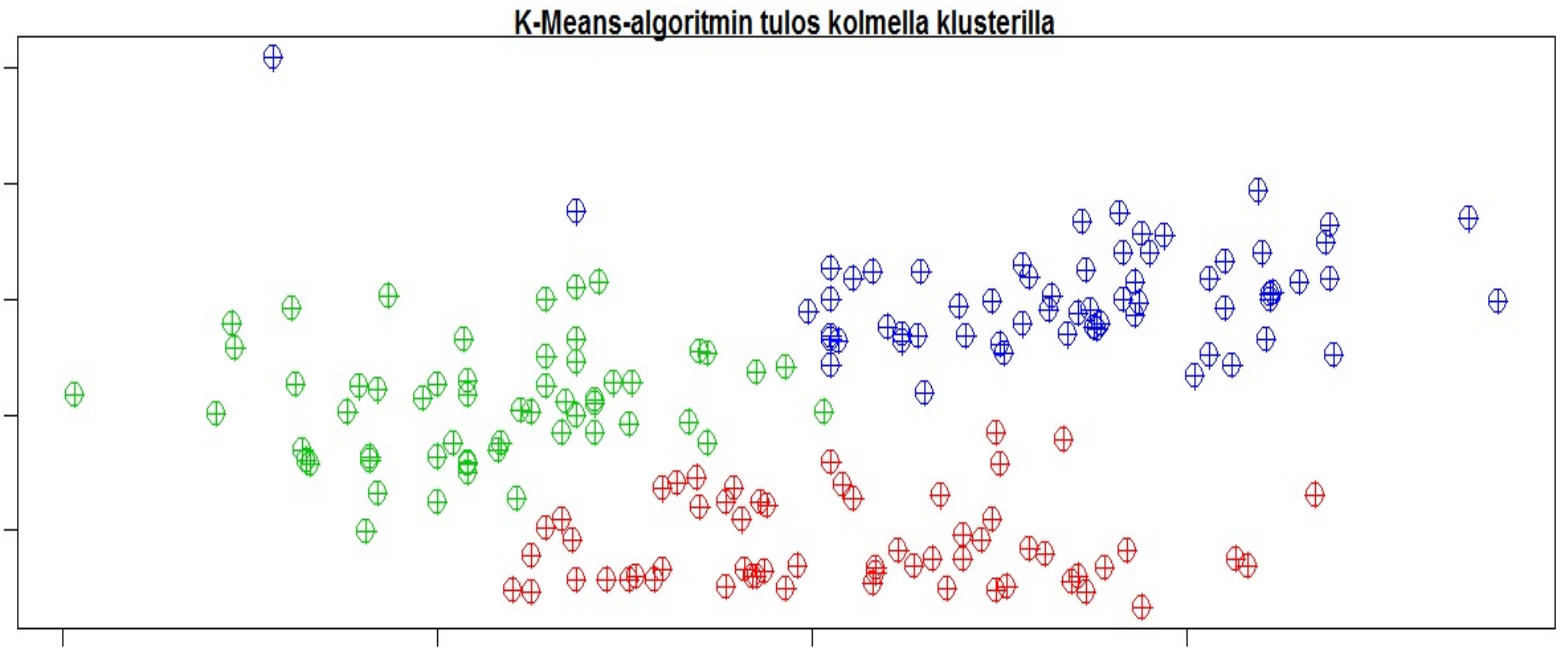
- Luokittelussa data kuvataan ennaltamäärättyihin ryhmiin, joita kutsutaan luokiksi. Luokat kuvataan yleensä diskreeteillä arvoilla.
- Luokittelua kutsutaan myös ohjatuksi oppimiseksi, koska aineiston sisältämien tapausten luokat ovat jo tiedossa tai ne määritetään ennen kuin itse tiedonlouhintaprosessi aloitetaan.

Klusterointi

- Klusterointi on luokittelun kaltainen toimenpide paitsi, että klustereiksi kutsuttuja ryhmiä ei ole määritelty etukäteen.
- Klusterointia kutsutaan myös ohjaamattomaksi oppimiseksi tai joskus myös segmentoinniksi.
- Klusteroinnissa tarkoituksena on jakaa data ryhmiin samankaltaisuuden perusteella käyttäen olemassaolevia muuttujia.
- Klusterin sisällä tulisi olla samankaltaisia havaintoja, mutta klustereiden väliset erot tulisivat olla suuria. Tällöin tulosten analysointi helpottuu.
- Monesti esiprosessoinnin jälkeen tiedonlouhinta aloitetaan klusteroinnilla.

Esimerkki klusteroinnista

- Aineisto: Wine dataset (UCI Machine Learning repository)
- Klustereiden lukumäärä: 3



Kuva: Esimerkki klusteroinnista

Sisältö

Koneoppimisen suhde tiedonlouhintaan

- Koneoppiminen ja tiedonlouhinta ovat hyvin läheisiä käsitteitä keskenään.
- Tiedonlouhinta voidaan lukea koneoppimisen sovellusalaksi.
- Koneoppimisen piirissä kehiteltyjä menetelmiä voidaan hyödyntää tiedonlouhinnassa.

Koneoppiminen

- Koneoppimiselle on olemassa lukuisia määritelmiä riippuen tarkasteltavasta lähteestä.
- Koneoppimisen lähtökohta on, että kone/tietokone suorittaa oppimisprosessin ihmisten asemesta.
- Mitä on oppiminen?
- Oppimiselle voidaan antaa seuraava määrittely:

  Oppiminen on algoritmin kyky parantaa sen suorituskykyä aikaisemman kokemuksen perusteella.

Koneoppiminen

- Toisin sanoen oppiminen on opetetun algoritmin kyky tehdä ennustuksia.
- Aikaisempi kokemus koneoppimisalgoritmien tapauksessa on “historiallista dataa”, joka on talletettuna esimerkiksi tietokantaan.
- Esimerkki: Algoritmi oppii tunnistamaan, mitkä sähköpostit ovat roskaposteja.
- Esimerkki: Suosittelujärjestelmät osaavat suositella käyttäjälle uusia tuotteita hänen aikaisempien hakujen perusteella.

Koneoppimisalgoritmien pääjaottelu

1. Ohjattu oppiminen (supervised learning)
2. Ohjaamaton oppiminen (unsupervised learning)

- Edellämainittujen kahden oppimismuodon lisäksi on olemassa vahvistusoppiminen (reinforcement learning), joka sijoittuu ohjatun ja ohjaamattoman oppimisen välimaastoon. Vahvistusoppimiseen emme perehdy tarkemmin tässä valmennuksessa.

Ohjattu oppiminen

- Ohjatussa oppimisessa datan havaintojen luokat/vasteet (diskreetti luokittelussa, jatkuva regressiossa) ovat etukäteen tiedossa.
- Jos luokkia ei ole etukäteen tiedossa, voidaan ohjatun oppimisen algoritmeja käyttää, jos datassa on klusteroituva rakenne. Tällöin ensin suoritetaan klusterointialgoritmi, jotta datan luonnolliset ryhmät tulevat esiin. Klusteroinnin jälkeen käyttäjä (sovellusalan asiantuntija) luokittelee klusterit ennen kuin ohjatun oppimisen piiriin kuuluvaa koneoppimisalgoritmia ryhdytään käyttämään.

Ohjattu oppiminen

- Jokaisen ohjatun oppimisen piiriin kuuluvalle koneoppimisalgoritmille täytyy antaa opetusjoukko, jota käyttäen algoritmi muodostaa mallin ennustamista varten.
- Opetusjoukko koostetaan olemassaolevasta aineistosta.
- Opetusvaiheen jälkeen algoritmille syötetään testijoukko, joka on riippumaton opetusjoukosta. Opetusjoukko ja testijoukko eivät saa samanaikaisesti pitää sisällään samoja havaintoja.
- Testijoukon perusteella määritämme, kuinka hyvin algoritmi on onnistunut uusien havaintojen ennustamisessa.

Ohjatun oppimisen menetelmiä

- Diskriminanttianalyysimenetelmät
- Päätöspuumenetelmät
- K:n lähimmän naapurin menetelmä
- Naiivi Bayes -luokittelija
- Tukivektorikone
- Neuroverkkomenetelmät

Ohjaamaton oppiminen

- Ohjaamatonta oppimista usein sanotaan klusteroinniksi. On kuitenkin huomattava, että myös assosiaatiosääntöjen etsiminen (“louhiminen”) kuuluu ohjaamattoman oppimisen piiriin.
- Klusteroinnissa luokkatietoa/vastetta havainnoille ei tarvita, vaan dataa voidaan analysoida ilman niitä.
- Klusteroinnin tavoitteena on löytää datasta ryhmät, joiden sisäinen samankaltaisuus on suuri ja eri ryhmien välinen samankaltaisuus on mahdollisimman pientä.

Ohjaamattoman oppimisen menetelmiä

- Itseorganisoituvat kartat (Self-Organizing Map)
- K-Means-algoritmi ja sen muunnelmat (K-Means algorithm)
- Hierarkkinen klusterointi (Hierarchical clustering)
- EM-algoritmi (Expectation–maximization algorithm)
- Neural Gas/Growing Neural Gas -algoritmit


K.J. Cios et al., *Data Mining: A Knowledge Discovery Approach*, Springer, 2007.


M. Juhola, *Tiedonlouhinta kurssin luentomateriaali*, Tampereen yliopisto, 2014.


I.H. Witten et al., *Data Mining: Practical Machine Learning Tools and Techniques*, 3rd ed, Morgan Kaufmann, 2011.

2. K-Means-algoritmi

Henry Joutsijoki

Sisältö

Tausta

- Klusterointi on datan havaintojen jakamista ryhmiin siten, että samankaltaiset havainnot ovat omissa ryhmissään (klustereissa) ja ryhmien väliset erot samankaltaisuuden suhteen olisivat mahdollisimman suuret.
- Klusterointialgoritmit voidaan jakaa kolmeen kategoriaan:
  1. Osittavat menetelmät (partitional clustering)
  2. Hierarkkiset menetelmät (hierarchical clustering)
  3. Malliperustaiset menetelmät (model-based clustering)
- K-Means-algoritmi kuuluu osittaviin menetelmiin.
- Osittavat klusterointialgoritmit ovat paljon käytettyjä menetelmiä käytännössä.

Osittavat menetelmät

- Jakavat datan siten, että kukin havainto kuuluu vain yhteen klusteriin.
- Klusterit yksinään ovat mahdollisimman samankaltaisia, mutta keskenään eroavaisia.
- Optimaalisen klusteroinnin löytäminen on laskennallisesti raskasta, koska mahdollisia jakoja on erittäin paljon.
- Jos esimerkiksi havaintojen lukumäärä $N = 100$ ja klusterien lukumäärä $c = 5$, on mahdollisia jakoja yli 10^{67} kappaletta.
- Käytännössä yleensä pystymme löytämään vain alioptimaalisen jaon havainnoille.

Osittavat menetelmät

- Perushaasteena osittavissa menetelmissä on kustannusfunktion määrittäminen, jolla mitataan klusteroinnin onnistumista.
- Yleensä kustannusfunktio määritellään havaintojen välisten etäisyyksien avulla.
- Monesti kustannusfunktiona käytetään klustereiden yhteenlasketun sisäisen vaihtelun minimointia. Mitä pienempi kustannusfunktion arvo on, sitä paremmin klusterointi on onnistunut ja kunkin klusterin havainnot ovat keskenään mahdollisimman samankaltaisia.

Sisältö

K-Means-algoritmi

- Tunnetuin, vanhin ja yksinkertaisin osittava klusterointimenetelmä.
- Tunnetaan myös nimellä Lloydin algoritmi.
- Algoritmin idea kehitettiin jo 1950-luvulla (Hugo Steinhaus, Stuart Lloyd).
- Käyttäjän kannalta K-Means-algoritmi on käytännöllinen, koska ainoat parametrit ovat klustereiden lukumäärä K, klustereiden alustaminen ja etäisyysmitta. Näistä parametreista klustereiden lukumäärän valitseminen on kaikista tärkein.

K-Means-algoritmin lähtökohta

- Olkoot $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$ havaintoja ja $\mathbf{x}_i \in \mathbb{R}^n$, $i = 1, 2, \ldots, N$. Olkoon klustereiden lukumäärä c.
- Oletetaan, että $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c$ ovat klustereiden edustajavektorit (prototyypit, keskipiste).
- Tavoite K-Means-algoritmissa on parantaa iteratiivisesti alussa muodostettua havaintojen jaottelua ja minimoida kustannusfunktion arvo päivittäen edustajavektoreita.

Kustannusfunktio

- K-Means-algoritmissa käytettävä kustannusfunktio on

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik} \|\mathbf{x}_k - \mathbf{v}_i\|^2,$$

  missä $\|\cdot\|^2$ on havainnon $\mathbf{x}_k$ ja edustajavektorin $\mathbf{v}_i$ välisen euklidisen etäisyyden neliö.
- Huom. Euklidisen etäisyyden neliö on laskennallisesti nopeampi laskea verrattuna varsinaiseen euklidiseen etäisyyteen, missä esiintyy neliöjuuri.
- Kustannusfunktioon liittyvä tärkeä käsite on jakomatriisi $U = [u_{ik}], i = 1, 2, \ldots, c, k = 1, 2, \ldots, N$, joka kohdistaa havainnot klustereihin.
- Jakomatriisin U kaikki alkiot ovat joko 0 tai 1.
- Jos k. havainto kuuluu i. klusteriin, on $u_{ik} = 1$ ja muutoin $u_{ik} = 0$.

Jakomatriisi

- Jakomatriisi täyttää kaksi ehtoa:
  1. Jokainen klusteri on epätriviaali. Se ei sisällä kaikkia havaintoja ja on epätyhjä.

$$0 < \sum_{k=1}^{N} u_{ik} < N, i = 1, 2, \ldots, c$$

  2. Jokainen havainto kuuluu yhteen klusteriin.

$$\sum_{i=1}^{c} u_{ik} = 1, k = 1, 2, \ldots, N.$$

Esimerkki jakomatriisista

- Esimerkki jakomatriisista:

$$U = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

- Jokainen rivi kuvaa yhden klusterin ja jokainen sarake esittää yhtä havaintoa.
- Näin ollen ensimmäinen klusteri sisältää havainnot $\{\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_6, \mathbf{x}_8\}$. Toisessa klusterissa on havainnot $\{\mathbf{x}_2, \mathbf{x}_3\}$ ja kolmannessa havainnot $\{\mathbf{x}_5, \mathbf{x}_7\}$.

K-Means klusteroinnin kuvaus

1. Valitse satunnaisesti edustajavektorit $\mathbf{v}_i, i = 1, 2, \ldots, c$ aineistosta.
2. Iteroi:
   - 2.1 Rakenna jakomatriisi U oheisen säännön perusteella:

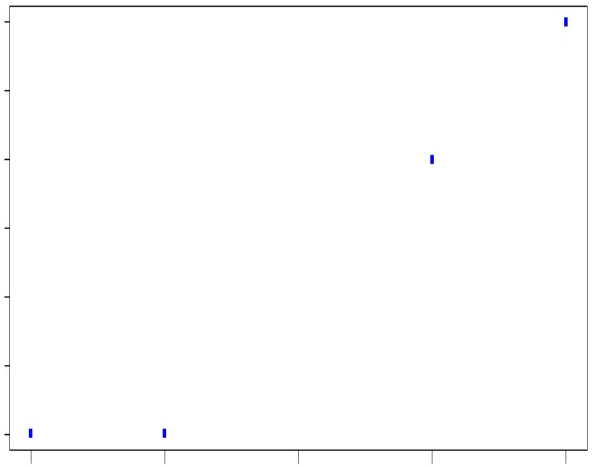
$$u_{ik} = \begin{cases} 1, & \text{jos } d(\mathbf{x}_k, \mathbf{v}_i) < \min_{j \neq i} d(\mathbf{x}_k, \mathbf{v}_j) \\ 0, & \text{muutoin.} \end{cases}$$

   - 2.2 Päivitä edustajavektorit laskemalla painotettu keskiarvo, missä jakomatriisin arvot otetaan huomioon

$$\mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}}.$$

   - 2.3 Jatka iterointia, kunnes kustannusfunktio Q ei muutu enää, muutokset ovat sallitun marginaalin rajoissa tai maksimimäärä iteraatiokierroksia on saavutettu.

Esimerkki K-Means klusteroinnin käytöstä



- Aineisto:

$\mathbf{x}_1$	0	0
$\mathbf{x}_2$	1	0
$\mathbf{x}_3$	3	2
$\mathbf{x}_4$	4	3

- Klustereiden lukumäärä: 2
- Tavoite: Klusteroi havainnot kahteen klusteriin.
- Olkoon klusterin 1 edustajavektori $\mathbf{v}_1^{(1)} = (0, 0)$.
- Olkoon klusterin 2 edustajavektori $\mathbf{v}_2^{(1)} = (1, 0)$.

Esimerkki jatkuu

- Lasketaan kunkin havainnon ja edustajavektorin välinen etäisyys $\Rightarrow$ muodostetaan etäisyysmatriisi D, missä alkio D_{ij} on i. havainnon euklidinen etäisyys j. klusterin edustajavektorista.
- Täten saadaan

$$D^{(1)} = \begin{bmatrix} 0 & 1 & \sqrt{13} & 5 \\ 1 & 0 & \sqrt{8} & \sqrt{18} \end{bmatrix}.$$

- Esimerkiksi havainnon $\mathbf{x}_3$ ja edustajavektorin $\mathbf{v}_2$ välinen etäisyys on $D_{23}^{(1)} = \sqrt{(3-1)^2 + (2-0)^2} = \sqrt{8}$.

Esimerkki jatkuu

- Katsotaan matriisin $D^{(1)}$ kultakin sarakkeelta pienin arvo ja asetetaan tätä arvoa vastaavalle kohdalle matriisissa $U^{(1)}$ arvo. Muut arvot kyseisellä sarakkeella ovat nollia.
- Jakomatriisi $U^{(1)}$:

$$U^{(1)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}.$$

- Lasketaan kustannusfunktion $Q^{(1)}$ arvo:

$$\begin{aligned} Q^{(1)} &= 1 \cdot 0^2 + 0 \cdot 1^2 + 0 \cdot \sqrt{13}^2 + 0 \cdot 5^2 \\ &\quad + 0 \cdot 1^2 + 1 \cdot 0^2 + 1 \cdot \sqrt{8}^2 + 1 \cdot \sqrt{18}^2 = 26 \end{aligned}$$

Esimerkki jatkuu

- Päivitetään edustajavektori klusterille 1:

$$\begin{aligned}\mathbf{v}_1^{(2)} &= \frac{\sum_{k=1}^{N} u_{1k}\mathbf{x}_k}{\sum_{k=1}^{N} u_{1k}} \\ &= \frac{1}{1+0+0+0}[1\cdot(0,0)+0\cdot(1,0)+0\cdot(3,2)+0\cdot(4,3)] \\ &= (0,0)\end{aligned}$$

- Päivitetään edustajavektori klusterille 2:

$$\begin{aligned}\mathbf{v}_2^{(2)} &= \frac{\sum_{k=1}^{N} u_{2k}\mathbf{x}_k}{\sum_{k=1}^{N} u_{2k}} \\ &= \frac{1}{0+1+1+1}[0\cdot(0,0)+1\cdot(1,0)+1\cdot(3,2)+1\cdot(4,3)] \\ &= (\tfrac{8}{3},\tfrac{5}{3})\end{aligned}$$

Esimerkki jatkuu

- Lasketaan havaintojen etäisyydet päivitettyjen edustajavektorien $\mathbf{v}_1^{(2)}$ ja $\mathbf{v}_2^{(2)}$ suhteen.
- Tällöin saadaan

$$D^{(2)} = \begin{bmatrix} 0 & 1 & \sqrt{13} & 5 \\ 3.14 & 2.36 & 0.47 & 1.89 \end{bmatrix}$$

ja

$$U^{(2)} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

- Kustannusfunktion $Q^{(2)}$ arvo on nyt:

$$\begin{aligned} Q^{(2)} &= 1 \cdot 0^2 + 1 \cdot 1^2 + 0 \cdot \sqrt{13}^2 + 0 \cdot 5^2 \\ &\quad + 0 \cdot 3.14^2 + 0 \cdot 2.36^2 + 1 \cdot 0.47^2 + 1 \cdot 1.89^2 = 4.79. \end{aligned}$$

Esimerkki jatkuu

- Päivitetään edustajavektori klusterille 1:

$$\begin{aligned}\mathbf{v}_1^{(3)} &= \frac{\sum_{k=1}^{N} u_{1k}\mathbf{x}_k}{\sum_{k=1}^{N} u_{1k}} \\ &= \frac{1}{1+1+0+0}[1\cdot(0,0)+1\cdot(1,0)+0\cdot(3,2)+0\cdot(4,3)] \\ &= (\tfrac{1}{2},0)\end{aligned}$$

- Päivitetään edustajavektori klusterille 2:

$$\begin{aligned}\mathbf{v}_2^{(3)} &= \frac{\sum_{k=1}^{N} u_{2k}\mathbf{x}_k}{\sum_{k=1}^{N} u_{2k}} \\ &= \frac{1}{0+0+1+1}[0\cdot(0,0)+0\cdot(1,0)+1\cdot(3,2)+1\cdot(4,3)] \\ &= (\tfrac{7}{2},\tfrac{5}{2})\end{aligned}$$

Esimerkki jatkuu

- Lasketaan havaintojen etäisyydet uusien edustajavektorien $\mathbf{v}_1^{(3)}$ ja $\mathbf{v}_2^{(3)}$ suhteen.
- Tällöin saadaan

$$D^{(3)} = \begin{bmatrix} 0.5 & 0.5 & 3.20 & 4.61 \\ 4.30 & 3.54 & 0.71 & 0.71 \end{bmatrix}$$

ja

$$U^{(3)} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}.$$

- Kustannusfunktion $Q^{(3)}$ arvo on nyt:

$$\begin{aligned} Q^{(3)} &= 1 \cdot 0.5^2 + 1 \cdot 0.5^2 + 0 \cdot 3.20^2 + 0 \cdot 4.61^2 \\ &\quad + 0 \cdot 4.30^2 + 0 \cdot 3.54^2 + 1 \cdot 0.71^2 + 1 \cdot 0.71^2 = 1.51. \end{aligned}$$

Esimerkki jatkuu

- Päivitetään edustajavektori klusterille 1:

$$\begin{aligned}\mathbf{v}_1^{(4)} &= \frac{\sum_{k=1}^{N} u_{1k}\mathbf{x}_k}{\sum_{k=1}^{N} u_{1k}} \\ &= \frac{1}{1+1+0+0}[1\cdot(0,0)+1\cdot(1,0)+0\cdot(3,2)+0\cdot(4,3)] \\ &= (\tfrac{1}{2},0)\end{aligned}$$

- Päivitetään edustajavektori klusterille 2:

$$\begin{aligned}\mathbf{v}_2^{(4)} &= \frac{\sum_{k=1}^{N} u_{2k}\mathbf{x}_k}{\sum_{k=1}^{N} u_{2k}} \\ &= \frac{1}{0+0+1+1}[0\cdot(0,0)+0\cdot(1,0)+1\cdot(3,2)+1\cdot(4,3)] \\ &= (\tfrac{7}{2},\tfrac{5}{2})\end{aligned}$$

Esimerkki loppuu

- Edustajavektorit ja jakomatriisi eivät muutu enää, mistä seuraa, että kustannusfunktion arvo ei vähene. On löydetty optimaaliset klusterit.
- Siis havainnot $\mathbf{x}_1$ ja $\mathbf{x}_2$ muodostavat klusterin 1 ja havainnot $\mathbf{x}_3$ ja $\mathbf{x}_4$ muodostavat klusterin 2.

R ja *K*-means

- *K*-Means löytyy R:n “stats” paketista
- Paketin lataus komennolla library(stats)
- Funktion nimi on kmeans()
- Lisätietoa saat komennolla help(“kmeans”)
- *K*-Means-algoritmin toteutus löytyy myös monesta muusta paketista.
- Seuraavaksi esimerkkejä *K*-Means-algoritmin käytöstä R:llä

K-Means klusteroinnin edut

- Menetelmä on yksinkertainen toteuttaa.
- Menetelmä sisältää vähän säädettäviä parametreja.
- Menetelmä on laskennallisesti kevyt.

K-Means klusteroinnin haittapuolet

- Miten määrittää optimaalinen K:n arvo? Käytännössä paras arvo selviää vain kokeilemalla eri K:n arvoja ja katsotaan, millä arvolla kustannusfunktion arvo on pienimmillään.
- Algoritmin tulos riippuu edustustajavektoreiden alustuksesta. Käytännössä algoritmia pitää toistaa useita kertoja eri edustajavektoreiden alkuvalinnoilla ja valita tuloksista paras. Tässä vaiheessa voi hyödyntää esimerkiksi rinnakkaislaskentaa, koska eri alkuvalinnat ovat riippumattomia toisistaan.
- Algoritmi ei osaa käsitellä hyvin poikkeavia havaintoja.
- Algoritmi ei osaa käsitellä puuttuvia arvoja.


K.J. Cios et al., *Data Mining: A Knowledge Discovery Approach*, Springer, 2007.


G. James et al.,*An Introduction to Statistical Learning - with Applications in R*, Springer, 2013.


M. Juhola, *Tiedonlouhinta kurssin luentomateriaali*, Tampereen yliopisto, 2014.

3. Hierarkkinen klusterointi

Henry Joutsijoki

Sisältö

Johdanto

- Osittavassa klusteroinnissa lähtökohtana oli se, että klusterien lukumäärä oli kiinnitetty. Tämä puolestaan aiheutti käytännön ongelmia:
  1. Miten määrittää oikea klusterien lukumäärä?
  2. Klustereiden edustajavektorien valinta
- Hierarkkisessa klusteroinnissa lähtökohta on varsin erilainen osittavaan klusterointiin verrattuna.
- Hierarkkiset klusterointimenetelmät yhdistävät havaintoja vähitellen tai jakavat klustereita osiin.
- Hierarkkisista klusterointimenetelmistä voidaan erottaa kaksi erilaista lähtökohtaa:
  1. Kokoavat hierarkkiset menetelmät (engl. agglomerative)
  2. Jakavat hierarkkiset menetelmät (engl. divisive)
- Keskitymme nyt ainoastaan kokoaviin hierarkkisiin menetelmiin.

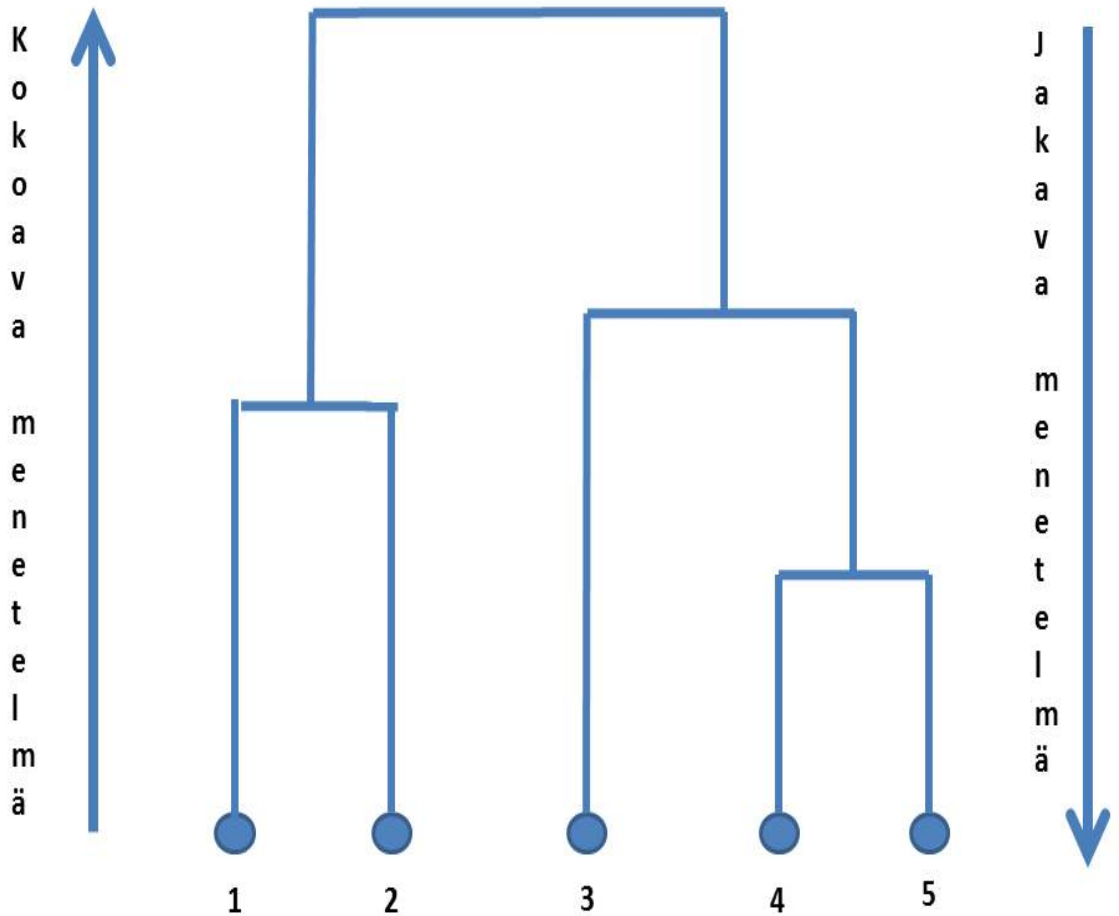
Johdanto

- Kokoavissa menetelmissä algoritmi yhdistää (sulauttaa) klustereita yhteen, kunnes loppujen lopuksi kaikki havainnot ovat samassa klusterissa.
- Jakavissa menetelmissä tehdään päinvastoin eli alussa meillä on kaikki havainnot samassa klusterissa. Alun jälkeen havaintoja jaetaan useampiin klustereihin ja viimeisessä vaiheessa kaikki havainnot ovat omassa klusterissaan.
- Kokoavat menetelmät ovat jakavia menetelmiä yksinkertaisempia, tärkeämpiä ja käytetympiä.

Hierarkkisen klusteroinnin ominaisuuksia

- Hierarkkisessa klusteroinnissa ei ole globaalia kustannusfunktiota.
- Globaalin kustannusfunktion asemesta on olemassa lokaaleja kustannusfunktioita, joita lasketaan lehtien pareille *puussa* siksi, että pystytään selvittämään, mikä klusterien pari on paras vaihtoehto kokoamiselle tai jaolle.
- Hierarkkiset menetelmät antavat kätevän graafisen esityksen, puun, jota kutsutaan *dendrogrammiksi*.
- Dendrogrammi pystytään määrittämään muuttujien lukumäärästä riippumatta, mikä antaa mahdollisuuden visualisoida moniulotteista dataa kaksiulotteisessa ympäristössä.

Esimerkki dendrogrammista



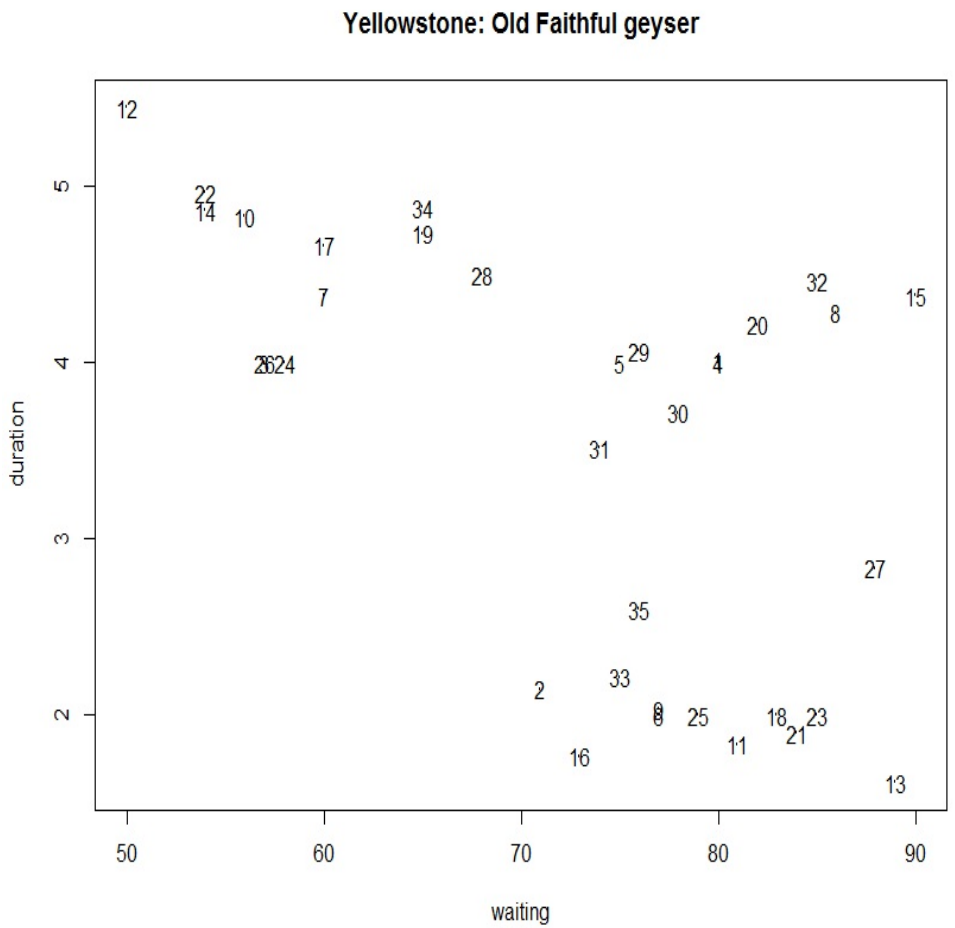
Dendrogrammin tulkitseminen

- Haarojen korkeudet (pystyakselin mittakaavassa) osoittavat, kuinka erilaisia kaksi havaintoa on keskenään.
- Havainnot, jotka yhdistyvät puun alaosassa ovat samankaltaisia keskenään, kun taas havainnot jotka yhdistyvät puun yläosassa eivät ole keskenään niin samankaltaisia.

Dendrogrammin tulkitseminen

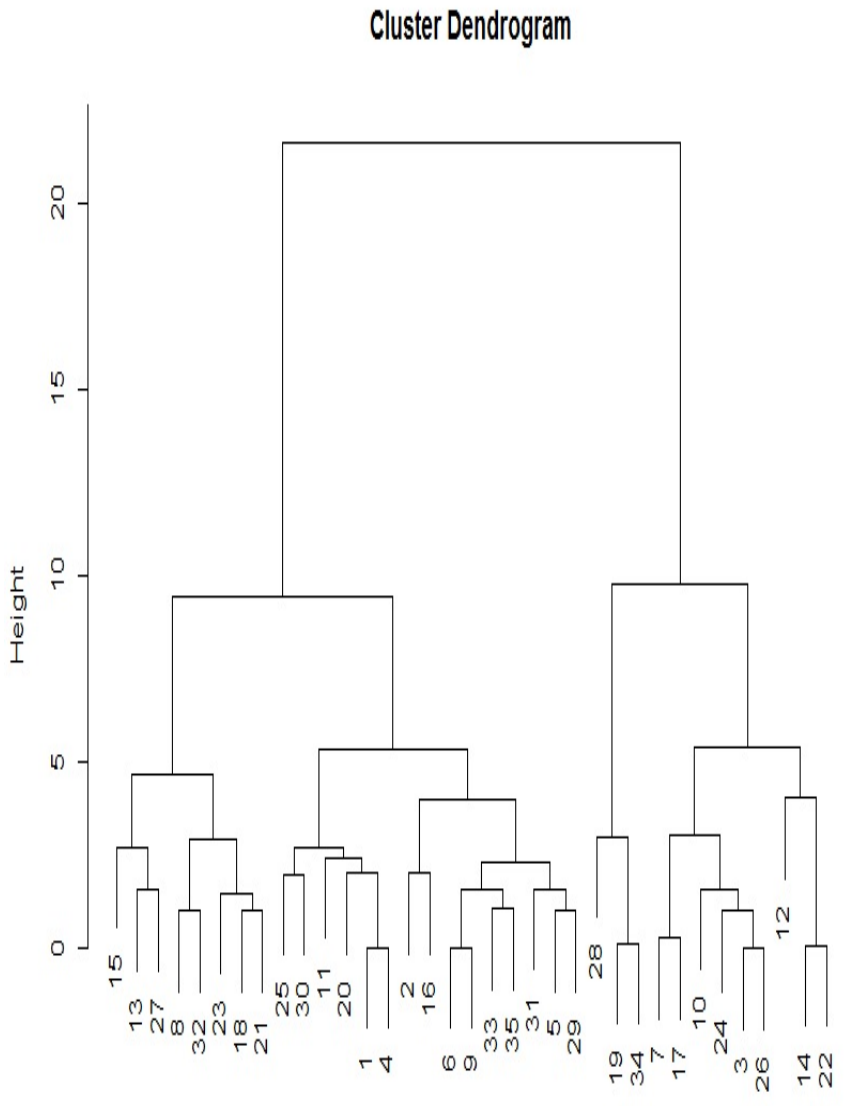
- Emme voi tehdä johtopäätöksiä kahden havainnon välille perustuen niiden sijaintiin vaaka-akselilla. Tämä johtuu matemaattisesta seikasta, että on olemassa 2^{N-1} (N on lehtien (havaintojen) lukumäärä) mahdollista eri järjestystä dendrogrammille.
- Järjestysten lukumäärä johtuu siitä, että jokaisessa $N-1$ pisteessä, missä yhdistämisiä tapahtuu, yhdistettävien haarojen paikat voidaan vaihtaa vaakasuunnassa vaikuttamatta dendrogrammin tulkintaan.

Toinen esimerkki dendrogrammista



- Data esittää Yellowstonen kansallispuiston Old Faithful -geysirin purkausten kestot suhteessa purkausten välisiin aikoihin.
- Aineisto sisältää kaiken kaikkiaan 299 havaintoa, mutta tarkasteluun niistä on otettu vain ensimmäiset 35 kappaletta.

Esimerkki jatkuu dendrogrammista



- Kuva esittää kokoavan yhdistämisen kahteen klusteriin kerrallaan. Haarojen korkeudet (pystyakselin mittakaavassa) esittävät kuinka erilaisia kaksi havaintoa ovat keskenään. Alussa pienimmän korkeuden omaavat pisteet (pisteet 1 ja 4) yhdistetään, joten ne ovat toisiaan lähimpänä.

Klusterien määrittäminen dendrogrammista

- Dendrogrammin perusteella voidaan määritellä klusterien määrä.
- Klusterien määrittäminen dendrogrammista tehdään vaakasuuntaisella leikkauksella valitulta korkeudelta.
- Leikkauksen alle jäävät havaintoja sisältävät erilliset joukot voidaan tulkita klustereiksi.

Klusterien määrittäminen dendrogrammista

- Uusia leikkauksia voidaan tehdä laskeuduttaessa dendrogrammissa alaspäin.
- Mahdolliset klusterien lukumäärät voivat olla 1:n (eli ei tehdä leikkauksia ollenkaan) ja N:n välillä (vastaa leikkausta korkeudelta 0, jolloin kaikki havainnot ovat omassa klusterisaan).
- Toisin sanoen dendrogrammin leikkaus vastaa samaa roolia kuin K:n arvo K-means-algoritmissa - se määrittelee saatavien klusterien lukumäärän.

Klusterien määrittäminen dendrogrammista käytännössä

- Käytännössä ihmiset usein katsovat dendrogrammia ja valitsevat silmämääräisesti järkevän lukumäärän klustereita perustuen yhdistämisien korkeuteen ja klustereiden haluttuun lukumäärää.
- Muistettava on kuitenkin se seikka, että aina oikean leikkauskohdan määrittäminen ei ole välttämättä selvää.
- Jokaisella datalla on omat erityispiirteensä ja yleispätevää sääntöä dendrogrammin leikkaamiselle ei ole olemassa.

Sisältö

Hierarkkisen klusteroinnin algoritmi

- Dendrogrammi saadaan määritettyä aineistosta hyvin helpolla algoritmilla.
- Ensimmäiseksi määritämme aineiston havaintoparien väliset etäisyydet käyttämällä valittua etäisyysmittaa.
- Yleensä käytämme euklidista etäisyyttä havaintojen välistä etäisyyttä määrittäessä, mutta voimme käyttää myös muita etäisyysmittoja kuten Manhattan-metriikkaa (korttelimetriikka), kosini- tai korrelaatiomittaa.
- Etäisyysmitan valinta on hyvin keskeinen kysymys, koska se vaikuttaa dendrogrammin muotoon.

Hierarkkisen klusteroinnin algoritmi

- Oikean etäisyysmitan valinta on sovellusriippuvainen ja tähän ei ole olemassa yhtä oikeata vastausta. Esimerkiksi tekstidokumenttien yhteydessä käytetään monesti kosinimittaa.
- Etäisyysmitan valinnan lisäksi on tärkeätä miettiä aineiston esiprosessointia. Skaalaammeko tai standardoimmeko aineiston muuttujat ennen kuin havaintojen väliset etäisyydet lasketaan?
- Tämä kysymys on täysin datariippuvainen, johon ei ole myöskään ole olemassa yhtä oikeata vastausta.
- Jos aineiston muuttujien arvot eroavat suuresti toisistaan, voi olla hyvä standardoida/skaalata muuttujat, jotta muuttujat saadaan tasapainoisempaan asemaan suhteessa toisiinsa.

Hierarkkisen klusteroinnin algoritmi

- Havaintoparien välisten etäisyyksien määrittämisen jälkeen algoritmi etenee iteratiivisesti.
- Aloitamme dendrogrammin pohjalta, missä kukin havainto (N kappaletta) on omana klusterinaan. Yhdistämme kaksi havaintoa, jotka ovat lähimpänä toisiaan, omaksi klusteriksi. Tämän jälkeen meillä on $N - 1$ klusteria jäljellä.
- Seuraavassa vaiheessa tilanteemme on erilainen, koska meillä $N - 2$ kappaletta yksittäisiä klustereita ja yksi klusteri, joka sisältää kaksi havaintoa. Miten määritämme klustereiden etäisyydet, jos yksi tai kumpikin yhdistettävistä klustereista sisältää enemmän kuin yhden havainnon?
- Vastaus: Meidän pitää laajentaa etäisyysmitan käsitettä koskemaan myös klustereiden välistä etäisyyttä!

Hierarkkisen klusteroinnin algoritmi

- Yleisimmät klustereiden etäisyysmitat (käsitellään tarkemmin myöhemmin):
  1. Lähimmän naapurin menetelmä (single link) klustereille C_i ja C_j

$$D_{sl}(C_i, C_j) = \min_{\mathbf{x},\mathbf{y}} \{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

  2. Kaukaisimman naapurin menetelmä (complete link) klustereille C_i ja C_j

$$D_{cl}(C_i, C_j) = \max_{\mathbf{x},\mathbf{y}} \{d(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in C_i, \mathbf{y} \in C_j\}$$

  3. Ryhmäkeskiarvomenetelmä (average link) klustereille C_i ja C_j

$$D_{al}(C_i, C_j) = \frac{1}{\#C_i \#C_j} \sum_{\mathbf{x} \in C_i} \sum_{\mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y}),$$

  missä $\#C_i$ tarkoittaa klusterin C_i kokoa.

Hierarkkisen klusteroinnin algoritmi

- Määriteltyämme klusterien välisen etäisyyden, voimme jatkaa dendrogrammin määrittämistä.
- Dendrogrammin muodostaminen jatkuu iteratiivisesti yhdistämällä kaksi lähintä klusteria, jolloin klusterien lukumäärä on $N - 2$. Seuraavassa vaiheessa klusterien lukumäärä on $N - 3$ jne, kunnes kaikki havainnot ovat yhdessä klusterissa ja dendrogrammi on valmis.

Hierarkkisen klusteroinnin algoritmi: Yhteenveto

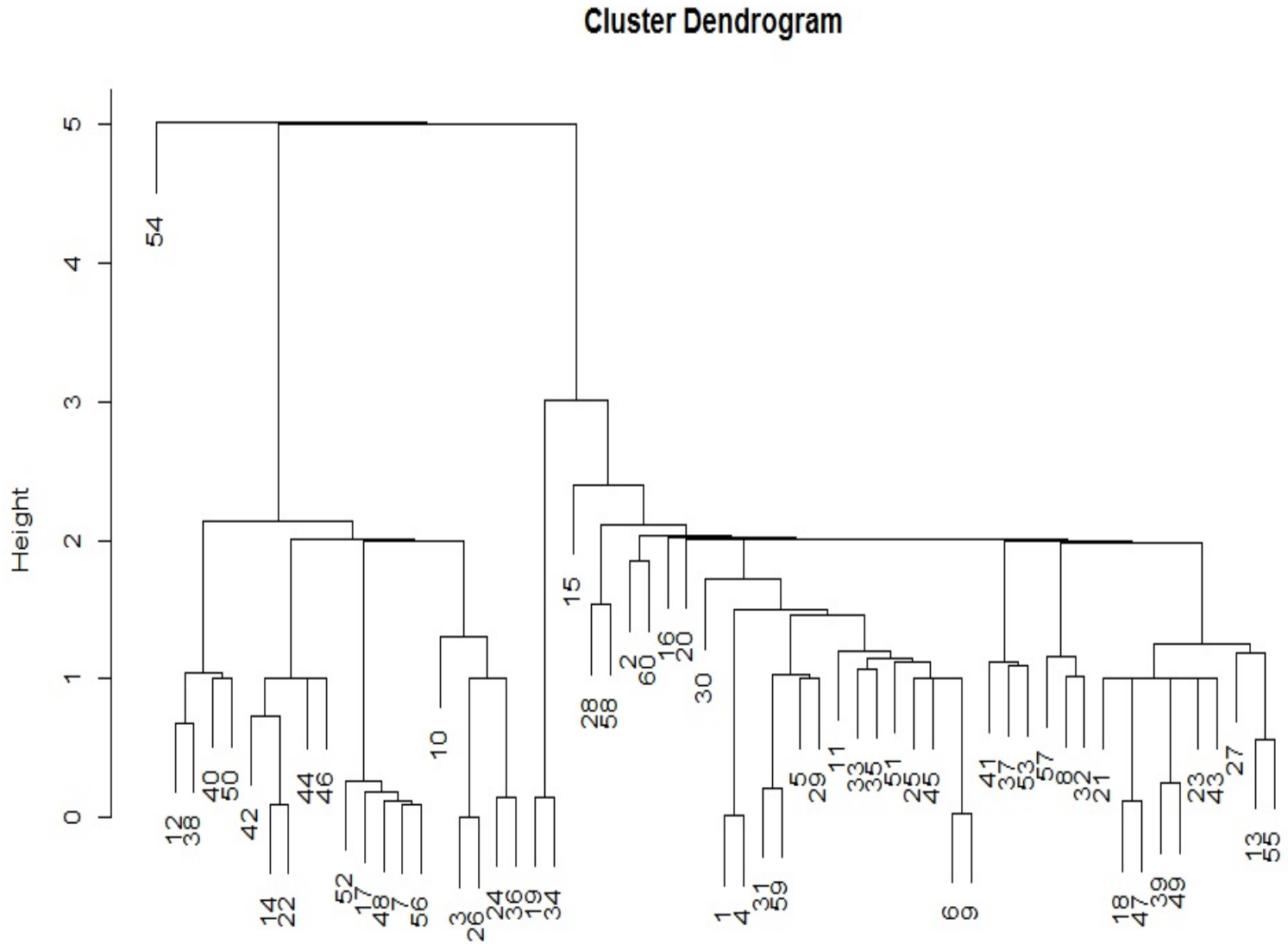
1. Olkoot aineistossa N kappaletta havaintoja. Määritä kaikkien havaintoparien väliset etäisyydet, joita on yhteensä $\frac{N(N-1)}{2}$ kappaletta.
2. Toista kaikilla $i = N, N-1, \ldots, 2$:
   - 2.1 Käy läpi kaikkien klusterien väliset etäisyydet (klustereita on i kappaletta) ja etsi klusteripari, jonka etäisyys on pienin. Yhdistä nämä klusterit. Näiden kahden klusterin välinen etäisyys osoittaa dendrogrammissa korkeuden, mihin klusterien yhdistäminen tulee sijoittaa.
   - 2.2 Määritä jäljelläolevien ($i-1$ kappaletta) klusterien väliset etäisyydet.

Sisältö

Lähimmän naapurin menetelmä

- Yksi varhaisimmista ja tärkeimmistä menetelmistä.
- Menetelmä on altis (hyvä tai huono asia, riippuen asiayhteydestä) “ketjuuntumisilmiölle”, jossa havaintojen pitkät nauhat määräytyvät samaan klusteriin.
- Menetelmä on myös herkkä poikkeaville arvoille.
- Menetelmällä on ominaisuus, että kun kaksi klusteriparia ovat yhtä läheisiä, ei ole merkitystä, kumpi yhdistetään ensin.

Esimerkki lähimmän naapurin menetelmästä

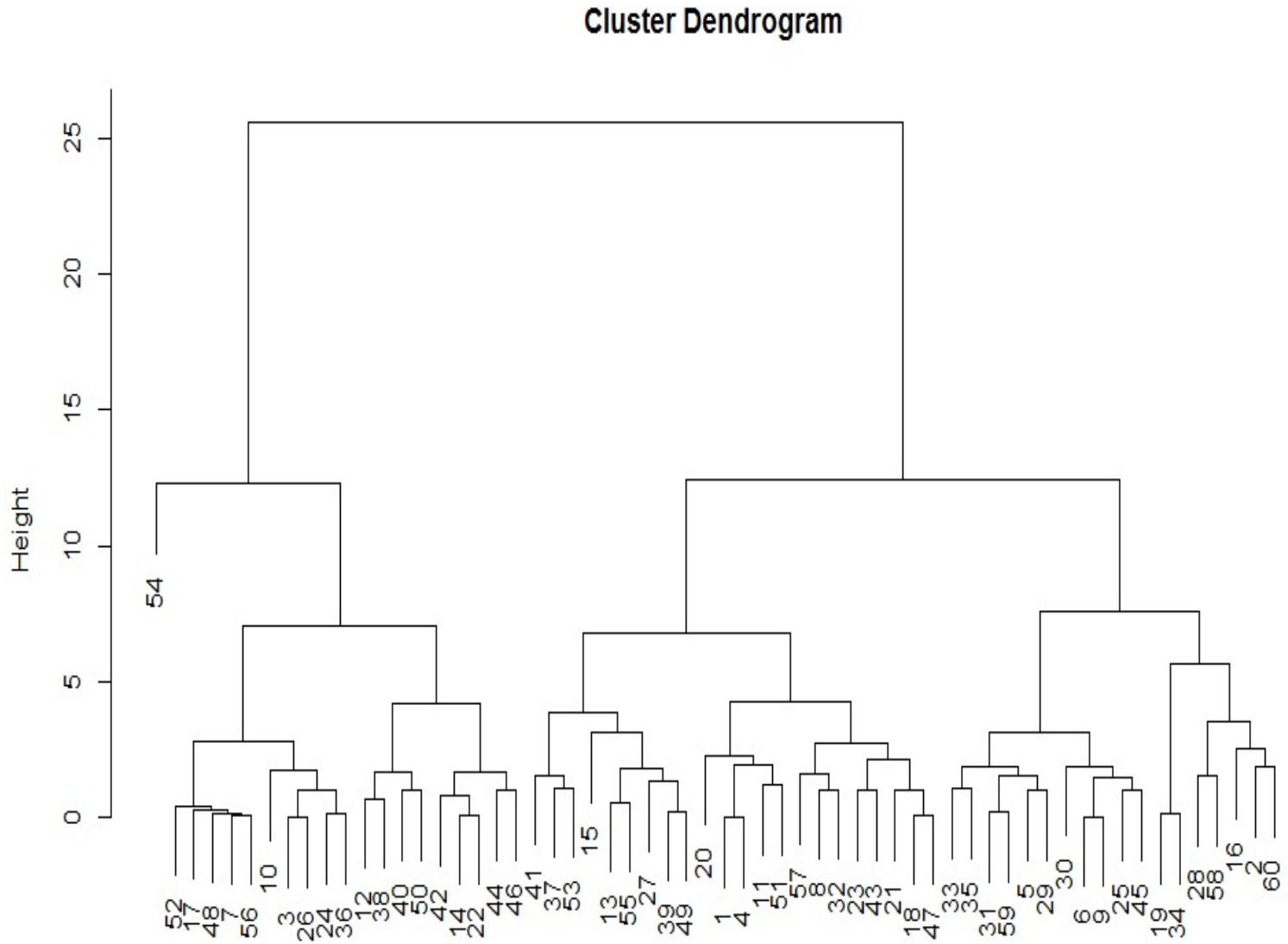


Kaukaisimman naapurin menetelmä ja ryhmäkeskiarvomenetelmä

- Menetelmät pyrkivät tasapainoisempiin klustereihin verrattuna lähimmän naapurin menetelmään.
- Tämä on monissa tapauksissa suotavampaa verrattuna lähimmän naapurin menetelmän ketjutukselle.

Esimerkki kaukaisimman naapurin menetelmästä

Esimerkki ryhmäkeskiarvomenetelmästä



Hierarkkinen klusterointi ja R

- Hierarkkinen klusterointi löytyy paketista “stats”. Paketin lataaminen tapahtuu komennolla library(stats).
- Hierarkkinen klusterointi tapahtuu käyttäen hclust() funktiota. Ennen kuin tätä funktiota voi käyttää, täytyy määrittää etäisyysmatriisi aineiston havainnoista käyttäen dist() funktiota, joka löytyy myös “stats” paketista.

Sisältö

Hierarkkisen klusteroinnin heikkoudet

- Aikakompleksisuus on $\mathcal{O}(N^2)$, koska ensimmäisessä iteraatiossa on paikallistettava lähin pari.
- Tilavaatimus on $\mathcal{O}(N^2)$, koska kaikkien pareittaisten etäisyyksien on oltava käytettävissä alussa.
- Suurilla aineistoilla hierarkkinen klusterointi on laskennallisesti raskas.
- Laajojen dendrogrammien tulkinta voi olla vaikeaa.
- Hierarkkisista klusterointimenetelmistä puuttuu adaptiivisuus. Klusterien yhdistämisiä ei voi myöhemmin enää peruuttaa.
- Käytetyt menetelmät ovat luonteeltaan ahneita. Aina yhdistetään lähimmät klusterit.

Hierarkkisen klusteroinnin edut

- Hierarkkinen klusterointi visualisoi datan luonnetta.
- Dendrogrammi rakentaminen ei ole riippuvainen muuttujien lukumäärästä. Dendrogrammi voidaan rakentaa, vaikka piirteitä/muuttujia olisi enemmän kuin 2 tai 3.
- Etäisyysmatriisia rakentaessa ensimmäistä kertaa voidaan havaintojen väliset etäisyydet määrittää käyttäen monenlaisia etäisyysmittoja.

 K.J. Cios et al., *Data Mining: A Knowledge Discovery Approach*, Springer, 2007.

 G. James et al.,*An Introduction to Statistical Learning - with Applications in R*, Springer, 2013.

 M. Juhola, *Tiedonlouhinta kurssin luentomateriaali*, Tampereen yliopisto, 2014.

4. Mallinvalinta ja -arviointi luokittelutehtävissä

Henry Joutsijoki

Sisältö

Ohjattu oppiminen

- Ohjatun oppimisen lähtökohtana on, että aineiston havaintojen luokat on etukäteen tiedossa. Siis kukin havainto on etukäteen sijoitettu ennaltamäärättyyn luokkaan.
- Käytännössä havaintojen luokan määrittäminen tapahtuu esimerkiksi sovellusalan asiantuntijan toimesta.
- Havaintojen luokkien määrittäminen on erittäin tärkeä tehtävä ja se on tehtävä hyvin huolellisesti, koska se vaikuttaa koneoppimisalgoritmien kykyyn ennustaa uusien havaintojen luokat. Huolellisesti tehty esiprosessointi muodostaa hyvän pohjan luokittelulle.

Ohjattu oppiminen

- Käytännössä olisi hyvä, jos toinen riippumaton asiantuntija samalta sovellusalalta tarkistaisi asiantuntijan tekemät luokitukset havaintojen suhteen, jotta aineistoon ei jäisi virheitä.
- Monesti toisen asiantuntijan mielipiteen kuuleminen on mahdotonta toteuttaa esimerkiksi ajan puutteen tai olemassaolevien rahallisten resurssien vähyyden vuoksi.
- Joskus myös kilpailuasetelma esimerkiksi teollisuudessa saattaa estää muiden asiantuntijoiden käytön.

Ohjattu oppiminen

- Ohjatussa oppimisessa lähtökohtana on rakentaa malli käyttäen olemassaolevaa aineistoa. Saatua mallia voidaan hyödyntää ennustamisessa.
- Luokittelussa perusideana on, että aineistosta rakennetun mallin perusteella pystytään luokittelemaan oikein uusia (mallin rakentamisessa käytetyn aineiston näkökulmasta) havaintoja mahdollisimman tarkasti oikeisiin luokkiin.

Sisältö

Lähtökohta

- Peruslähtökohta on, että käytössä oleva aineisto jaetaan satunnaisesti kahteen osaan, jotka ovat toisiinsa nähden täysin erilliset.
  1. Opetusjoukko
  2. Testijoukko
- Opetusjoukkoa käyttäen koneoppimisalgoritmi (luokittelumenetelmä) rakentaa mallin ja testijoukolla mitataan mallin yleistettävyyttä (opetetun mallin arviointi).
- Luokittelumenetelmä ennustaa opetusjoukon avulla rakennetun mallin perusteella ennusteet testijoukon havainnoille.
- Ennustetuista luokkatiedoista voidaan edelleen määrittää, kuinka hyvin luokittelu on onnistunut vertaamalla ennustettuja luokkatietoja oikeisiin luokkatietoihin.

Jako opetus- ja testijoukkoihin

- Aineiston jakaminen opetus- ja testijoukkoihin on yksinkertainen lähestymistapa, mutta sillä omat heikkoutensa.
- Ensimmäinen ongelma koskee suhdetta, miten aineisto jaetaan opetus- ja testijoukkoihin. Erityistapausta, missä aineisto jaetaan puoliksi, kutsutaan hold-out-menetelmäksi. Tarkoituksena olisi, että mallin rakentamiseen käytettäisiin suurempi osa aineistosta.
- Yleisiä opetus-testi-jakosuhteita ovat 70%-30%, 80%-20% ja 90%-10%.
- Käytännössä opetus- ja testijoukkoihin jakamisessa tulee ottaa huomioon aineiston luokkajakauma, koska vähintään opetusjoukossa pitää olla kaikkien luokkien havaintoja.

Jako opetus- ja testijoukkoihin

- Toinen ongelma tässä lähestymistavassa on, että testijoukolla saatavat tulokset ovat vahvasti riippuvaisia siitä, mitkä havainnot on satunnaisesti valittu opetus- ja testijoukkoihin.
- Jos teemme luokittelun vain yhdellä testi-/opetusjoukkojaolla, voimme saada vahingossa liian hyvän/huonon tuloksen suhteessa datan oikeaan luonteeseen.
- Tämän vuoksi käytännössä on tarpeellista toistaa luokittelu useammalla opetus- ja testijoukkojaolla (koko aineisto jaetaan siis useaan kertaan opetus- ja testijoukkoihin) ja laskea keskiarvo saaduista tuloksista.

Jako opetus- ja testijoukkoihin

- Monet luokittelumenetelmät sisältävät parametreja, joiden arvoja täytyy optimoida. Parametriarvot määräävät muodostettavan mallin ja parametriarvojen valinta vaikuttaa mallin yleistettävyyteen eli ennustamiskykyyn. Tarkoituksena on siis löytää optimaaliset parametriarvot aineiston suhteen. Tehdään siis **mallinvalintaa**.
- Jos luokittelumenetelmä sisältää muokattavia parametreja, ei jako opetus- ja testijoukkoihin ole riittävä, koska tällöin testijoukon tehtäväksi tulisi samanaikaisesti parametriarvojen optimointi sekä testihavaintojen luokitteleminen. Tarvitaan siis erillinen validointijoukko.
- Optimaalisten parametriarvojen etsiminen on puolestaan osa mallin oppimisprosessia ja mallinvalintaa, kun taas testijoukon tehtävänä on ainoastaan toimia mallin yleistettävyyden mittarina (**mallin arviointi**).

Jako opetus-, validointi- ja testijoukkoihin

- Yhteenvetona aineiston jakamisesta opetus-, validointi- ja testijoukkoihin voidaan todeta seuraavaa:
  1. Opetusjoukkoa käytetään mallin parametrien säätämiseen.
  2. Validointijoukolla testataan parametriarvojen järkevyyttä (mallinvalinta).
  3. Testijoukkoa käytetään mallin yleistettävyyden testaamiseen (mallin arviointi).

Jako opetus-, validointi- ja testijoukkoihin

- Käytännössä aineisto voidaan jakaa monessa eri suhteessa opetus-, validointi- ja testijoukkoihin. Yleisiä jakosuhteita ovat 70%-20%-10% ja 80%-10%-10%.
- On myös hyvä tehdä useampia opetus-,validointi-, testijakoja, jotta vältetään tilanne, missä yksi ainoa satunnaisesti muodostettu jako sattuu tuottamaan hyvän lopputuloksen.
- Tärkeätä käytännössä on muistaa käyttää samaa opetus-, validointi- ja testijoukkojakoa, kun testataan eri parametriarvoja ja eri luokittelumenetelmiä. Tällöin tulokset ovat keskenään vertailukelpoisia.
- Jos meillä on olemassa suuri aineisto, voimme valita opetus-,validointi- ja testijoukot ilman ongelmia. Aina näin ei kuitenkaan ole, jolloin tarvitaan erilaista lähestymistapaa.

Esimerkki opetus-, validointi- ja testijoukkoihin jakamisesta

	Luokka		
$\mathbf{x}_1$	1	1	3
$\mathbf{x}_2$	1	2	7
$\mathbf{x}_3$	1	0	2
$\mathbf{x}_4$	1	5	5
$\mathbf{x}_5$	1	4	0
$\mathbf{x}_6$	2	1	6
$\mathbf{x}_7$	2	7	7
$\mathbf{x}_8$	2	4	8
$\mathbf{x}_9$	2	0	0
$\mathbf{x}_{10}$	2	2	4

Taulukko: Esimerkki kahden luokan aineistosta.

- Opetusjoukko: $\{\mathbf{x}_1, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9\}$
- Validointijoukko: $\{\mathbf{x}_2, \mathbf{x}_{10}\}$
- Testijoukko: $\{\mathbf{x}_3, \mathbf{x}_6\}$

Ali- ja ylioppiminen

- Kaksi tärkeää käsitettä mallien opettamisen suhteen ovat ali- ja ylioppiminen.
- Alioppiminen tarkoittaa, että liian yksinkertainen malli ei sovi aineistoon hyvin. Tällöin myös mallin yleistämiskyky eli kyky ennustaa uusien havaintojen luokkia oikein on heikko.
- Ylioppimisessa opetettu malli on sovitettu kuvaamaan liian tarkasti opetusjoukon sisältämää informaatiota, jolloin mallin yleistämiskyky on heikko.
- Sopivan mallin tulisi suoriutua hyväksyttävästi sekä opetusjoukossa ja testijoukossa, kun määritetään virhe oikeiden luokkatietojen ja ennustettujen luokkatietojen välillä.

Sisältö

Tausta

- Jaettaessa aineisto satunnaisesti opetus- ja testijoukkoihin (tai opetus-, validointi- ja testijoukkoihin) yhtenä ongelmana oli se, että tulos on riippuvainen siitä, mitkä havainnot sattuvat olemaan opetus-/validointi-/testijoukossa.
- Jos käytössämme on N kappaletta havaintoja, voidaan ne jakaa opetus-/validointi-/testijoukkoihin hyvin monella tavalla. Käytännössä kaikkia mahdollisia kombinaatioita ei pystytä edes testaamaan.
- Näin ollen jokin havainto tai jotkut havainnot eivät välttämättä koskaan olisi opetus- tai testijoukossa.

Tausta

- Joskus tulokset eivät ole tyydyttäviä käytettäessä esim. hold-out-menetelmää. Tällöin täytyy käyttää laskennallisesti raskaampia menetelmiä.
- Yksi hyvin paljon käytetty menetelmä käytännössä on ristiinvalidointi (cross-validation), millä voidaan estää myös ylioppiminen.
- Ristiinvalidoinnilla estimoidaan mallin ennustamiskyvyn hyvyyttä.
- Ristiinvalidointia voidaan myös hyödyntää parametrien optimoinnissa eli mallinvalinnassa.
- Ristiinvalidointi muistuttaa aiemmin kuvattua perinteistä opetus-testijoukkoasetelmaa, mutta pyrkii selvittämään tämän lähestymistavan heikkouksia.

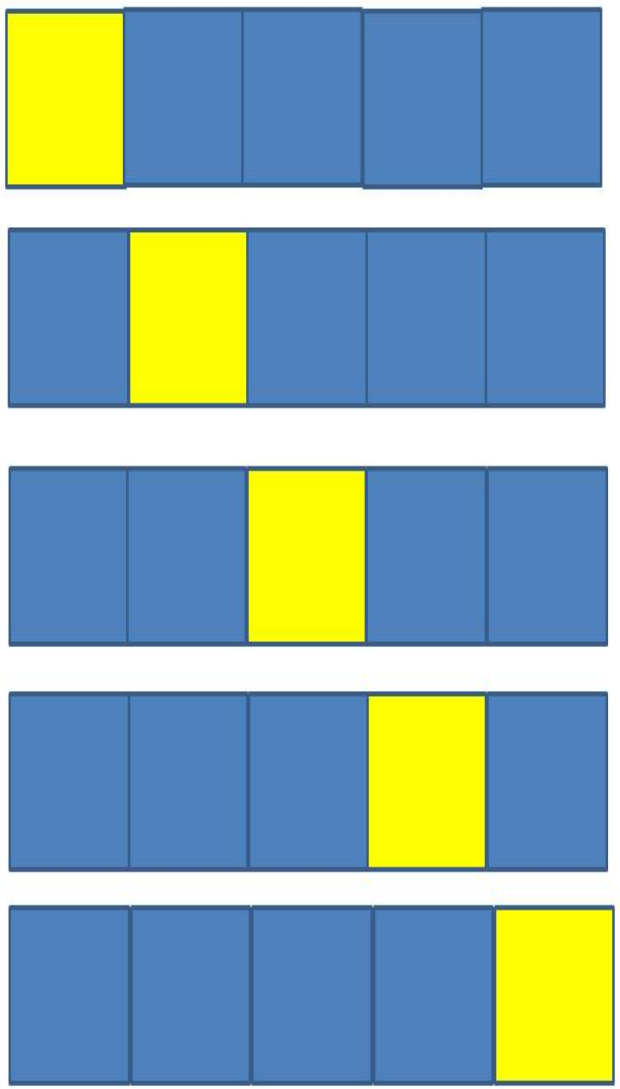
Ristiinvalidoinnin periaate

- Ristiinvalidoinnista löytyy monenlaisia variaatioita, mutta perusperiaate jokaisessa muunnelmassa on sama.
- Lähtökohta on k-kertainen ristiinvalidointi (k-fold cross-validation), missä koko aineisto jaetaan k:hon mahdollisimman yhtä suureen osaan satunnaisesti.
- Aineiston jakamisessa on otettava jälleen kerran huomioon aineiston luokkajakaumat. Arvon k valitseminen riippuu aineiston koosta ja siitä, kuinka suuria luokkia aineisto sisältää.
- Jaetut osat eivät ole päällekkäisiä toistensa kanssa.

Ristiinvalidoinnin periaate

- Käytettäessä k-kertaista ristiinvalidointia luokittelussa, suoritamme k kappaletta kierroksia, joissa kullakin kierroksella k. osa toimii testijoukkona ja loput $k-1$ osaa opetusjoukkona.
- Tällöin aineiston jokainen havainto tulee olemaan $k-1$ kertaa opetusjoukossa ja kerran testijoukossa.
- Ristiinvalidoinnissa tulee hyödynnettyä koko aineisto niin opetuksen kuin testaamisen näkökulmasta.

Havainnollistus ristiinvalidoinnista



- Havainnollistus 5-kertaisesta ristiinvalidoinnista. Keltainen laatikko tarkoittaa testijoukkoa ja sininen väri tarkoittaa opetusjoukkoa. Huomattavaa on, että testi- ja opetusjoukko vaihtuvat jokaisen kierroksen jälkeen.

Ristiinvalidoinnin periaate

- k-kertaisessa ristiinvalidoinnissa jokaisella kierroksella lasketaan testijoukosta ennustevirhe (tai luokittelutarkkuus) vertaamalla testijoukon oikeita luokkatietoja ja ennustettuja luokkatietoja keskenään.
- Lopputuloksena lasketaan keskiarvo kaikkien testijoukkojen ennustevirheistä (luokittelutarkkuuksista).
- Ristiinvalidoinnin yhteydessä keskeiseksi kysymykseksi tulee, kuinka moneen osaan aineisto pitää jakaa?
- Tähän ei ole yhtä ainoata oikeata vastausta, vaan jokaisen aineiston yhteydessä pitää päättää erikseen, mikä määrä jakoja olisi sopiva.
- Käytetyimmät vaihtoehdot ovat $k = 10$, $k = 5$ ja $k = N$ (leave-one-out), kun aineistossa on N kappaletta havaintoja.

Esimerkki 4-kertaisesta ristiinvalidoinnista

Havainto	*RV Jako*	*Luokka*	$Muuttuja_1$	$Muuttuja_2$
$\mathbf{x}_1$	1	1	1	3
$\mathbf{x}_2$	3	1	2	7
$\mathbf{x}_3$	2	1	0	2
$\mathbf{x}_4$	4	1	5	5
$\mathbf{x}_5$	1	1	4	0
$\mathbf{x}_6$	1	2	1	6
$\mathbf{x}_7$	4	2	7	7
$\mathbf{x}_8$	3	2	4	8
$\mathbf{x}_9$	2	2	0	0
$\mathbf{x}_{10}$	2	2	2	4

Taulukko: Aineiston toinen sarake näyttää ristiinvalidointijaon eli mihin osaan kukin havainto kuuluu ristiinvalidoinnissa.

Esimerkki 4-kertaisesta ristiinvalidoinnista jatkuu

- 1. kierros : testijoukko=$\{\mathbf{x}_1, \mathbf{x}_5, \mathbf{x}_6\}$
- 1. kierros : opetusjoukko=$\{\mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_7, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{10}\}$
- 2. kierros : testijoukko=$\{\mathbf{x}_3, \mathbf{x}_9, \mathbf{x}_{10}\}$
- 2. kierros : opetusjoukko=$\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_8\}$
- 3. kierros : testijoukko=$\{\mathbf{x}_2, \mathbf{x}_8\}$
- 3. kierros : opetusjoukko=$\{\mathbf{x}_1, \mathbf{x}_3, \mathbf{x}_4, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_7, \mathbf{x}_9, \mathbf{x}_{10}\}$
- 4. kierros : testijoukko=$\{\mathbf{x}_4, \mathbf{x}_7\}$
- 4. kierros : opetusjoukko=$\{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_5, \mathbf{x}_6, \mathbf{x}_8, \mathbf{x}_9, \mathbf{x}_{10}\}$

Ristiinvalidointi ja parametriarvot

- Lähtökohtana on, että ristiinvalidointia sovelletaan opetusjoukkoon. Ristiinvalidointiprosessi toistetaan (käyttäen samaa ristiinvalidointijakoa joka kerta) eri parametriarvoilla ja kullekin parametriarvolle lasketaan keskimääräinen luokittelutarkkuus.
- Parhaan luokittelutarkkuuden opetusjoukossa saava parametriarvo valitaan optimaaliseksi parametriarvoksi. Optimaalinen parametriarvo siis määrittää muodostettavan mallin.
- Yleisenä tapana on, että parhaan parametriarvon löytämisen jälkeen (eli mallinvalinnan jälkeen) luokittelumenetelmä opetetaan uudestaan käyttäen koko opetusjoukkoa ja optimaalisia parametriarvoja. Lopuksi opetetulla luokittelijalla luokitellaan testijoukon havainnot ja määritetään luokittelutarkkuus testijoukolle. Saatu luku on luokittelun lopputulos.

Ristiinvalidointi ja parametriarvot

- Ristiinvalidoinnin hyödyntäminen parametriarvojen optimoinnissa on siis vaihtoehtoinen tapa alussa mainitulle opetus-, validointi- ja testijoukkojaolle, missä eri parametriarvojen sopivuutta testattiin itsenäisellä validointijoukolla.
- Voimme siis käyttää yksinkertaisempaa jakoa opetus- ja testijoukkoihin ja soveltaa ristiinvalidointia opetusjoukkoon.

Ristiinvalidointi: yhteenveto

- Ristiinvalidointi on yleisesti käytetty tekniikka mallinvalinnassa, luokittelussa ja parametriarvojen optimoinnissa.
- Ristiinvalidointia käytetään erityisesti mallien tuottaman luokittelutarkkuuden arvioinnissa.
- Ristiinvalidointijako aineistolle tehdään satunnaisesti, mutta huomioiden että aineiston sisältämät luokat tulee olla edustettuna jokaisen ristiinvalidointikierroksen opetus- ja testijoukossa.
- Lopullinen luokittelutarkkuus lasketaan ristiinvalidoinnin yhteydessä käytettävien testijoukkojen luokittelutarkkuuksien keskiarvona.

Ristiinvalidointi: yhteenveto

- Ristiinvalidoinnissa käytettävien osioiden lukumäärä on riippuvainen aineiston koosta sekä luokkajakaumasta. Käytännössä täytyy miettiä huolellisesti, minkälaista ristiinvalidointia käyttää mihinkin aineistoon.
- Koska aineiston havainnot jaetaan ristiinvalidoinnissa satunnaisesti, eri jaot tuottavat eri lopputuloksia. Näin ollen hyvä käytäntö on suorittaa toistettu k-kertainen ristiinvalidointi.
- Esimerkiksi voidaan muodostaa 10 erilaista 10-kertaista ristiinvalidointijakoa ja toistaa luokittelu kaikilla näillä ristiinvalidointijaoilla. Lopputuloksena otetaan keskiarvo kaikista luokittelutarkkuuksista.

Ristiinvalidointi: yhteenveto

- Erikoistapauksena ristiinvalidoinnista on LOO-menetelmä.
- Toistettua k-kertaista ristiinvalidointia ei voi tehdä käyttäen leave-one-out eli LOO-menetelmää. LOO-menetelmän lopputulos on yksikäsitteinen, koska siinä testijoukon muodostaa yksi havainto ja kaikki aineiston havainnot ovat vuorollaan testijoukossa.

Sisältö

Johdanto

- Kun olemme löytäneet sopivan mallin, tulee meidän määrittää mallin ennustamiskyky (prediction performance) testijoukosta.
- Luokittelutehtävissä ennustamiskyky voidaan määrittää sekaannusmatriisin avulla.
- Sekaannusmatriisi voidaan määrittää myös mallinvalinnan yhteydessä.
- Sekaannusmatriisista pystymme määrittämään monenlaisia tunnuslukuja, joita käytetään luokittelun onnistumisen mittareina (tai vastaavasti parametrien hyvyyden mittareina mallinvalinnan yhteydessä).

Sekaannusmatriisi

- Sekaannusmatriisi on neliömatriisi, missä rivit edustavat oikeita luokkia ja sarakkeet ennustettuja luokkia. Huom! Tämä ei ole vakiintunut käytäntö, vaan joskus roolit voivat olla päinvastoin.
- Sekaannusmatriisin *i*. rivin ja *j*. sarakkeen luku tarkoittaa niitä luokan *i* havaintojen lukumäärää jotka luokittelumenetelmä on luokitellut luokkaan *j* kuuluvaksi.
- Sekaannusmatriisin diagonaalilla olevat lukuarvot kertovat kuhunkin luokkaan oikein luokittuneiden havaintojen lukumäärän. Diagonaalin ulkopuolella olevat alkiot kertovat väärin luokittuneiden havaintojen lukumäärän sekä mihin luokkaan havainnot on luokiteltu väärin.

Sekaannusmatriisin määrittäminen

Hav.	OL	EL
$\mathbf{x}_1$	1	1
$\mathbf{x}_2$	1	2
$\mathbf{x}_3$	1	2
$\mathbf{x}_4$	1	1
$\mathbf{x}_5$	1	1
$\mathbf{x}_6$	2	2
$\mathbf{x}_7$	2	2
$\mathbf{x}_8$	2	2
$\mathbf{x}_9$	2	1
$\mathbf{x}_{10}$	2	2

Taulukko: Ensimmäinen sarake ilmoittaa havainnon. Toinen sarake osoittaa havainnon oikean luokkatiedon ja viimeinen sarake ennustetun

	Luokka 1	Luokka 2
Luokka 1	3	2
Luokka 2	1	4

Taulukko: Rivit osoittavat oikeaa luokkatietoa ja sarakkeet ennustettua luokkatietoa.

Sekaannusmatriisi kahden luokan tapauksessa

- Kahden luokan sekaannusmatriisista (binääriluokittimen sekaannusmatriisi) voidaan määrittää useita tunnuslukuja.
- Yleinen muoto kahden luokan sekaannusmatriisille (rivit kuvaavat oikeita luokkia ja sarakkeet ennustettuja luokkia):

	Luokka A	Luokka B
Luokka A	TP	FN
Luokka B	FP	TN

Sekaannusmatriisin lyhenteet

- TP (true positive) tarkoittaa luokan A havaintojen lukumäärä, jotka on luokiteltu oikein.
- FN (false negative) tarkoittaa luokan A havaintojen lukumäärää, jotka on luokiteltu väärin.
- FP (false positive) tarkoittaa luokan B havaintojen lukumäärää, jotka on luokiteltu väärin.
- TN (true negative) tarkoittaa luokan B havaintojen lukumäärää, jotka on luokiteltu oikein.

Esimerkkejä sekaannusmatriisista lasketuista tunnusluvuista

- Sensitiivisyys (sensitivity, true positive rate): $\frac{TP}{TP+FN}$
- Spesifisyys (specificity): $\frac{TN}{TN+FP}$
- FPR (false positive rate): $\frac{FP}{FP+TN}$
- FNR (false negative rate): $\frac{FN}{TP+FN}$
- Tarkkuus (accuracy) $\frac{TP+TN}{TP+FN+FP+TN}$ kuvaa, kuinka hyvin kaiken kaikkiaan luokittelu on onnistunut.

Sekaannusmatriisi useamman luokan tapauksessa

- Sekaannusmatriisi voidaan määrittää myös useamman kuin kahden luokan tapauksessa. Yleisesti ottaen sekaannusmatriisi on kooltaan $M \times M$, kun luokkien lukumäärä on M.
- Monet kahden luokan sekaannusmatriisista laskettavat tunnusluvut eivät ole yleistettävissä useamman luokan tapaukseen tai niiden yleistäminen on hyvin vaikeata.
- Näin ollen käytännössä käytetään lähinnä tarkkuutta (sekaannusmatriisin diagonaalialkioiden summa jaettuna matriisin kaikkien alkioiden summalla) ja sensitiivisyyttä (voidaan määrittää joka luokalle erikseen).

5. K:n lähimmän naapurin menetelmä

Henry Joutsijoki

Tausta

- k:n lähimmän naapurin menetelmä (k-Nearest Neighbour, k-NN) on yksi käytetyimmistä ja varhaisimmista luokittelumenetelmistä.
- Ensimmäinen versio menetelmästä julkaistiin jo 1950-luvulla.
- k-NN-menetelmää voidaan käyttää niin luokittelussa kuin regressiossa. Lisäksi puuttuvien arvojen imputoinnissa k-NN-menetelmää voidaan käyttää.
- k-NN-menetelmän suosio perustuu sen yksinkertaisuuteen ja hyviin tuloksiin useissa eri sovelluksissa (esimerkiksi kuvien luokittelussa, bioinformatiikassa, lääketieteellisissä sovelluksissa)

Tausta

- k-NN-menetelmää käytetään yhä runsaasti ja se on aktiivisen tutkimuksen kohteena.
- k-NN-menetelmästä on kehitetty lukuisia erilaisia variaatioita.
- k-NN kuuluu etäisyyspohjaisten menetelmien joukkoon, missä hyödynnetään valittua etäisyysfunktiota.
- Luonteeltaan k-NN on lokaali luokittelumenetelmä, koska lopullinen luokittelu tapahtuu uutta havaintoa lähinnä olevien k:n opetusjoukon havainnon perusteella.
- Huolimatta lokaalista luonteestaan menetelmää käytettäessä joudutaan laskemaan uuden luokiteltavan havainnon etäisyydet kaikkiin opetusjoukon havaintoihin.

k-NN-menetelmän perusoletukset

- Oletetaan, että meillä on opetusjoukko $D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)\}$, missä $\mathbf{x}_i \in \mathbb{R}^n, i = 1, 2, \ldots, N$ ovat opetusjoukon havaintoja ja $y_i, i = 1, 2, \ldots, N$ on opetusjoukon havaintoa $\mathbf{x}_i$ vastaava luokkatieto.
- Oletetaan, että opetusjoukossa esiintyvien luokkien määrä on l. Näin ollen $y_i \in \{1, 2, \ldots, l\}$ kaikilla $i = 1, 2, \ldots, N$.
- Oletetaan lisäksi, että meillä on valittuna etäisyysfunktio d ja $k > 0, k \in \mathbb{N}$.
- Tavoite: Ennusta luokkatieto uudelle havainnolle $\mathbf{x} \in \mathbb{R}^n$.

k-NN-menetelmän algoritmi

1. Määritä havainnon **x** etäisyys jokaisesta opetusjoukon havainnosta käyttäen valittua etäisyysfunktiota.
2. Hae opetusaineistosta k lähintä havaintoa luokiteltavan havainnon **x** suhteen.
3. Määritä saatujen opetusjoukon havaintojen (k kappaletta) joukosta yleisin (edustetuin) luokka.
4. Aseta tämä luokka luokiteltavan havainnon ennustetuksi luokkatiedoksi.

Etäisyysfunktio

- K:n lähimmän naapurin menetelmässä tärkeänä käsitteenä on etäisyysfunktio.
- Etäisyysfunktion valinta voi vaikuttaa suuresti luokittelutarkkuuteen.
- Etäisyysfunktio voi olla metriikka, mutta monesti k-NN-menetelmän yhteydessä käytetään myös mittoja.
- Mitä tarkoitetaan mitalla ja metriikalla?

Metriikka

- Metriikka määritellään funktiona $d : X \times X \to [0, \infty)$, missä X on joukko ja $X \times X$ on joukkojen karteesinen tulo. Lisäksi $[0, \infty)$ on joukko, joka sisältää kaikki ei-negatiiviset reaaliluvut.
- Kahden joukon karteesisella tulolla tarkoitetaan järjestettyjen parien joukkoa $X \times Y = \{(a, b) \mid a \in X, b \in Y\}$.
- Esimerkki: $\{1, 2\} \times \{3, 4\} = \{(1, 3), (1, 4), (2, 3), (2, 4)\}$.
- Määritelmästä näemme myös, että funktio d eli etäisyysfunktio ei voi olla negatiivinen.

Metriikka

- Metriikan tulee täyttää neljä ehtoa kaikilla $\mathbf{x}, \mathbf{y}, \mathbf{z} \in X$:
  1. $d(\mathbf{x}, \mathbf{y}) \geq 0$
  2. $d(\mathbf{x}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{x} = \mathbf{y}$
  3. $d(\mathbf{x}, \mathbf{y}) = d(\mathbf{y}, \mathbf{x})$
  4. $d(\mathbf{x}, \mathbf{y}) \leq d(\mathbf{x}, \mathbf{z}) + d(\mathbf{z}, \mathbf{y})$
- Viimeistä ehtoa kutsutaan kolmioepäyhtälöksi.
- $k - NN$-menetelmässä voidaan käyttää myös etäisyysmittoja, joka on heikompi kuin metriikka. Etäisyysmitta ei täytä metriikan kaikkia ehtoja.
- Esimerkiksi kosinimitta ei toteuta kolmioepäyhtälöä.

Yleisimmät metriikat ja mitat

- Yleisimmin käytettyjä metriikoita ja mittoja:
  1. Chebychevin etäisyys (metriikka): $d(\mathbf{x}, \mathbf{y}) = \max_i\{|x_i - y_i|\}$, kun $i = 1, 2, \ldots, N$.
  2. Euklidinen etäisyys (metriikka): $d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N}(x_i - y_i)^2}$
  3. Manhattan-etäisyys (metriikka): $d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N}|x_i - y_i|$
  4. Minkowskin etäisyys (metriikka): $d(\mathbf{x}, \mathbf{y}) = \left(\sum_{i=1}^{N}|x_i - y_i|^q\right)^{\frac{1}{q}}$, kun $q \geq 1$
  5. Kosinimitta: $d(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|}$
  6. Hamming-etäisyys (vain kategorisille muuttujille): Määritetään niiden piirteiden lukumäärä, missä $x_i \neq y_i$.

Miten valita oikea etäisyysfunktio?

- Yleistä sääntöä optimaalisen etäisyysfunktion valitsemiseen ei ole.
- Monesti perinteinen euklidinen etäisyys toimii hyvin, mutta se ei ole yleispätevä sääntö.
- Optimaalisen etäisyysfunktion löytäminen käy käytännössä yrityksen ja erehdyksen kautta.
- Jos on mahdollista saada sovellusalan asiantuntijalta mielipide asiasta, kannattaa se käyttää, koska kullakin sovellusalalla on omat erityispiirteensä, joiden tietäminen voi helpottaa etäisyysfunktion valintaa.
- Perussääntö on, että mahdollisimman montaa erilaista etäisyysfunktiota kannattaa kokeilla käytännössä. Näin saadaan empiiristä vahvistusta omalle lopulliselle valinnalle.

Miten valita oikea k:n arvo?

- Myöskään tähän kysymykseen ei ole olemassa yhtä oikeata sääntöä.
- Jokainen aineisto on oma tapauksensa ja optimaalinen k:n arvo täytyy tutkia erikseen.
- Standardi lähestymistapa on käyttää ristiinvalidointia optimaalisen k: estimoimisessa. Huom! Ristiinvalidointia sovelletaan opetusjoukkoon. Se, mitä ristiinvalidointitekniikkaa käytetään, on riippuvainen aineiston koosta.
- Jos kyseessä on kaksiluokkainen luokittelutehtävä, kannattaa valita pariton k:n arvo.

Miten ratkaista tasapelitilanne?

- k-NN-menetelmän yhteydessä voi ilmetä tasapelitilanteita uuden havainnon luokkaa määrittäessä. Tällöin täytyy soveltaa jotakin ratkaisumallia.
- Yleisiä ratkaisumalleja:
  1. Satunnainen valinta
  2. Valitaan se luokka, jonka havainto opetusjoukossa on lähimpänä luokiteltavaa havaintoa.
  3. Valitaan se luokka, jolla on enemmän havaintoja opetusjoukossa.

Esimerkki 1 k-NN-menetelmän käytöstä

Havainto	Y	X_1	X_2
$\mathbf{x}_1$	1	1	3
$\mathbf{x}_2$	1	2	7
$\mathbf{x}_3$	1	5	5
$\mathbf{x}_4$	1	4	0
$\mathbf{x}_5$	2	1	6
$\mathbf{x}_6$	2	7	7
$\mathbf{x}_7$	2	4	8
$\mathbf{x}_8$	2	2	4

Taulukko: Toinen sarake kuvaa havaintojen luokkatiedon.

- Luokkien määrä: 2
- Piirteet: X_1 ja X_2
- Luokkatieto: Y
- Opetusjoukon koko: 8 havaintoa.
- Luokittelija: k-NN
- Etäisyysfunktio: Euklidinen etäisyys
- Tavoite: Ennusta havainnolle $\mathbf{x} = [5\ 1]$ luokkatieto

Luokkatiedon määrittäminen havainnolle $\mathbf{x} = [5\ 1]$ (euklidinen etäisyys)

	Y	X_1	X_2
$\mathbf{x}_1$	1	1	3
$\mathbf{x}_2$	1	2	7
$\mathbf{x}_3$	1	5	5
$\mathbf{x}_4$	1	4	0
$\mathbf{x}_5$	2	1	6
$\mathbf{x}_6$	2	7	7
$\mathbf{x}_7$	2	4	8
$\mathbf{x}_8$	2	2	4

Etäisyys	Järjestys
$d(\mathbf{x}, \mathbf{x}_1) = \sqrt{(5-1)^2 + (1-3)^2} = \sqrt{20}$	4
$d(\mathbf{x}, \mathbf{x}_2) = \sqrt{(5-2)^2 + (1-7)^2} = \sqrt{45}$	7
$d(\mathbf{x}, \mathbf{x}_3) = \sqrt{(5-5)^2 + (1-5)^2} = 4$	2
$d(\mathbf{x}, \mathbf{x}_4) = \sqrt{(5-4)^2 + (1-0)^2} = \sqrt{2}$	1
$d(\mathbf{x}, \mathbf{x}_5) = \sqrt{(5-1)^2 + (1-6)^2} = \sqrt{41}$	6
$d(\mathbf{x}, \mathbf{x}_6) = \sqrt{(5-7)^2 + (1-7)^2} = \sqrt{40}$	5
$d(\mathbf{x}, \mathbf{x}_7) = \sqrt{(5-4)^2 + (1-8)^2} = \sqrt{50}$	8
$d(\mathbf{x}, \mathbf{x}_8) = \sqrt{(5-2)^2 + (1-4)^2} = \sqrt{18}$	3

- Miten käy havainnon $\mathbf{x}$ luokittelun, kun $k = 1$,$k = 2$, $k = 3$ tai $k = 5$?

Esimerkki 2 k-NN:n käytöstä

Havainto	Y	X_1	X_2
$\mathbf{x}_1$	1	1	3
$\mathbf{x}_2$	1	2	7
$\mathbf{x}_3$	1	5	5
$\mathbf{x}_4$	1	4	0
$\mathbf{x}_5$	2	1	6
$\mathbf{x}_6$	2	7	7
$\mathbf{x}_7$	2	4	8
$\mathbf{x}_8$	2	2	4

Taulukko: Toinen sarake osoittaa opetusjoukon havaintojen luokat.

- Luokkien määrä: 2
- Piirteet: X_1 ja X_2
- Luokkatieto: Y
- Opetusjoukon koko: 8 havaintoa.
- Luokittelija: k-NN
- Etäisyysfunktio: Manhattan metriikka
- Tavoite: Ennusta havainnolle $\mathbf{x} = [5\ 1]$ luokkatieto.

Luokkatiedon määrittäminen havainnolle $\mathbf{x} = [5\ 1]$ (Manhattan-metriikka)

	Y	X_1	X_2
$\mathbf{x}_1$	1	1	3
$\mathbf{x}_2$	1	2	7
$\mathbf{x}_3$	1	5	5
$\mathbf{x}_4$	1	4	0
$\mathbf{x}_5$	2	1	6
$\mathbf{x}_6$	2	7	7
$\mathbf{x}_7$	2	4	8
$\mathbf{x}_8$	2	2	4

Etäisyys	Järjestys
$d(\mathbf{x}, \mathbf{x}_1) = \lvert 5-1 \rvert + \lvert 1-3 \rvert = 6$	3
$d(\mathbf{x}, \mathbf{x}_2) = \lvert 5-2 \rvert + \lvert 1-7 \rvert = 9$	7
$d(\mathbf{x}, \mathbf{x}_3) = \lvert 5-5 \rvert + \lvert 1-5 \rvert = 4$	2
$d(\mathbf{x}, \mathbf{x}_4) = \lvert 5-4 \rvert + \lvert 1-0 \rvert = 2$	1
$d(\mathbf{x}, \mathbf{x}_5) = \lvert 5-1 \rvert + \lvert 1-6 \rvert = 9$	7
$d(\mathbf{x}, \mathbf{x}_6) = \lvert 5-7 \rvert + \lvert 1-7 \rvert = 8$	5
$d(\mathbf{x}, \mathbf{x}_7) = \lvert 5-4 \rvert + \lvert 1-8 \rvert = 8$	5
$d(\mathbf{x}, \mathbf{x}_8) = \lvert 5-2 \rvert + \lvert 1-4 \rvert = 6$	3

- Miten käy havainnon $\mathbf{x}$ luokittelun, kun $k = 2$ tai $k = 3$?

k-NN: edut, haitat ja käytäntö

- k-NN on helppo toteuttaa.
- Oikean etäisyysfunktion ja k arvon löytäminen saattaa olla käytännössä työläs tehtävä.
- Puuttuvien tietojen käsittely on yksinkertaisin mahdollinen; käsitellään vain tunnettujen muuttujien aliavaruutta.
- k-NN-menetelmän heikkous on, ettei se muodosta mallia vaan luottaa “laiskana” menetelmänä opetusjoukkoon.

k-NN: edut, haitat ja käytäntö

- Opetusjoukon ollessa suuri k lähimmän naapurin etsintä voi olla paljon aikaa vievä toimenpide.
- Optimaalisen k arvon etsimisessä voidaan hyödyntää rinnakkaislaskentaa, koska testattavat k:n arvot ovat toisistaan riippumattomia. Rinnakkaislaskennan hyödyntäminen puolestaan tehostaa laskentaa.

k-NN ja R

- k-NN luokittelija löytyy ainakin seuraavista R:n paketeista: “class” ja “kknn” (kernel-versio k-NN luokittelijasta)
- työkaluja ristiinvalidointiin löytyy ainakin paketeista “cvTools” ja “caret”.


K.J. Cios et al., *Data Mining: A Knowledge Discovery Approach*, Springer, 2007.


M. Juhola, *Tiedonlouhinta kurssin luentomateriaali*, Tampereen yliopisto, 2014.