Luonnollisen kielen tilastollinen käsittely

T-61.281 (3 ov) L

Kevät 2003

Luennot: **Timo Honkela, Krista Lagus**

Laskuharjoitukset: **Vesa Siivola**

Luentokalvot: Krista Lagus ja Timo Honkela

13. Klusterointi

(Lähteet: kirjan luku 14 ja kurssin T-61.231 'Hahmontunnistuksen perusteet' kalvot)

Ohjaamattomassa oppimisessa tyypillinen tehtävä on klusterointi, jossa pyritää muodostamaan havainnoista rykelmiä (klustereita), joiden sisällä havainnot ovat keskenään jollain lailla samankaltaisia ja klustereiden välillä erikaltaisia.

Havaintoihin ei liity mitään luokkatietoa. Myöskään klustereiden lukumäärää ei välttämättä tiedetä ennalta.

Klusterointia voidaan käyttää myös silloin kun jotain luokkatietoa on annettu, mutta halutaan selvittää tarkemmin luokkien sisäinen rakenne.

Klusterointialgoritmien ryhmittely

Klusterointialgoritmeja voidaan jakaa ryhmiin eri tavoin, mm.

- parametriset (EM, mikstuurimallit) vs. epäparametriset
- hierarkkiset (esim. single-link clustering) vs. litteät (esim. K-means)
- kovan vs. pehmeän klusteroinnin tekevät menetelmät

Parametriset algoritmit yrittävät samanaikaisesti mallittaa näytteiden kuuluminen klustereihin sekä klusterien todennäköisyysjakauman parametrit piirreavaruudessa.

Epäparametriset algoritmit jakavat havainnot eri luokkia vastaaviksi osajoukoik si, luokkien tn-jakaumia ei estimoida.

Ensimmäiset ovat tilastollisesti perustellumpia, kun taas jälkimmäiset voidaan usein toteuttaa yksinkertaisesti ja tehokkaasti.

Klusterointiongelman ratkaisun vaiheet

- Piirteiden valinta (esikäsittely & normalisointi!)
- Samankaltaisuus/erilaisuusmitan valinta havaintoparien, havaintojen ja klustereiden, sekä klusteriparien välille (piirteiden samanarvoinen kohtelu!)
- Klusterikriteerin valinta
- Klusterointialgoritmin valinta
- Tulosten validointi erilaisten testien avulla
- Tulosten tulkinta

Huom! Saatu ratkaisu on subjektiivinen, koska sama havaintojoukko voidaan jakaa erilaisiin klustereihin riippuen em. valinnoista. Yleensä ratkaisun tulkinnan ja sen hyvyyden arvioi sovellusalan asiantuntija.

13.1 Erilaisia klusterin määritelmiä

- *Luonnolliset* klusterit ('natural clusters'): piirreavaruuden alueita, joissa havainnot ovat suhteellisen tiheässä ja joiden välissä havainnot ovat suhteellisen harvassa.
- Yksikäsitteiset eli *kovat* klusterit ('hard','crisp'): näyte kuuluu kokonaan yhteen luokkaan. NLP:n kannalta tämä on epätoivottavaa johtuen esim. monitulkintaisuuden yleisyydestä.
- Pehmeät eli *sumeat* klusterit: havainnon kuulumisasteet eri klustereihin ilmaistaan jäsenyysfunktioiden ('membership functions') avulla.
- *Todennäköisyyksiin* perustuvat klusterit: havainto $\mathbf{x}$ kuuluu klusteriin C_k, jonka *a posteriori* tn on korkein eli $P(C_k|\mathbf{x}) \geq P(C_i|\mathbf{x})\ \forall i = 1, \ldots, m$
- *Disjunktiivisessa* klusterimallissa näyte voi samanaikaisesti kuulua aidosti moneen luokkaan. Kuitenkin tällöin joudutaan määrittelemään lisäksi kriteerit joista voidaan päätellä, milloin joku näyte kuuluu moneen eri klusteriin, milloin vain yhteen.

13.2 Hierarkkinen klusterointi

Hierarkkiset klusterointialgoritmit muodostavat havainnoille klusterointiratkaisujen sarjan, joilla on selkeä sisäkkäinen ('nested') rakenne.

Algoritmit voidaan jakaa kahteen alaluokkaan: kasaaviin ('agglomerative') ja pilkkoviin ('divisive') algoritmeihin.

Kasaavat algoritmit muodostavat klusterointiratkaisujen sarjan yhdistelemällä klustereita. Algoritmi lähtee liikkeelle tilanteesta, jossa jokainen havainto vastaa yhtä klusteria ($m = N$). Algoritmin edetessä klustereiden lkm pienenee, kunnes kaikki havainnot kuuluvat yhteen klusteriin ($m = 1$).

Pilkkovat algoritmit toimivat päinvastoin. Aluksi kaikki havainnot kuuluvat yhteen klusteriin ($m = 1$) ja algoritmin edetessä klustereita jaetaan kahtia, kunnes jokaisessa klusterissa on tasan yksi havainto ($m = N$).

Hierarkkisen klusteroinnin tulokset voidaan esittää *dendrogrammina* jossa samaan klusteriin kuuluvat havainnot on kytketty viivalla tietyllä korkeudella.

Hierarkkisten kasaavien klusterointialgoritmien kanssa sovellettavia samankaltaisuusmittoja

- *Single link* -klusterointi: Samankaltaisuus lasketaan *kahden samankaltaisimman* yksilön välillä.
  Seuraus: klustereilla hyvä paikallinen koherenssi, mutta klusterit voivat olla repaleisia ja monimuotoisia. Kompleksisuus $\mathcal{O}(n^2)$.

- *Complete link* -klusterointi: Samankaltaisuus lasketaan klusterin *kahden erikaltaisimman* yksilön välillä.
  Seuraus: klusterit ovat globaalisti kiinteämpiä. Kuitenkin laskennallisesti raskaampi $\mathcal{O}(n^3)$

- *Group average* -klusterointi: Samankaltaisuus lasketaan *keskimääräisenä samankaltaisuutena* ryhmän jäsenten välillä. Kompromissi edellisten väliltä ja laskennallisesti kohtuullisen tehokas $\mathcal{O}(n^2)$, jos havainnot esitetään normalisoituina vektoreina ja käytetään pistetuloetäisyyttä.

13.3 Ei-hierarkkinen klusterointi

Useat ei-hierarkkisista algoritmeista iteroivat klusterointia siten, että se vähitell paranee jonkin optimoitavan funktion tai mitan mielessä.

Joissakin algoritmeissa asetetaan klusterien määrään vaikuttava parametri, esim. havainnon suurin sallittu etäisyys klusterin painopisteestä.

Algoritmien perusperiaate:

1. Alustetaan klusterointi jakamalla havainnot ryhmiin jollain tavalla, esim. satunnaisesti.

2. Iteroidaan: Uudelleenallokoidaan datoja siten, että kustannusfunktion arvo pienenee.

3. Lopetetaan, esim. kun kustannusfunktion arvo alkaa kasvaa, tai kun parantuminen saavuttaa melko tasaisen alueen.

K-means -algoritmi

K-means tekee kovan klusteroinnin. Klusterit esitetään prototyyppivektorien avulla siten, että prototyyppivektori on klusteriin kuuluvien havaintojen massakeskipiste (centroid).

Alustus voidaan tehdä satunnaisesti esim. valitsemalla prototyyppien arvoiksi satunnaisesti valittu osajoukko havainnoista.

Havainto $\mathbf{x}_i$ sijoitetaan siihen klusteriin j, jonka prototyypin $\mathbf{c}_j$ etäisyys havainnosta on pienin ($\arg\min_j d(\mathbf{x}_i, \mathbf{c}_j)$).

Algoritmi: Vuorottele vaiheita 1 ja 2:
Vaihe 1: Uudelleenallokoi kaikki havainnot klustereihin.
Vaihe 2: Laske klusterien keskipisteet
Lopetusehto: lopeta, kun klusterointi ei enää muutu tai kun on ajettu pyydetty määrä iteraatioita

K-means -algoritmi, kommentteja

Etäisyysfunktio d on tavallisesti Euklidinen etäisyys (L_2-normi) mutta myös muita metriikoita voidaan käyttää.

Jos yhtäsuuria minimietäisyyksiä esiintyy, nämä voidaan ratkaista satunnaisesti (mistä voi kuitenkin seurata ettei algoritmi konvergoidu, jos lopetusehtona on klusteroinnin muuttumattomuus).

Jos iteraatioita käydään läpi vakiomäärä, algoritmin kompleksisuus on $\mathcal{O}(n)$.

Algoritmi on herkkä alustukselle, joten on suotavaa ajaa useilla eri satunnaisalustuksilla, tai alustaa muuten 'fiksusti'.

Eräs tapa tehdä pehmeä klusterointi on toteuttaa se Gaussin mikstuurimallia käyttäen, alustaa klusterit K-meansilla ja optimoida ne EM-algoritmilla.

Klusterien määrän määrittäminen

Useiden klusteroinnin optimointikriteerien arvolla on taipumus parantua (opetusdatalla mitattuna), kun klusterien määrä nousee. Mitat ovat siis herkkiä ylioppimiselle, eivätkä sovellu mallin rakenteen optimointiin.

Klusterien optimaalisesta määrästä voi olla sovelluskohtaista prioritietoa, jolloin määrä voidaan valita etukäteen. Optimaalinen määrä voi löytyä myös klusterien sovelluksen tarjoaman evaluointitiedon kautta.

Matemaattisesti hyvin perusteltuja tapoja klusterien lukumäärän optimointiin ilman aineiston ulkopuolista hyvyyskriteeriä tarjoavat MDL-periaate ja Bayeslainen mallinnus, jossa sovelletaan variaatioanalyysiä. Molemmat näistä mittaavat yhtäaikaa sekä datan todennäköisyttä että mallin kompleksisuutta eli ne voidaan laskea myös opetusdatalla.

Klusterien sopiva lukumäärä voidaan myös estimoida mittaamalla hyvyyttä erillisellä validointidatajoukolla: Valitaan klusterien määrä jolla validointijoukon hyvyysarvo on maksimi. (ks. kirjan kuva 16.4).

13.4 Klusteroinnin sovelluksia

Sovellustyyppejä yleensä:

- Exploratory data analysis: suuren ja kompleksisen aineiston havainnollistaminen ja tuki hypoteesien muodostamiselle
- kandidaattiluokittelun muodostaminen, jos luokkatietoa ei ole
- yleistäminen, kun dataa on liian vähän per yksittäinen näytetyyppi (esim. kielimalleissa).
- Datan tehokas koodaus ja tiedonsiirto: välitetään kunkin havainnon osalta vain tieto klusterista, johon se kuuluu ja etäisyys klusterista tai etäisyydet useista/kaikista klustereista. (Tätä kutsutaan myös nimellä *vektorikvantisaatio*= kvantisoidaan datavektorien arvot johonkin pienem joukkoon sovittuja 'perusarvoja')

13.5 Klusteroinnin sovelluksia kieliteknologiassa

- Syntaktisten tai semanttisten sanaluokitusten automaattiseksi löytämise tai hypoteesien tarjoamiseksi
- Sanojen ryhmittely kielimallinnusta varten (yleistäminen)
- Ohjaamaton sananmerkitysten disambiguointi (kontekstien klusteroinnti)
- Puheentunnistuksen foneemimallien (tai ali-sellaisten) ryhmittely (yleistä
- Dokumenttien ryhmittely tiedonhakua ja eksplorointia varten

Huom: tässä käytiin läpi vain muutamia esimerkkejä tavallisimmista klusterointialgoritmeista. Klusterointi käsitellään kattavammin ja yksityiskohtaisem min kurssilla T-61.231 Hahmontunnistuksen perusteet.

13.6 Visualisointi ja datan eksplorointi

Ohjaamattoman oppimisen ja klusteroinnin eräs tärkeä sovellusalue on *exploratory data analysis (EDA)*.

Klusterointimenetelmien lisäksi tällä alueella sovelletaan ja kehitetään datan *visualisointimenetelmiä*, kuten:

- Pääkomponenttianalyysi (principal component analysis, PCA)
- Moniulotteinen skaalaus (multidimensional scaling, MDS)
- Itseorganisoiva kartta (self-organizing map, SOM)

Itseorganisoivasta kartasta

Itseorganisoiva kartta on itse asiassa samanaikaisesti klusterointi- ja visualisointimenetelmä. Klusterointimenetelmänä se muistuttaa K-meansia.

Keskeinen ero on että SOM:in prototyypit muodostavat aineistosta epälineaarisi järjestyneen kuvauksen (tavallisesti) 2-ulotteiselle hilalle. Kuvaus järjestyy itsestään mallin opetuksen aikana.

SOMin muodostama kuvaus eroaa MDS:n muodostamasta siten että MDS (eli 'Sammonin kuvaus') painottaa enemmän pitkien (globaalien) etäisyyksien esittämistä oikein.

SOM taas painottaa enemmän lokaalien (lyhyiden) etäisyyksien esittämistä, ja käyttää samalla visualisointipinta-alan paremmin hyväkseen. Kummallakin kuvauksella on meriittinsä.

Luennolla esitetään esimerkkejä itseorganisoivan kartan kieliteknologian alueen sovelluksista.

14. Tekstin luokittelu

(Text categorization)

Ohjatussa oppimisessa tyypillinen tehtävä on *luokittelu*.

Luokittelussa on annettuna joukko näytteitä joille kullekin tunnetaan luokkamuuttujan arvo. Tehtävänä on valita tai päätellä luokkamuuttujan optimaalinen arvo uudelle näytteelle, jonka luokkaa ei tunneta.

Päättely tehdään vertailemalla näytteen ominaisuuksia sekä luokiteltujen näytt ominaisuuksia, sekä hyödyntämällä tietoa tunnettujen näytteiden luokkamuuttujien arvoista.

Potentiaalisia sovelluksia SNLP:n piiristä on tuotu esiin aiemmin kurssin aikana, esim. sananmerkitysten yksikäsitteistäminen, ja tiedonhaun piirissä dokumenttien kategorisointi joihinkin ennalta annettuihin aiheluokkiin.

Erilaisia luokittelumenetelmiä käsitellään tarkemmin muilla informaatiotekniikan laboratorion kursseilla.

14.1 Evaluointi

Tavallisin mitta: classification accuracy = oikein luokiteltujen osuus kaikista näytteistä.

Lisäksi tiedonhausta tutut mitat, kuten precision, recall, fallout

Julkisesti saatavia datakokoelmia tekstidokumenttien luokitteluun mm. Reuters collection.

Evaluointiin sovelletaan aiemmin kuvattuja periaatteita aineiston jaosta osiin.

14.2 Esimerkkejä luokittelumenetelmistä

Seuraavaksi esitellään joitain esimerkkejä hyvin erityyppisistä, yleisesti käytetyis
luokitusmenetelmistä.

- Naive Bayes -luokitin
- Päätöspuut
- Monikerrosverkko
- KNN (K-nearest-neighbor)

Naive Bayes -luokitin

Naive Bayes -luokitin (esiteltiin kurssin alkupuolella) on esimerkki tilastollisista luokittimista, jotka soveltavat Bayesin päätösteoriaa.

Erotuksena monimutkaisemmista Bayes-luokittimista, Naive Bayes olettaa piirteiden vaikuttavan luokitukseen toisistaan riippumattomasti.

Menetelmä toimii yllättävän hyvin huolimatta siitä että oletus ei useinkaan pidä paikkaansa.

Käytetään usein baseline-luokitusmenetelmänä.

Päätöspuu (decision tree)

Osittaa dataa aina jonkin piirteen mielessä kerrallaan, kunnes jokaisessa osassa on pelkästään yhden luokan alkioita.

Osiin jako tapahtuu ns. maksimaalisen informaationlisäyksen kriteerillä (maximum information gain): Valitaan sellainen piirre-arvo -pari jolla jakaminen maksimoi äiti-noodin ja sen lapsi-noodien välisen luokkaentropian erotuksen.

Puita rakennettaessa yleensä ensin jaetaan dataa osiin yhä uudelleen, ja lopuksi 'karsitaan' puuta ylioppimisen välttämiseksi.

Päätöspuu-menetelmän ominaisuuksia

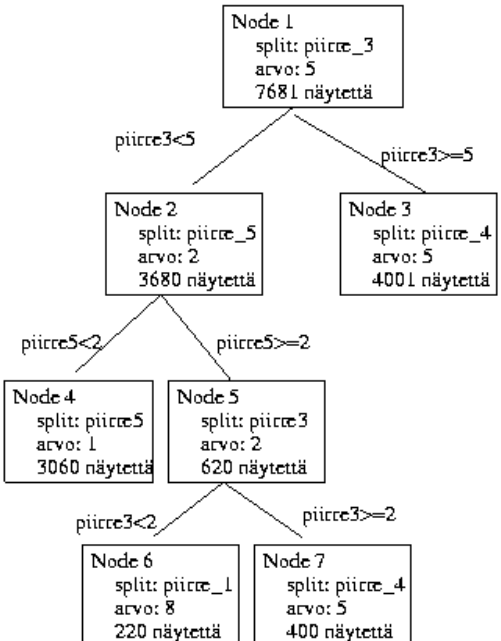
Menetelmän paras puoli on se että se on ihmiselle helposti ymmärrettävä ja helposti tulkittavissa.

Menetelmän eräs huono puoli on, että se tarkastelee vain yhtä muuttujaa kerrallaan, ts. jakaa data-avaruutta vain alkuperäisten muuttujien suunnissa (ks. kuva). Useat ongelmat ovat kuitenkin aidosti monimuuttujaisia.

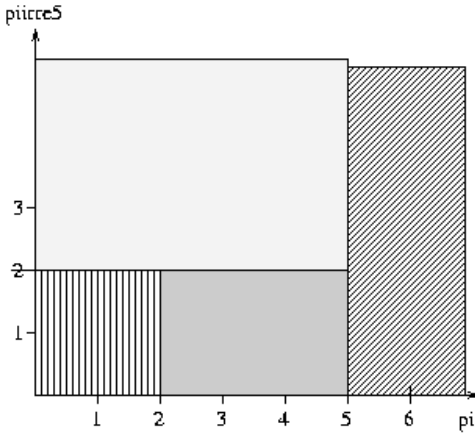
Menetelmä myös johtaa helposti epäoptimaalisiin partitiointijärjestyksiin, ja epätasapainoisiin puihin, jonka seurauksena päätöksenteko hidastuu radikaalisti ($\mathcal{O}(\log(n)) \rightarrow \mathcal{O}(n)$).

Puiden karsimisesta huolimatta ylioppiminen on menetelmälle tyypillinen ongelma.

Päätöspuu



Luokittelupäätösten geometrinen tulk



Monikerrosverkko (multi-layer perceptron, MLP)

Monikerrosverkko on ehkä tunnetuin neuroverkkoluokitin. Tunnetaan yleisesti myös nimellä backpropagation network. Menetelmä käydään läpi mm. neuraalilaskennan peruskurssilla.

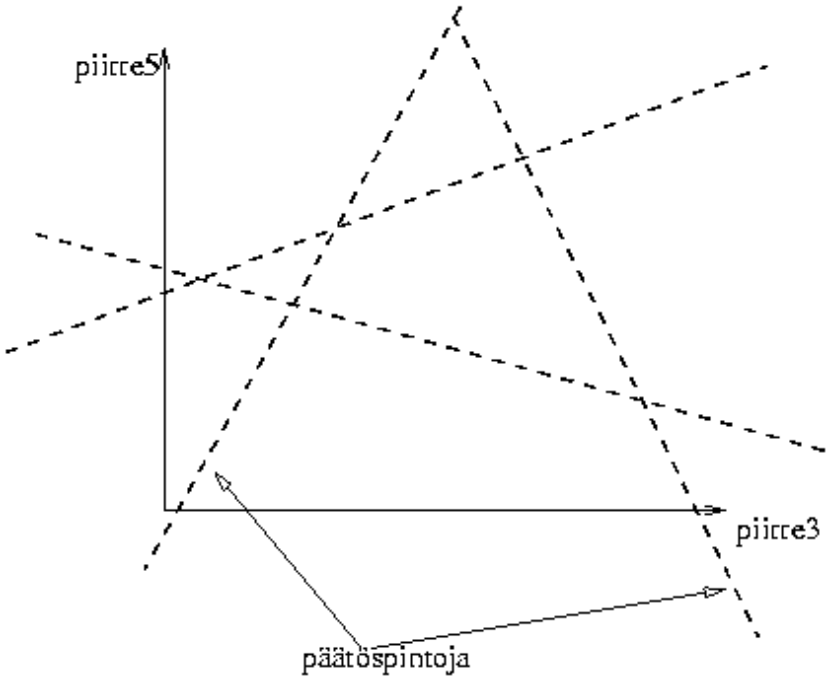
MLP jakaa data-avaruutta osiin hypertasoilla, jotka eivät rajoitu alkuperäisten muuttujien määräämiin suuntiin (ks. kuva, päätöspinnat).

MLP asettaa päätöspinnat Bayesin päätösteorian mielessä optimaalisiin sijainteihin.

Hankaluutena on mm. verkon välikerrosten neuronien määrän valinta, sekä epälineaarisen menetelmän mahdolliset lokaalit optimit.

Menetelmästä on kehitetty suuri joukko variantteja.

MLP:n geometrinen tulkinta



K:n lähimmän naapurin luokitin (k nearest neighbor, KNN)

KNN on ei-parametrinen menetelmä, joka tekee luokituspäätöksen äänestyksen näytteen kanssa lähimpien k:n luokitellun näytteen kesken.

Tasatilanteet ratkaistaan arvalla.

Kutsutaan joskus myös nimellä Memory-based learning (MBL), koska pienen joukon malleja sijasta menetelmässä tallennetaan koko opetusdata.

Etäisyysmitta voidaan valita vapaasti tai jopa vaihdella tilanteen mukaan.

Toimii usein yllättävän hyvin yksinkertaisuudestaan huolimatta.

Mikäli dataa on käytettävissä kovin paljon, menetelmä on luokittelun aikana hidas ja vie paljon muistia: jokaisen luokiteltavan näytteen kohdalla on laskettava sen etäisyys kaikkiin opetusjoukon näytteisiin.

Usein käytetty baseline-menetelmä.