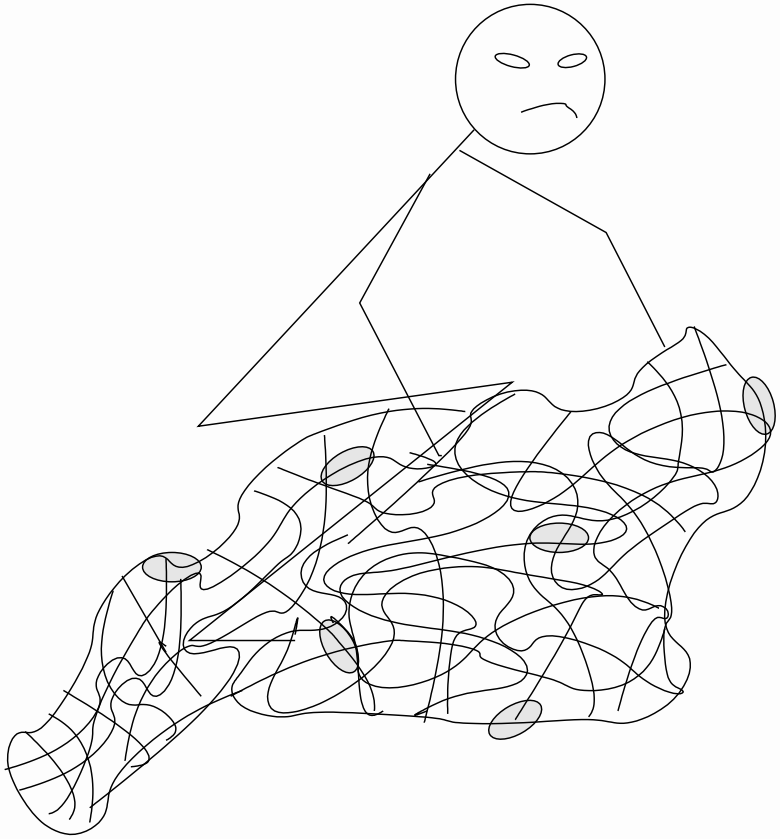
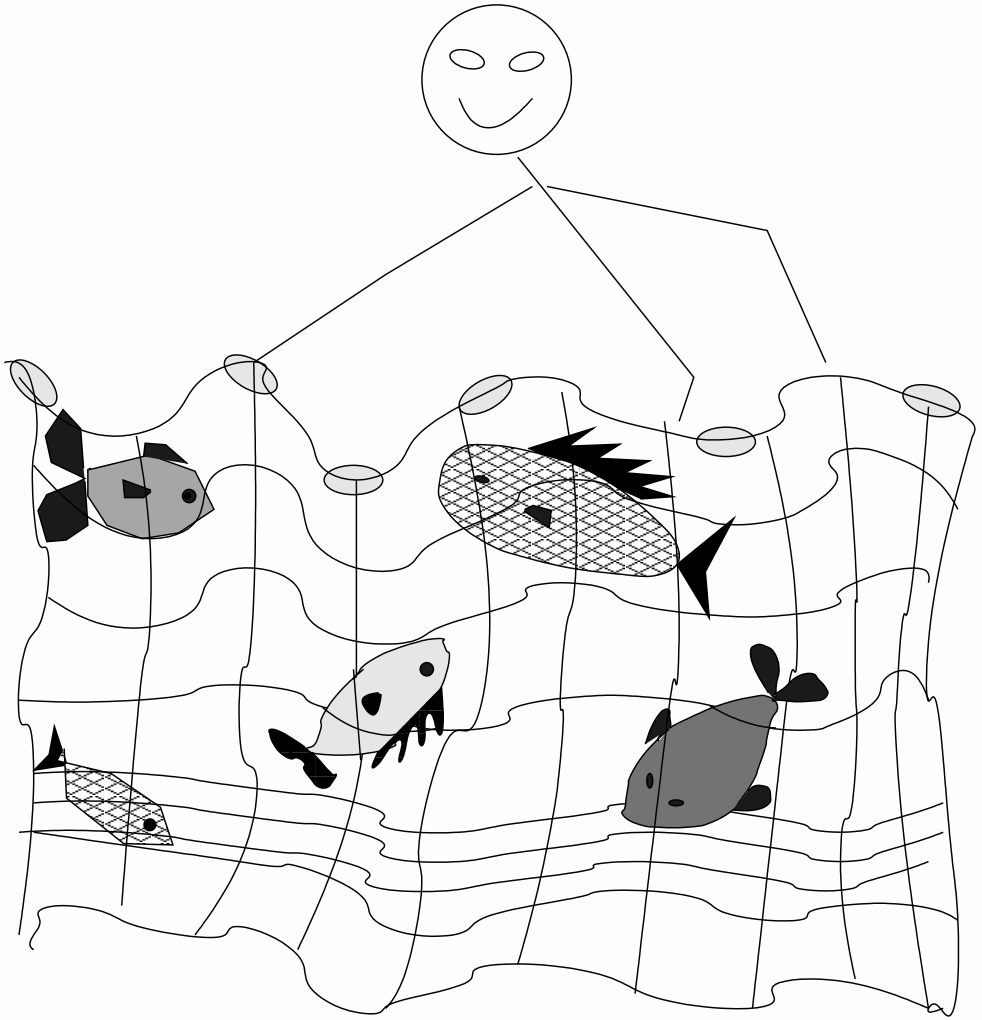
Tiedonlouhinta (kl 2013)





Kurssin kuvaus

- joko kandidaatti- tai maisteritason valinnainen kurssi
- Toteutus pääasiassa ongelmalähtöisesti (tiedonlouhintaprosessin harjoittelu oikealla datalla)
- Yleiskuva menetelmistä ja niiden valinnasta luennoilla (vain 4 vk)
- Perustaitojen harjoittelu laskuharjoituksissa (4 vk)
- Ei koetta, mutta edellyttää aktiivista työskentelyä kurssin aikana! (4op $\approx$ 108h)

Esitiedot

- TKT-opiskelijat: Ohjelmointi-kurssit, TRA1 ja perus-linux-taidot
- Muiden alojen ihmiset: perustaidot tietokoneen käytöstä ja sopiva **datajoukko** harjoitustyöaiheeksi
- Täytä esitietokysely heti! (kurssisivulla)
- Luentoja saavat tulla kuuntelemaan muutkin (mutta harjoitusryhmiin ei ylimääräisiä)

Kurssin arvostelu

- Harjoitustyöt 75% pisteistä (sis. esityksen arvion)
- Harjoituspisteet 25% pisteistä (mikroharjoituksista 3 läsnäolopistettä)
- Esitykset arvioidaan yhdessä
- Ryhmä saa ehdottaa, miten harjoitustyöpisteet jaetaan ryhmäläisten kesken

Opettajat

Wilhelmiina Hämäläinen (luennoija, Kuopion harjoitukset ja ohjaus) whamalai@cs.uef.fi
Kurssin ajan vastaanottoaika ti klo 15-16 (Technopolis F007) tai sopimuksen mukaan

Mikko Malinen (Joensuun harjoitukset ja ohjaus) mmali@cs.uef.fi

Alustava ohjelma

Viikko 15 Luento ma 8.4. Luento ti 9.4. Muu	 Kurssin esittely, johdatus tiedonlouhintaan Datan tyypit ja esiprosessointi. Harjoitustöiden jako ja ohjeistus. Mikroharjoitukset. Tapaamiset ohjaajan kanssa
Viikko 16 Luento ma 15.4. Luento ti 16.4. Muu	 Tilastollisten riippuvuuksien haku Bayesläinen riippuvuuksien mallinnus Harjoitukset. Loput tapaamiset ohjaajan kanssa

Viikko 17 Luento ma 22.4. Luento ti 23.4. Muu	 Luokittelusäännöistä ja luokittelijoista Klusterointi Harjoitukset
Viikko 18 Luento ma 29.4. Luento to 2.5. Muu	 Erikoisempien hahmojen haku (verkkodata, aikasarjadata) Oikean menetelmän valinta. Tulosten järkevyyden ja merkitsevyyden arviointi. Harjoitukset

Viikko 19 Muu	Omatoimista työskentelyä. Tapaamiset ohjaajan kanssa
Viikko 20	Omatoimista työskentelyä.
Viikko 21 Luento ma 22.5. Luento ti 23.5. Muu	 Loppuesitykset Loppuesitykset Loppuraporttien palautus

Mitä on Tiedonlouhinta?

- Data mining (DM) (“data fishing”, “data snooping”, “information harvesting”)
- Knowledge discovery (from databases) (KDD)

Intuitiivisesti

- Laskennallisesti epätriviaalia data-analyysia (uuden informaation etsimistä datasta)

Määritelmiä

“A non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data.” (Fayyad et al. 1996)

“Automatic or semi-automatic analysis of large quantities of data to extract previously unknown interesting patterns such as groups of data records, unusual records and dependencies.” (wikipedia)

“Analysis of (often large) observational data sets to find unsuspected relationships and to summarize the data in novel ways that are both understandable and useful to the data owner.” (Hand et al. luku 1)

Data-analyysi

- raaka data=prosessoimattomia ja tulkitsemattomia faktoja, esim. mittausten tuloksia
- datalla itsessään ei mitään merkitystä
- informaatio=tietoa jolla on merkitys, “tulkittua dataa”
- informaation voi esittää myös rakenteellisena mallina
- suhteellisia termejä: yhden prosessin tuloksena saatu informaatio voi olla toiselle prosessille dataa

Esimerkki: Data vs. informaatio

Data: maanpinnan korkeus eri mittauspisteissä ympäri Suomea eri vuosina

Informaatiota:

- Suomen korkein kohta on Haltilla
- suurin osa maasta on alankoa (alle 200m merenpinnasta)
- Suomen keskikorkeus on n. 154m
- Oulun–Kemin alueella maa kohoaa n. 9 mm vuodessa
- maanpinnan korkeusmalli

Data-analyysia harjoittavia tieteenaloja

Tiedonlouhinnan lisäksi

- tilastotiede (erityisesti deskriptiivinen tilastotiede, eksploratiivinen data-analyysi)
- koneoppiminen (machine learning) (sis. hahmon tunnistuksen, pattern recognition)
- informaation eristäminen tekstistä (information extraction from text) (luonnollisen kielen prosessoinnin, tiedonhaun (information retrieval) ja DM:n välimaastossa)

Tiedonlouhinta vs. tilastotiede?

- Tiedonlouhinnan juuret eksploratiivisessa data-analyysissa (exploratory data analysis): tavoite etsiä uusia piirteitä datasta
- → laskennallisesti hankalaa! Tarve algoritmeille
- Tiedonlouhinta mahdollisti monimutkaisempien mallien etsinnän
- Keskeinen ero **laskennallinen elementti** – tiedonlouhinta edellyttää epätriviaalia laskentaa eli kekseliäitä algoritmeja
- Käytännön eroja:
  - tiedonlouhinnassa paljon suuremmat datajoukot
  - tilastotieteessä tilastollisen merkitsevyyden arviointi keskeistä

Tiedonlouhinta vs. koneoppiminen?

Rajanveto vaikeaa!
Joskus molemmat luetaan kattokäsitteen Knowledge discovery alle

Koneoppimisen osa-alueet

- Ohjattu koneoppiminen (supervised ML): ennustava (prediktiivinen) mallinnus (esim. luokittelu)
- Ohjaamaton koneoppiminen (unsupervised ML): kuvaava (deskriptiivinen) mallinnus (esim. klusterointi)

↔ Tiedonlouhinnassa lähinnä deskriptiivistä mallinnusta

Filosofinen ero

Koneoppiminen

- Uskomus että data on jonkin mallin tuottamaa.
- Tavoite löytää datan parhaiten selittävä/kuvaava malli.

Tiedonlouhinta:

- Data on ensisijaista. Sillä ei välttämättä ole mitään mallia tai vain lokaaleja hahmoja (osittaismalleja).
- Tavoite löytää kaikki kiinnostavat hahmot.

Malli vs. hahmo (Hand et al luku 6)

Malli (model)

- kattava, abstrakti esitys datasta
- globaali eli koskee kaikkia datapisteitä
- esim. Kaikki opiskelijat voidaan luokitella laiskoihin ja ahkeriin laskuharjoituspisteiden ja luentoaktiivisuuden perusteella.

Hahmo (pattern)

- jotain datan osaa kuvaava piirre (paikallinen malli)
- lokaali eli koskee vain tiettyjä datapisteitä
- esim. Korkeat harjoituspisteet indikoivat hyvää arvosanaa.

DM vs. ML: Käytännön eroja (yleistys)

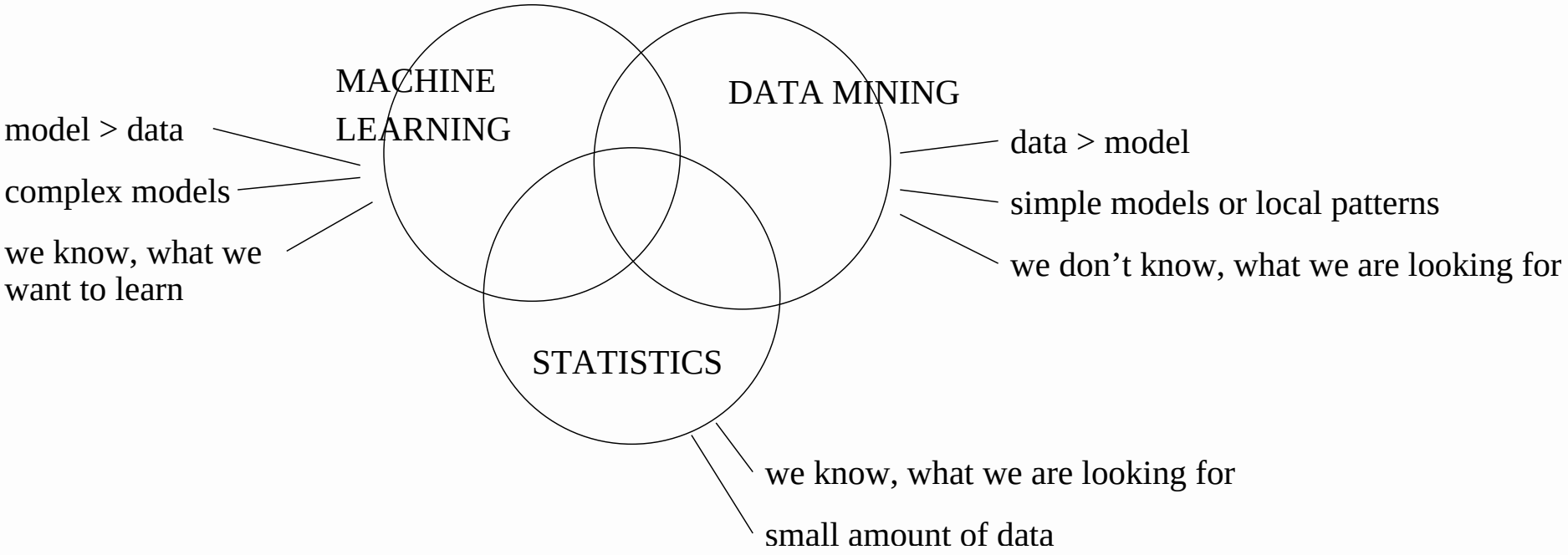
Koneoppimisessa

- monimutkaiset mallit → huonommin skaalautuvat algoritmit
- pienemmät datajoukot (satoja–tuhansia rivejä, kymmeniä muuttujia)
- data ja muuttujat valittu huolella mallinnustarkoitukseen

Tiedonlouhinnassa

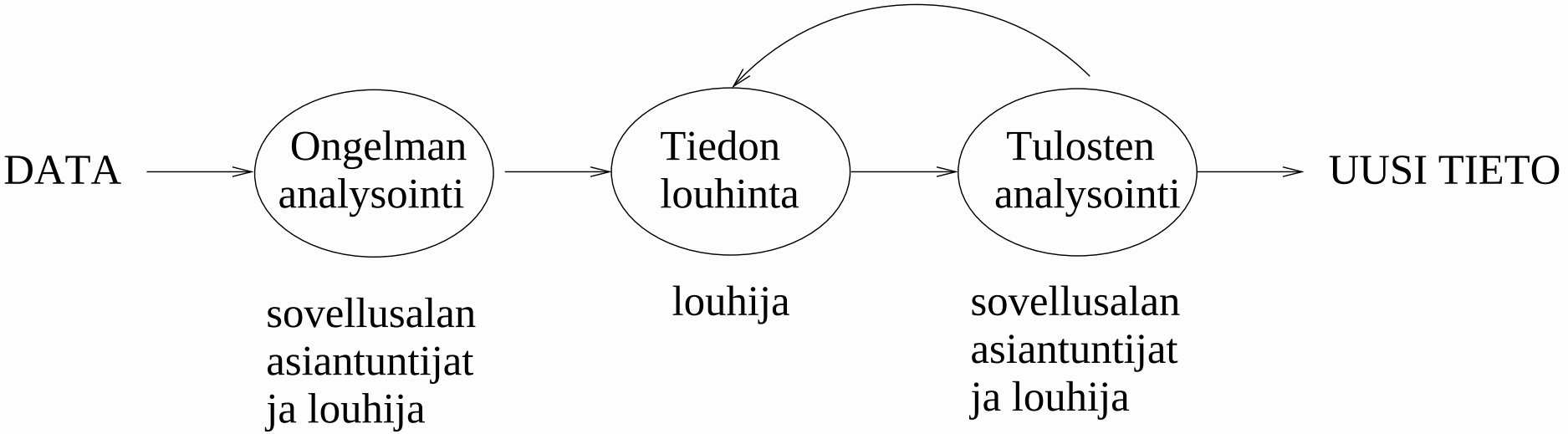
- suhteellisen yksinkertaiset hahmot → tehokkaammat algoritmit
- suuret datajoukot (jopa miljoonia rivejä, satoja–tuhansia muuttujia)
- data saatu usein jonkin toisen projektin sivutuotteena

Yhteenveto: Tieteenalojen suhteet

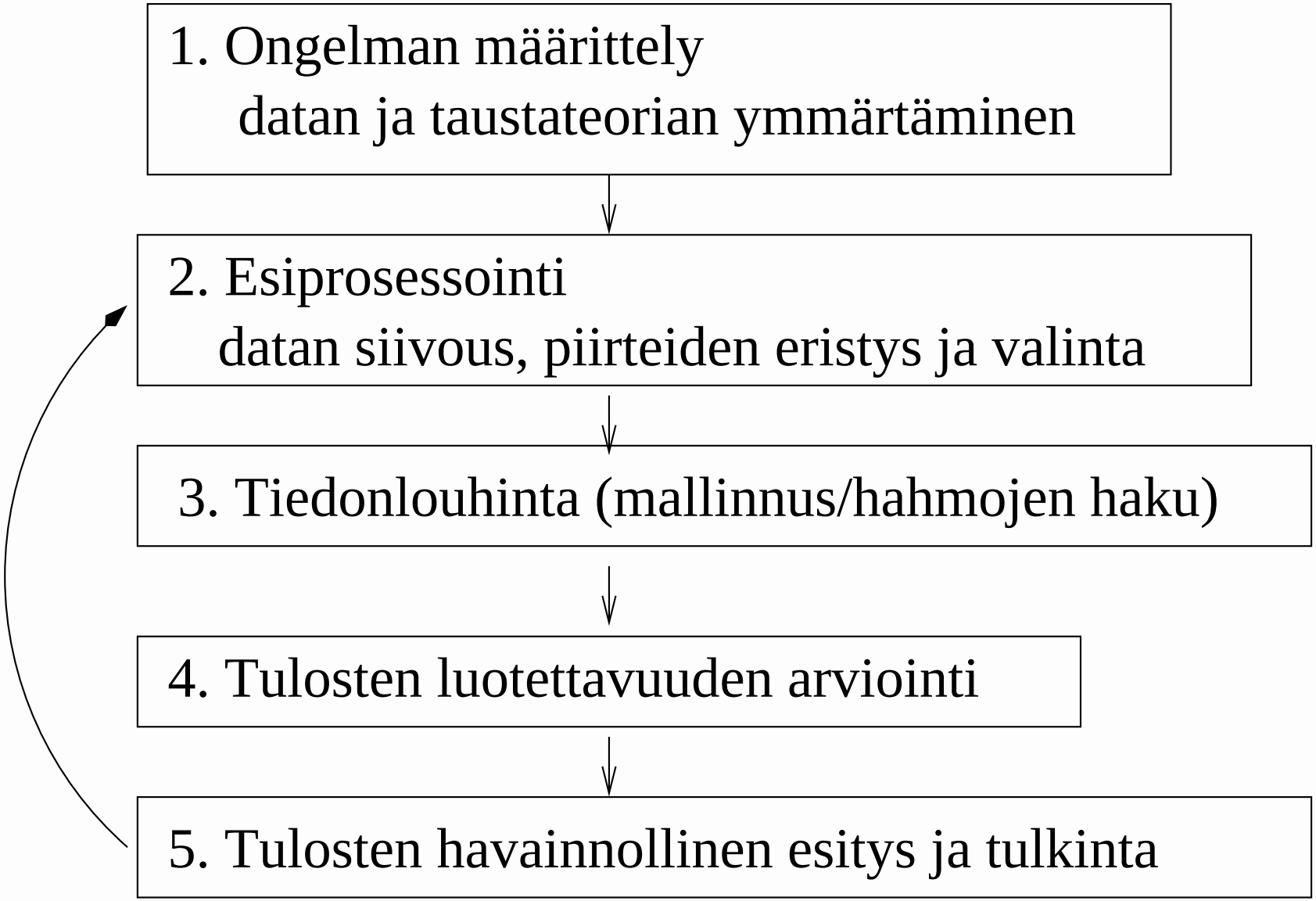


Tiedonlouhintaprosessi

Edellyttää yhteistyötä eri alojen ihmisten välillä!



Prosessin tyypilliset vaiheet



1. Ongelman määrittely

- Datan ymmärtäminen: mitä muuttujat mittaavat/kuvaavat?
- Paljonko dataa on (suhteessa muuttujien määrään)?
- Millaisia hahmoja olisi kiintoisaa/hyödyllistä löytää?
- Mitä tiedetään ennestään?
- Taustateoriaan kannattaa louhijankin perehtyä!
- Vaikeus: Eri alojen ihmisten löydettävä yhteinen kieli (koeta puhua siten, että “maallikkokin” ymmärtää)

2. Esiprosessointi

- Datan yhdistys (eri lähteistä) (voi vaatia normalisointia tms.)
- Alustava analysointi: muuttujien keskiarvot, keskihajonnat, jakaumat, korrelaatiot (tilastollisilla työkaluilla)
- Datan siivous: puuttuvien arvojen käsittely, virheiden tarkistus ja korjaus
- Uusien potentiaalisesti hyvien muuttujien johtaminen (esim. muuttujien yhdistys, diskretointi, normalisointi)

2. Esiprosessointi (jatk.)

- Mallinnuksen kannalta hyvien muuttujien valinta
  - Jos mahdollista, aluksi ei kannata karsia mitään, ennen kuin tiedetään, mitkä muuttujat tarpeettomia
  - Jos laskennallisesti mahdotonta, luodaan vaihtoehtoisia datajoukkoja (samaa asiaa kuvaavat vaihtoehtoiset muuttujat eri joukkoihin)
  - Muuttujien valintamenetelmät eivät koskaan takaa optimaalista tulosta! (*NP*-kova ongelma)

3. Tiedonlouhinta

- Aina kannattaa aloittaa riippuvuusanalyysistä!
  → muuttujien ja mallinnusmenetelmien valinta
- Yleensä deskriptiivinen mallinnus auttaa prediktiivisten mallien muodostuksessa
- esim. Deskr: Etsitään 100 merkitsevintä muuttujaan “Diabetes” liittyvää riippuvuussääntöä
- Pred: Etsitään (valikoidusta datasta) paras diabetesta ennustava malli
- Tyypillisiä mallinnusmenetelmiä riippuvuusanalyysi, klusterointi, luokittelu

Tiedonlouhinta: Riippuvuusanalyysi

Annettu joukko kategorisia muuttujia $R = A_1, \ldots, A_k$, etsi merkitsevimmät tilastolliset riippuvuussäännöt muotoa $X \rightarrow C$ ($X \subsetneq R$, $C \in R$)

- esim. $A_1, A_4 \wedge A_9 \rightarrow A_2$ (A_1 ja A_4 lisäävät A_2:n todennäköisyyttä)
- esim. geeni–elintapa–sairaus -datasta voi löytyä *ApoE-e4, alkoholiakohtuudella → Alzheimer*

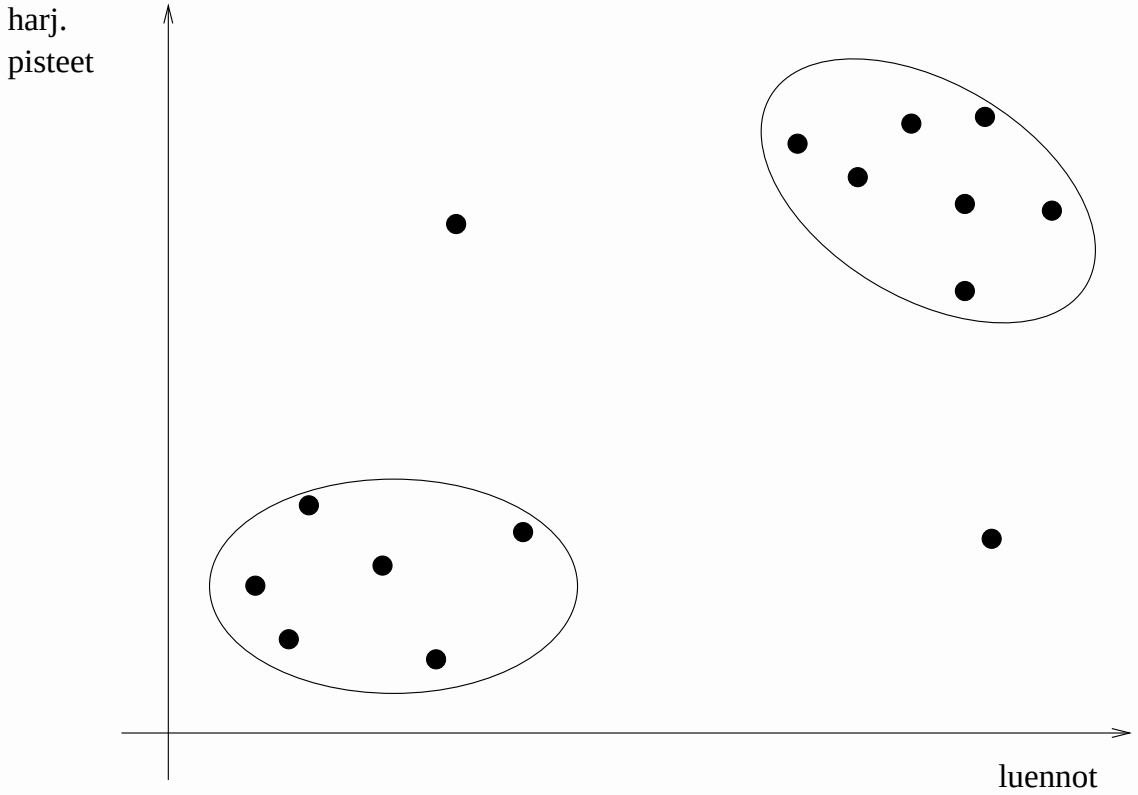
Annettu joukko numeerisia muuttujia $R = A_1, \ldots, A_k$ sekä C, muodosta lineaarinen malli joka ennustaa parhaiten C:tä

- Malli muotoa $C = \beta_1 B_1 + \beta_2 B_2 + \ldots \beta_m B_m + \alpha$, missä $B_i \in R$.

Tiedonlouhinta: Klusterointi

Jaa datapisteet samankaltaisten/lähekkäisten pisteiden muodostamiin ryhmiin eli klustereihin

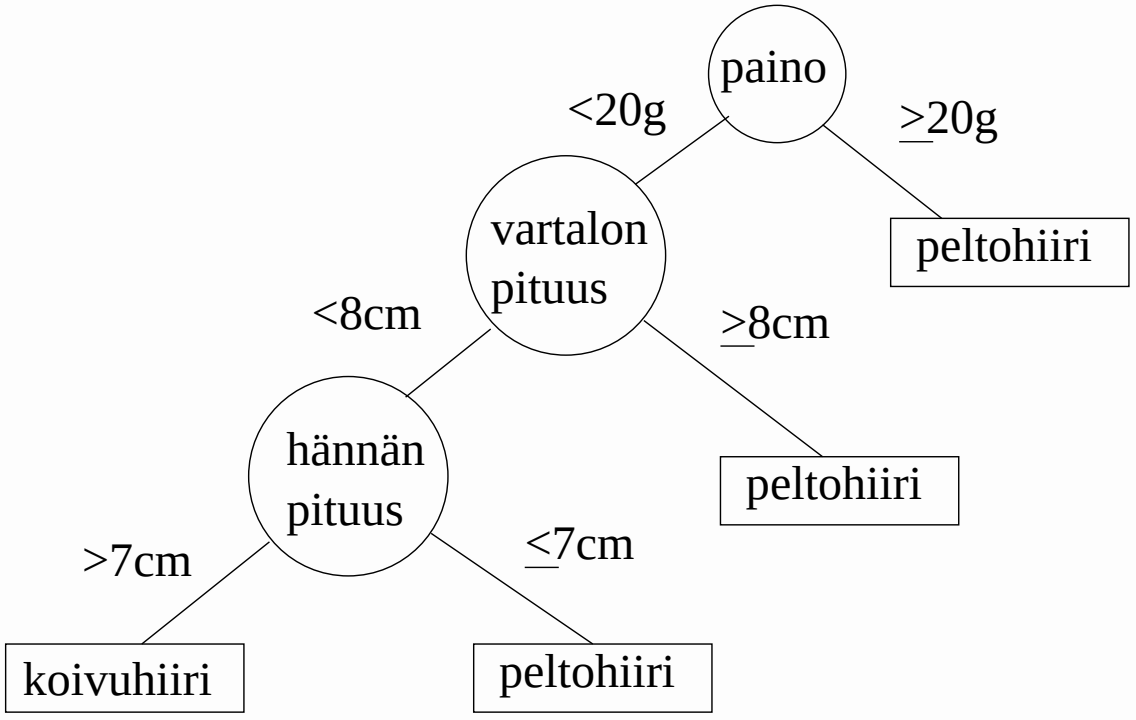
- Esim. opiskelijoiden klusterointi luennoille osallistumisen (kerrat) ja harjoituspisteiden perusteella



Tiedonlouhinta: Luokittelu

Annettuna datapisteiden luokittelu (jako luokkiin), muodosta malli, joka määrittää pisteen luokan muiden muuttujien perusteella.

- esim. päätöspuu pelto- ja koivuhiiren erotukseen painon, vartalon ja hännän pituuden perusteella



4. Tulosten luotettavuuden arviointi

- Ovatko löydetyt hahmot tai mallit järkeviä?? (Muista **aina** tarkistaa!)
  - voi olla, ettei datasta oikeasti löydy mitään mallia – mallinnusmenetelmä palauttaa silloin jonkin satunnaismallin
- Prediktiivisten mallien validointi helppoa:
  - määritetään hyvyysmitta (tai virhemitta, esim. väärin luokiteltujen alkioiden osuus)
  - malli opitaan opetusjoukosta (training set) ja sen virhettä tutkitaan testijoukossa
  - opetus- ja testijoukkojen oltava edustavia!
  - vaihtoehto ristiinvalidointi

4. Tulosten luotettavuuden arviointi (jatk.)

- Deskriptiivisten mallien luotettavuuden arviointi vaikeampaa
- Tavoite: halutaan takuita, että löydetty hahmo ei johdu sattumasta
- Mielellään takuita myös siitä, että samanlainen hahmo pätee jatkossa, samasta–samantapaisesta populaatiosta kerättävässä datassa
- välineenä hahmon tilastollisen merkitsevyyden arviointi

5. Tulosten esitys ja tulkinta

- Tulokset kannattaa esittää mahdollisimman havainnollisesti (esim. graafisesti), jolloin oleellinen erottuu
- Alan asiantuntija keskeisessä roolissa!
- Löytyikö jotain uutta? Voiko tulosten pohjalta muodostaa uutta hypoteesia? Miten sitä voisi tutkia lisää?
- Johtaa useimmiten uuteen louhintakierrokseen: kokeillaan uusia muuttujia ja mahdollisesti uusia menetelmiä
- Kierrokset lopetetaan vasta, kun ollaan tyytyväisiä tai mitään uutta ei näytä enää löytyvän

- “Täydellinen” data-analyysi silti jokseenkin mahdotonta – Miksi?

Tarvitaanko tiedonlouhintaan tietojenkäsittelytieteilijää?

- Soveltava tiedonlouhinta: käytetään yleensä valmiita työkaluja
- Periaatteessa kuka tahansa oppii käyttämään (pitää vain tuntea perusteet – mitä kannattaa tehdä ja miten)
- Valmiita helppokäyttöisiä työkaluja on, mutta ei uusimmista algoritmeista (ohjelmoija osaa toteuttaa algoritmin)
- Joskus ongelma vaatii ihan uutta menetelmää (uudenlaisen hahmon hakua) → tarvitaan tiedonlouhinta-algoritmien suunnittelijaa ja ohjelmoijaa

Tarvitaanko tietojenkäsittelytieteilijää?

- Esiprosessointivaiheessa tarvitaan yleensä ohjelmointitaitoja (uusien hyvien muuttujien muodostus on taidelaji!)
- Mutta: jos data on tosi siistiä, muuttujat hyviä ja työkalu kattava, voi louhinnasta selvitä osaamatta koodata riviäkään.
- Työelämässä tiedonlouhijoilla ja data-analyytikoilla on paljon kysyntää (ja kysyntä kasvaa)!

Datan tyypittely

- Onko rakenteellista (structured)?
- Onko temporaalista vai staattista?
- Minkä tyyppisiä muuttujia?

Rakenteellinen vs. rakenteeton data

- rakenteeton data on vain yksi merkkijono, ilman tunnistettavaa rakennetta
  esim. teksti, ääni, kuva
- rakenteellisen datan esitys noudattaa ennalta määriteltyä rakennetta
  - tyypillisesti: data koostuu riveistä ja jokainen rivi luettelee tiettyjen muuttujien arvot
  - ts. datarivi on joukko **piirteitä** eli muuttuja–arvo-pareja

Rakenteellinen vs. rakenteeton (jatk.)

- muun tyyppisiä rakenteellisia datoja kutsutaan toisinaan puolirakenteellisiksi (semistructured)
  esim. tekstitiedosto jonka kieliopillinen rakenne kuvattu tageilla
  (< *subj* >Kissa< /*subj* > < *pred* >söi< /*pred* >
  < *obj* >hiiren< /*obj* >)
- rakenteettomille datoille annetaan rakenne **eristämällä piirteitä** eli muodostamalla muuttujia ja määrittämällä niiden arvot
- tiedonlouhinnassa oletetaan yleensä, että data on rakenteellista
  – jos ei ole, on ensin määritettävä rakenne

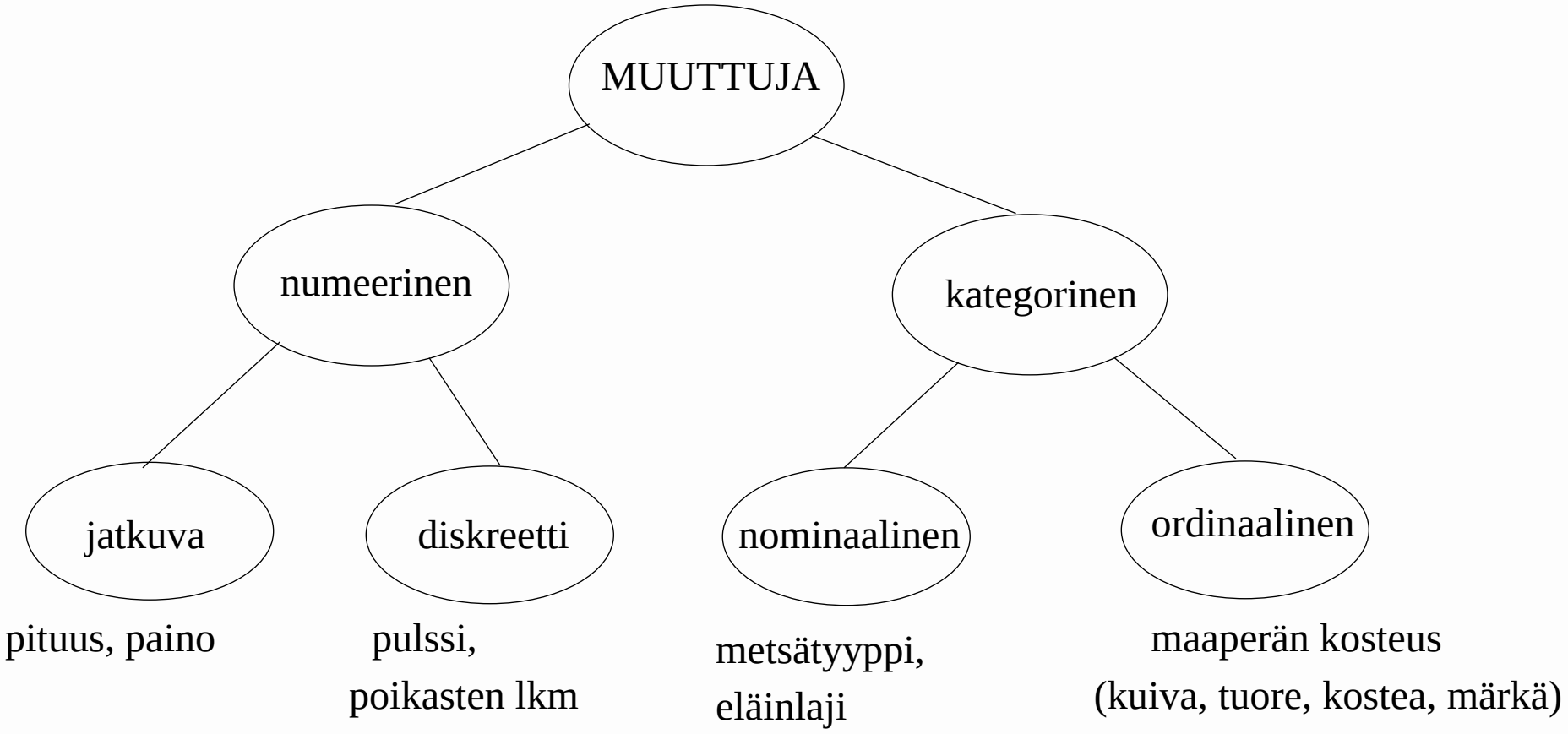
Staattinen ja temporaalinen data

- staattinen: muuttujien arvot eivät muutu ajan kuluessa (tai mitattu vain kerran, tietyllä ajanhetkellä)
- temporaalinen (dynaaminen): muuttujan arvot riippuvat ajanhetkestä
  - laitteiden lokidata (log data): aikaleima, tapahtuma, käyttäjä, ym.
  - aikasarjadata (time series data): muuttujan arvo mitattu tasavälein, esim. lehmän x-,y- ja z-kiihtyvyydet 0.1s välein

Staattinen ja temporaalinen (jatk.)

- aina ajankohdalla ei ole väliä!
  - esim. lokista voi analysoida tapahtumia ajankohdasta piittaamatta
  - tai ajankohdan voi esittää karkealla tasolla (aamu, päivä, ilta, yö)
- jos mittauksia vain parilla–muutamalla ajanhetkellä, voi luoda uudet stattiset muuttujat (esim. painoennenkoetta, painokokeenjälkeen)
- yksilön tai järjestelmän dynamiikan mallinnus edellyttää temporaalisia malleja/hahmoja (ja paljon dataa!)

Muuttujien perustyypit



Muuttujien perustyypit (jatk.)

- diskreetin muuttujan mahdollisia arvoja vain numeroituva määrä (esim. kokonaisluvut) ja tietyllä arvovälillä äärellinen määrä
- jatkuva muuttuja voi saada minkä tahansa arvon annetulta arvoväliltä (eli mahdollisia arvoja ylinumeroituva määrä)
- käytännössä esitetään kuitenkin jollain tarkkuudella (jolloin mahdollisia arvoja annetulla välillä on vain äärellinen määrä)

Muuttujien perustyypit (jatk.)

- ordinaalimuuttujilla on järjestys, vaikka ei numeerista tulkintaa
- nominaalimuuttujien arvot vain nimiä, joilla ei numeerista tulkintaa tai järjestystä
- erikoistapaus binaarimuuttuja, jonka arvo 0 tai 1 (jokin ominaisuus pätee tai ei päde) – ei silti numeerista tulkintaa!
  esim. muuttujat (attribuutit) sairas, naaras, muuntogeeninen

Datan esiprosessointi

Tärkeimmät tehtävät:

1. Datan puhdistus (Data cleaning): Virheiden, puuttuvien arvojen ym. käsittely
2. Piirteiden eristys (Feature extraction): Uusien muuttujien muodostus vanhoja muuntamalla ja yhdistämällä
3. Piirteiden valinta (Feature selection): Hyvien muuttujien valinta

Datan puhdistus

- Tavoite tunnistaa ja eliminoida mahdollisimman hyvin virheet, puuttuvat arvot, duplikaatit, kohina ja “outlierit” (poikkeavat havainnot)
- Automaattisten mittauslaitteiden ongelma lähetyskatkot → dataa voi puuttua tietyltä aikaväliltä tai sama datapala on lähetetty kahteen kertaan (duplikaatteja)
- Jos muuttujat ovat eri lähteistä (esim. askelmittarista lehmän aktiivisuus ja pötsin pH-mittarista pH ja lämpötila), voi puuttuvia arvoja olla vain joissain muuttujissa

Datan puhdistus: Kohina ja virheet

- kohina (noise) = häiriöistä johtuva epätarkkuus/satunnaisvirheet muuttujien arvoissa esim. märehtimismittarin pitäisi äänittää lehmän suun ääniä, mutta taustan traktorin hurina tulee mukaan
- virheet johtuvat usein inhimillisistä syistä (huolimattomuus) tai laitevioista (laite ei toimi oikein)

Datan puhdistus: Outlierit

- “outlierit” voivat paljastaa virheen (tai kiinnostavan poikkeavan tapahtuman!)
- usein outlierit selviävät vasta klusteroidessa (tai sotkevat klusteroinnin)
- muuttujien jakaumia kannattaa tutkia graafisesti! näkee helposti poikkeavat arvot
- joskus outlier ei kuitenkaan ole poikkeava minkään yksittäisen muuttujan suhteen, vaan ainoastaan muuttujien arvokombinaation suhteen

Puhdistus: Virheiden ja puuttuvien arvojen korjaus

Ensisijaisesti korjataan oikeilla arvoilla. Muuten

- jos paljon, kannattaa muuttuja ehkä karsia kokonaan
- jos jollain datarivillä paljon puuttuvia, rivi kannattaa poistaa
- voi korvata muuttujan keskiarvolla tai mediaanilla
- jos mallinnusmenetelmä sallii puuttuvat arvot, voi ne merkitä vain erikoissymbolilla (esim. “?”, “n/a” tai ohjelmassa $-FLT_MAX$)

Huom! Joskus puuttuvat arvot voivat myös paljastaa kiinnostavan hahmon!

Puhdistus: Kohina ja outlierit

- Kohinaa voi pienentää pehmentämällä dataa (data smoothing)
  skaalaus, diskretointi ym. arvoaluetta pienentävät tekniikat → piirteiden eristys
- mallia/sen parametreja valitessa muista: mediaani ei ole yhtä herkkä outlierien vaikutukselle kuin keskiarvo
- robustit mallinnusmenetelmät pienentävät sekä kohinan että outlierien vaikutusta

Piirteiden eristys: Diskretointi

Tärkeä menetelmä!

- (jatkuva) numeerinen muuttuja muunnetaan diskreetiksi numeeriseksi tai kategoriseksi muuttujaksi
  esim. lehmän painon diskretointi: <=400kg “kevyt”, 400–600kg “normaali”, >600kg painava
- binarisointi=diskretoinnin erikoistapaus, kun uusi muuttuja binaarinen
  esim. lämpötilan tilalle binäärinen pakkasta-muuttuja
- Huom! Myös kategorisia muuttujia voi binarisoida
- arvovälin voi jakaa tasakokoisiksi väleiksi, yhtä monta pistettä sisältäviksi tai määrittää osavälit sekventoimalla

Diskretointi (jatk.)

- diskretointi eliminoi kohinaa ja muuta satunnaisvaihtelua (esim. yksilöllisiä eroja), jolloin mallit ja hahmot voivat erottua paremmin
- pelkästään numeerisestakin datasta kannattaa tutkia diskretoitua versiota!
  - vähemmän kohinaa, selvemmät hahmot
  - tehokkaammat algoritmit
  - diskreetin datan hahmot auttavat myös numeerisen mallinnusmenetelmän valinnassa

Piirteiden eristys: Skaalaus

- uusi muuttuja muotoa $X = \alpha Y + \beta$, missä Y vanha muuttuja
- normalisointi: kaikki muuttujat muunnetaan samaan skaalaan
- standardointi: $X = \frac{Y - mean(Y)}{stdev(Y)}$
  - suositellaan jos datassa outliereita ja käytetään Euklidista metriikkaa (klusterointi, lineaariregressio)
  - voi käyttää diskretoinnin esivaiheena (uudet muuttuja-arvot esim. poikkeuksellisen pieni, normaali, poikkeuksellisen suuri)

Piirteiden eristys: Yleistys

- Vastaa diskretointia kategorisille muuttujille – muunnetulla muuttujalla vähemmän arvoja
- nostetaan abstraktiotasoa, esim. yhdistetään laji-muuttujan arvot peltomyyrä, kenttämyyrä, lapinmyyrä ja idänkenttämyyrä uudeksi arvoksi “kenttämyyrät”
- Sekä diskretointi että yleistys tarpeen, jos dataa liian vähän suhteessa muuttujien lukumäärään/arvoalueiden kokoon! (muuten malli ylisovittuu)

Piirteiden eristys: PCA ja ICA

- numeerisille muuttujille, kun **paljon** dataa
- uudet muuttujat vanhojen lineaarisia kombinaatioita
- PCA:ssa (principal component analysis) uudet muuttujat korreloimattomia ja ICA:ssa (independent component analysis) riippumattomia
- PCA edellyttää normaalijakautunutta dataa!

Piirteiden valinta

- Annettuna muuttujajoukko R, mikä on mallinnuksen kannalta paras mahdollinen $X \subseteq R$?
- tärkeää klusteroinnissa ja ennustavassa mallinnuksessa
- Kaikkia mahdollisia osajoukkoja $X \subseteq R$ on $2^k - 1$ kpl, kun $|R| = k \rightarrow$ kaikkia ei voi tutkia! (*NP*-kova ongelma)
- käytettävä heuristisia menetelmiä
  - testataan jokaisen muuttujan hyvyys erikseen
  - muodostetaan X ahneella heuristiikalla

Piirteiden valinta (jatkoa)

- joskus helppo tunnistaa epärelevantit muuttujat
  - jos muuttuja on johdettu toisesta, ei molempia tarvita
  - jos muuttujien jakaumat hyvin samanlaiset, voi ehkä yhdistää
- riippuvuusanalyysi kannattaa tehdä ilman valikointia, jos mahdollista

Ekskursio 1: NP-kova ongelma (NP-hard problem)

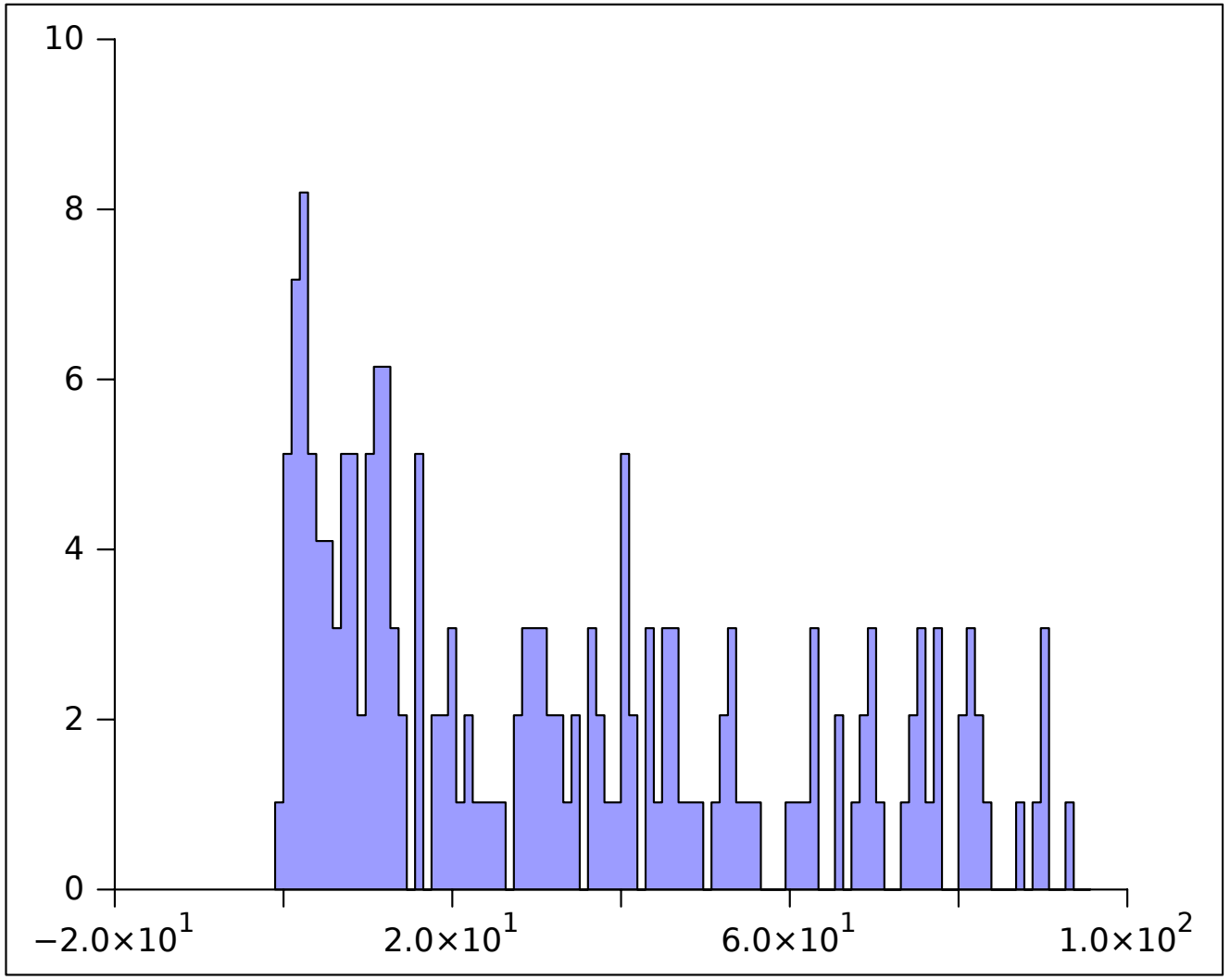
- kertoo että ongelma on laskennallisesti *** vaikea!
- vähintään yhtä vaikea kuin aikavaativuusluokan NP (non-deterministic polynomial time) ongelmat
- todennäköisesti eksponentiaalinen aikavaativuus eli aika-askelten lukumäärä kasvaa eksponentiaalisesti syötteen koon funktiona
- esim. jos ohjelman pahin aikavaativuus $f(k) = 2^k$ aika-askelta, $k =$ muuttujien lkm ja kukin askel 0.001s (riippuu datarivien lkm:stä). Jos $k = 100$, suoritus voi vaatia 400 triljoonaa ($4 * 10^{20}$) vuotta! (“maailmanloppu” jo 4 miljardin v. päästä)
- onneksi datat harvoin niin patologisia! (riippuu jakaumasta)

Ekskursio 2: Visuaalinen muuttujan diskretointi

Perusidea: Tutkitaan muuttujan histogrammia ja etsitään “aukot”

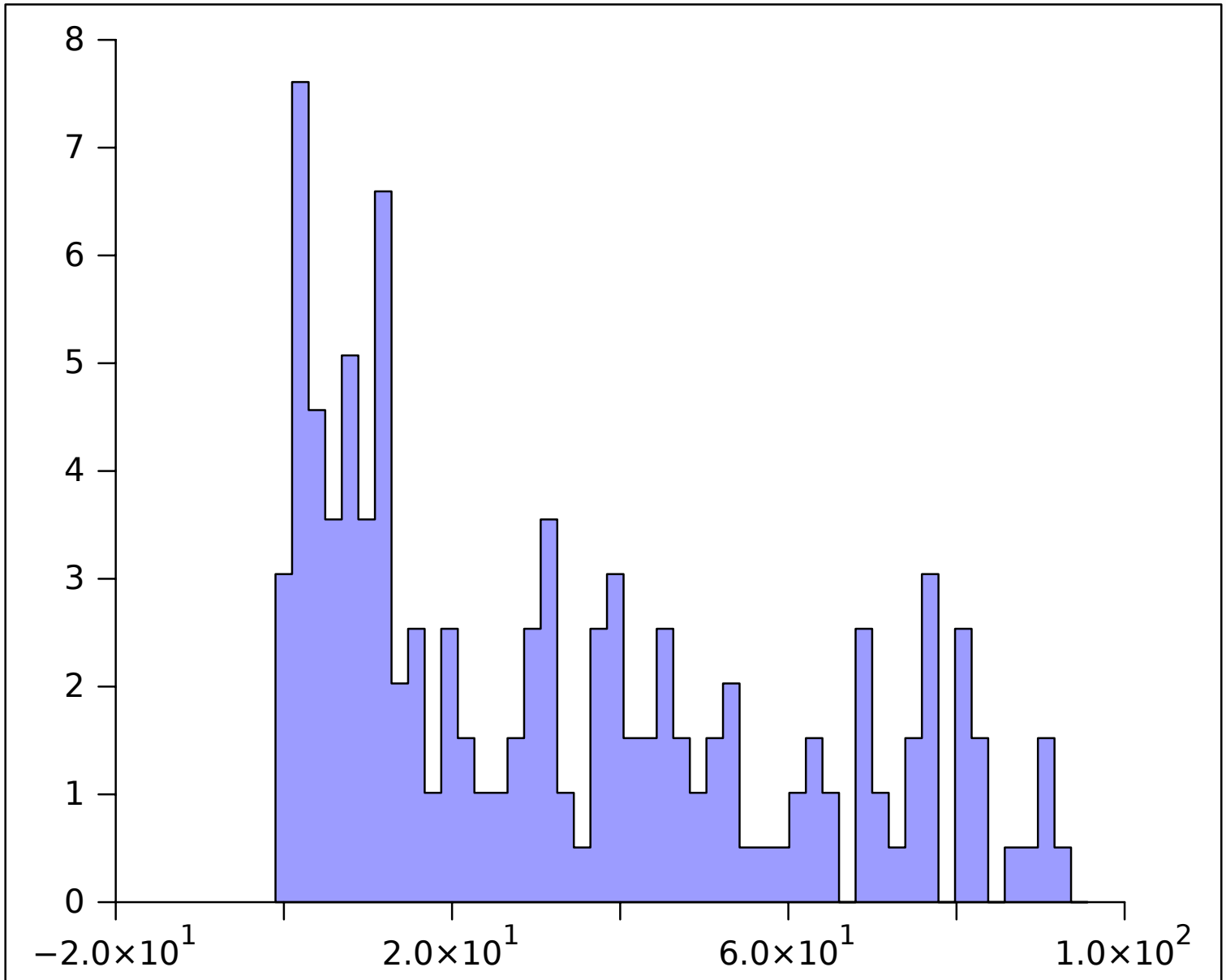
1. Valitse aluksi suuri osavälien (bins) lukumäärä, esim. 100
2. Muodosta histogrammi ja esitä graafisesti
3. Jos histogrammissa selviä kukkuloita ja välillä selviä aukkoja, voi aukkojen kohdalle asettaa diskretointirajan
4. Jos matala tasainen jakauma, vähennä osavälien lkm:ää
5. Jos korkea, mutta ei aukkoja tai edes laaksoja, kokeile lisätä osavälien lkm:ää

Esimerkki: internet-käyttäjien lkm maailman valtioissa (per 100 as.) – 100 bins

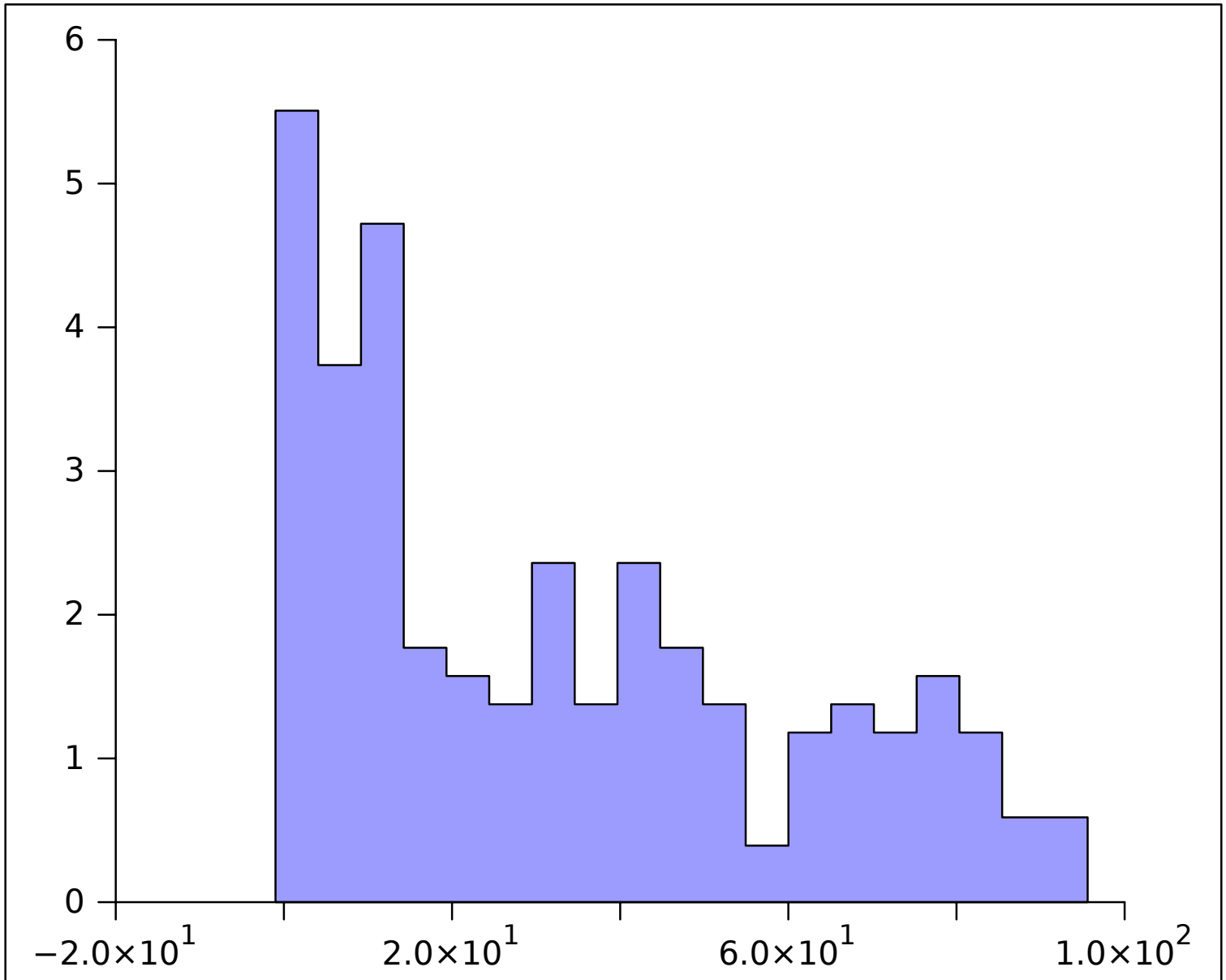


esim. 3-jako:
$x < 25$,
$25 \leq x < 58$,
$x \geq 58$

Esimerkki: internet-käyttäjien lkm maailman valtioissa (per 100 as.) – 50 bins



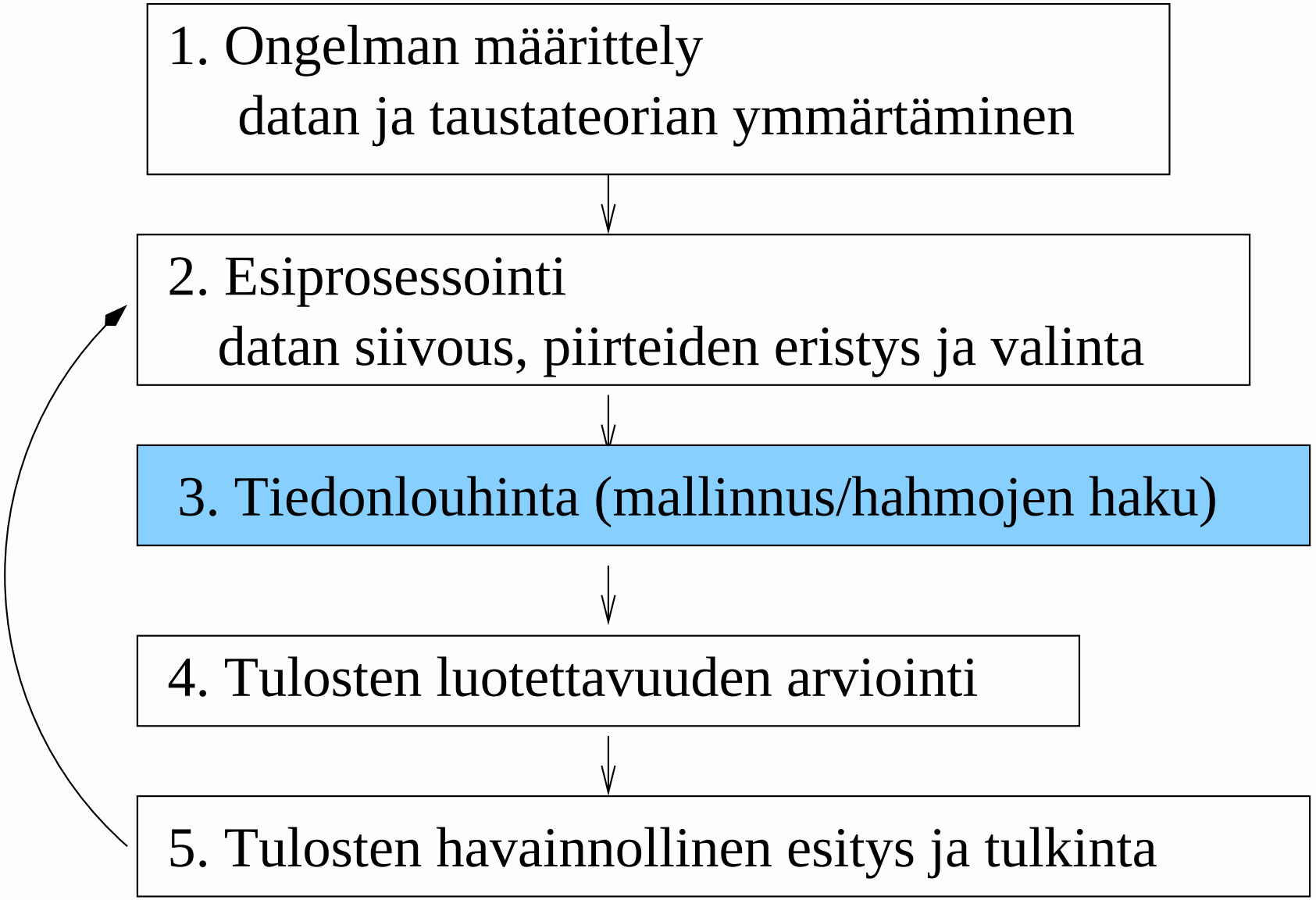
Esimerkki: internet-käyttäjien lkm maailman valtioissa (per 100 as.) – 20 bins



Muodostus gnumericissa

1. Maalaa haluttu sarake (muuttuja)
2. Valitse Statistics → Descriptive statistics → Frequency tables → Histogram
3. Aseta jakovälien lkm (Cutoffs-valikko) ja ruksaa “histogram chart” (Graphs & options -valikko). Sitten ok.
4. Tutki tuloslehteä (kuvan voi siirtää taulukkoesityksen päältä)
5. Histogrammin voi tallettaa kuvana (klikkaa hiiren oikealla napilla)

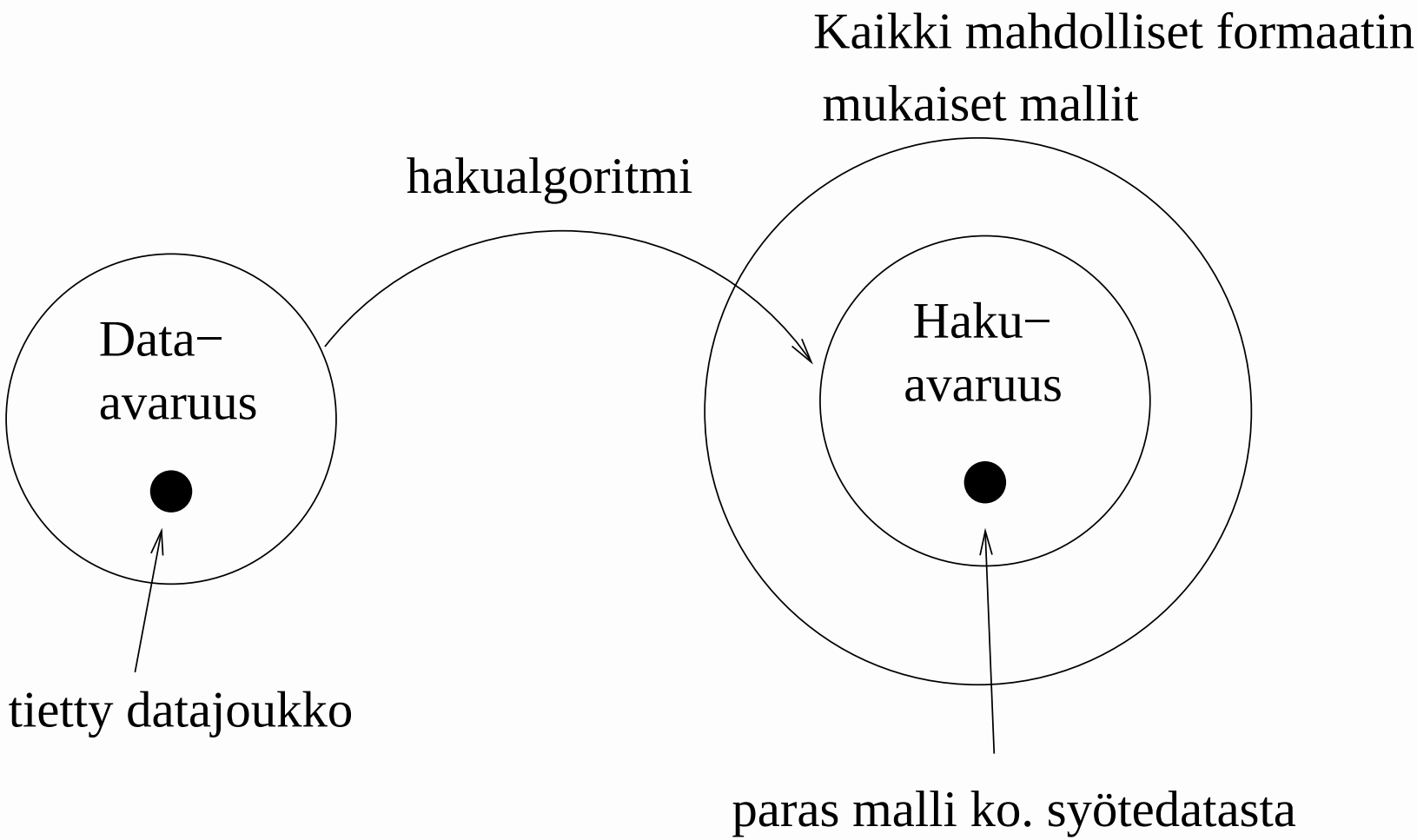
Luento 3: 1 Yleiset mallinnusperiaatteet ja 2 Riippuvuusanalyysi



Yleiset mallinnusperiaatteet

- data-avaruus
  = kaikki mahdolliset muuttujien arvokombinaatiot
- hakuavaruus
  = kaikki mahdolliset annetun formaatin mukaiset mallit/hahmot, jotka datasta voisi johtaa
- globaalisti ja lokaalisti optimaaliset hakualgoritmit
  = Ovatko löydetyt mallit/hahmot varmasti parhaita mahdollisia?
- ylisovittuminen
  Kuvaako malli liian hyvin syötedatan?

Parhaan mallin haku datasta



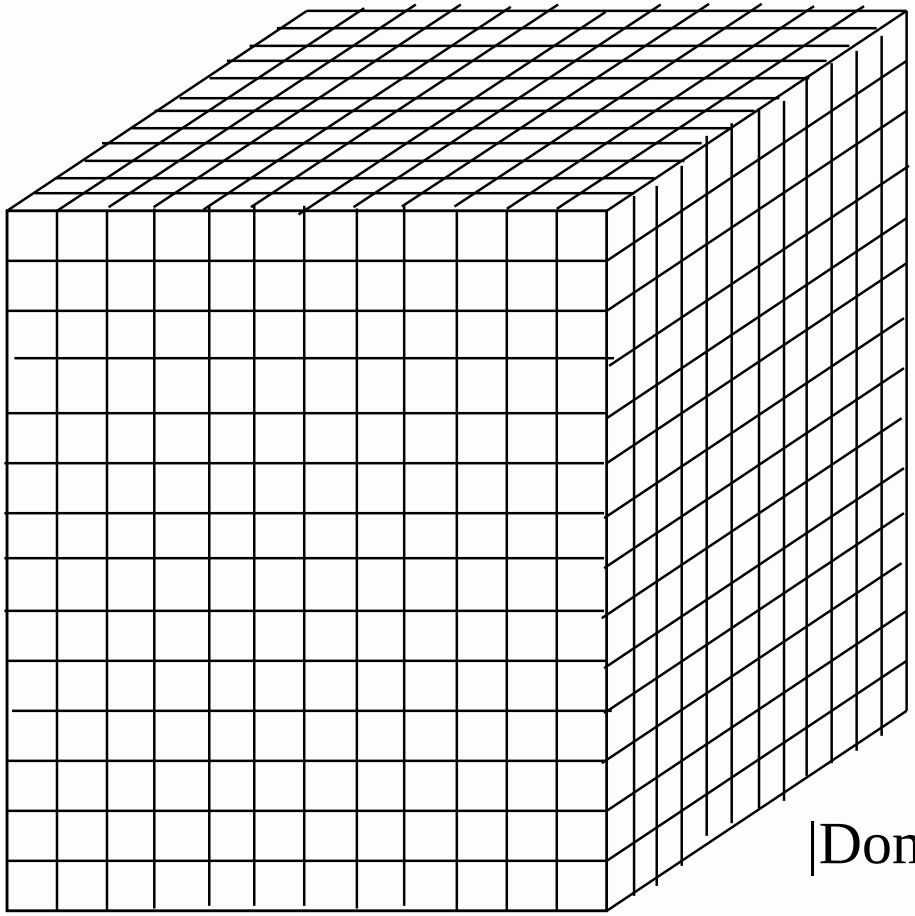
Huom! Annetut muuttujat määräävät data-avaruuden.

Data-avaruus

Olk. muuttujat $A_1, \ldots, A_k$, joiden arvoalueet $Dom(A_1), \ldots, Dom(A_k)$. Silloin $\mathcal{D} = Dom(A_1) \times \ldots \times Dom(A_k)$ määrittää data-avaruuden.

- Datan dimensio k
- Jos kaikki muuttujat diskreettejä, voi kokoa luonnehtia solujen lukumäärällä: $|\mathcal{D}| = |Dom(A_1)| \cdot \ldots \cdot |Dom(A_k)|$.
  - Nyt $|Dom(A_i)| = |\{a_{i1}, \ldots, a_{im}\}| = m$
- Jos jatkuvia muuttujia, ei voi laskea solujen lkm:ää!
  - Nyt $|Dom(A_i)| = |[a_{min}, a_{max}]| = a_{max} - a_{min}$ (voi olla ∞)

Esim. 3-ulotteisen diskreetin datan esitys



Koko $12 \cdot 12 \cdot 14 = 2016$ solua.

Hakuavaruus

- Kattaa kaikki mahdolliset mallit, jotka mallinnusmenetelmän ja data-avaruuden mukaisia ja jotka voi löytää annetulla hakualgoritmilla
- Data-avaruus (annetut muuttujat ja niiden arvoalueet) määräävät, millaisia malleja voi löytää
- Samaan mallinnusmenetelmään (“mallinnusparadigmaan”, esim. päätöspuut, neuroverkot, riippuvuussäännöt) voi liittyä erilaisia hakualgoritmeja!
  → voivat löytää eri malleja

Esimerkki: riippuvuussääntöjen hakuavaruus

Binäärimuuttujat $R = \{A_1, \ldots, A_k\}$ ja etsitään säännöt muotoa $X \rightarrow A$, $X \subsetneq R$, $A \in X$. Hakuavaruus koostuu kaikista mahdollisista säännöistä eli

1. Tutkitaan kaikki mahdolliset R:n osajoukot Y, joiden koko vähintään 2 attribuuttia ($|Y| \geq 2$)
   - R:n **potenssijoukko** (koko 2^k)

$$\mathcal{P}(R) = \{\emptyset, A_1, A_2, \ldots, A_k, A_1A_2, A_1A_3, \ldots, A_{k-1}A_k, A_1A_2A_3, \ldots, \ldots, A_1A_2 \ldots A_k\}$$

   - Ei tarvitse tutkia joukkoja $\emptyset, \{A_1\}, \ldots, \{A_k\}$.

Esimerkki (jatk.)

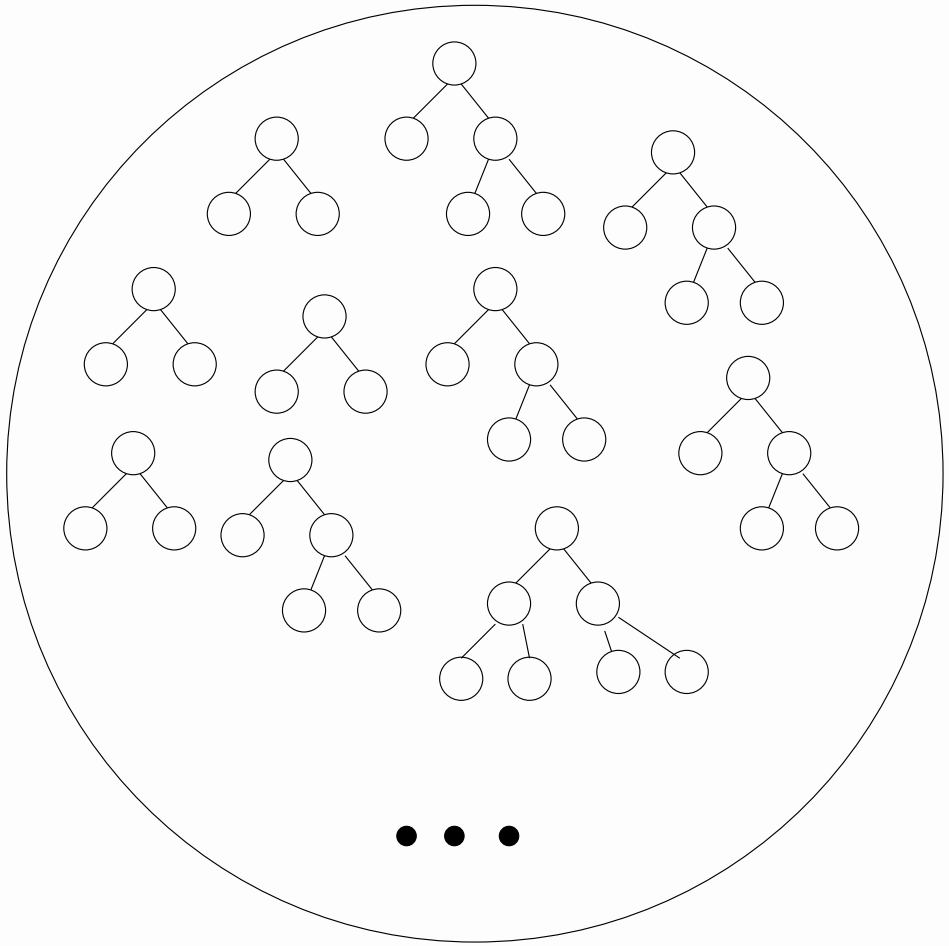
2 Jokaisesta osajoukosta Y muodostetaan kaikki mahdolliset säännöt

- Jos $|Y| = l$, voi muodostaa l sääntöä, $Y \setminus \{A_i\} \rightarrow A_i$ $(\forall A_i \in Y)$.

Hakuavaruuden koko on siis $\sum_{i=2}^{k} i \cdot \binom{k}{i} = O(k \cdot 2^k)$

Hakuavaruus (jatk.)

Jos binäärimuuttujat $R = \{A_1, \ldots, A_k\}$ ja luokkamuuttuja C.
Millainen olisi kaikkien mahdollisten päätöspuiden joukko?



Mallin hyvyysmitta

- Tarvitaan jokin hyvyysmitta M, joka kertoo mallin hyvyyden.
  - esim. miten monta alkiota luokittelee oikein, miten vahva tai merkitsevä riippuvuus on, miten tiheä klusteri on tms.
  - yleensä pyrkii heijastamaan myös sitä, miten hyvin malli yleistyy mallin muodostusdatan ulkopuolelle (ts. että hahmo pätee “oikeasti” tai malli toimii tulevaisuudessa)
- Jos malli paranee, kun M kasvaa, on M “hyvyyden mukaan kasvava” (esim. luokittelun accuracy, χ^2-mitta)
- muuten taas “hyvyyden mukaan vähenevä” (esim. luokitteluvirhe, p-arvot)

Hakuongelma: 2 variaatiota

Annettu mallinnusmenetelmä, data ja hyvyysmitta M.

1. Luettelointiongelma (enumeration problem): Etsi kaikki riittävän hyvät mallit (annettu jokin M:n raja-arvo).
2. Optimointiongelma (optimization problem): Etsi K parasta mallia (ML:ssä yleensä $K = 1$)

- Loppujen lopuksi ei suurta eroa!
- Käyttäjä ei halua tulokseksi tuhansia (saati miljoonia) hahmoja!

$\rightarrow$ Keskitytään optimointiongelmaan

Hakualgoritmin optimaalisuus?

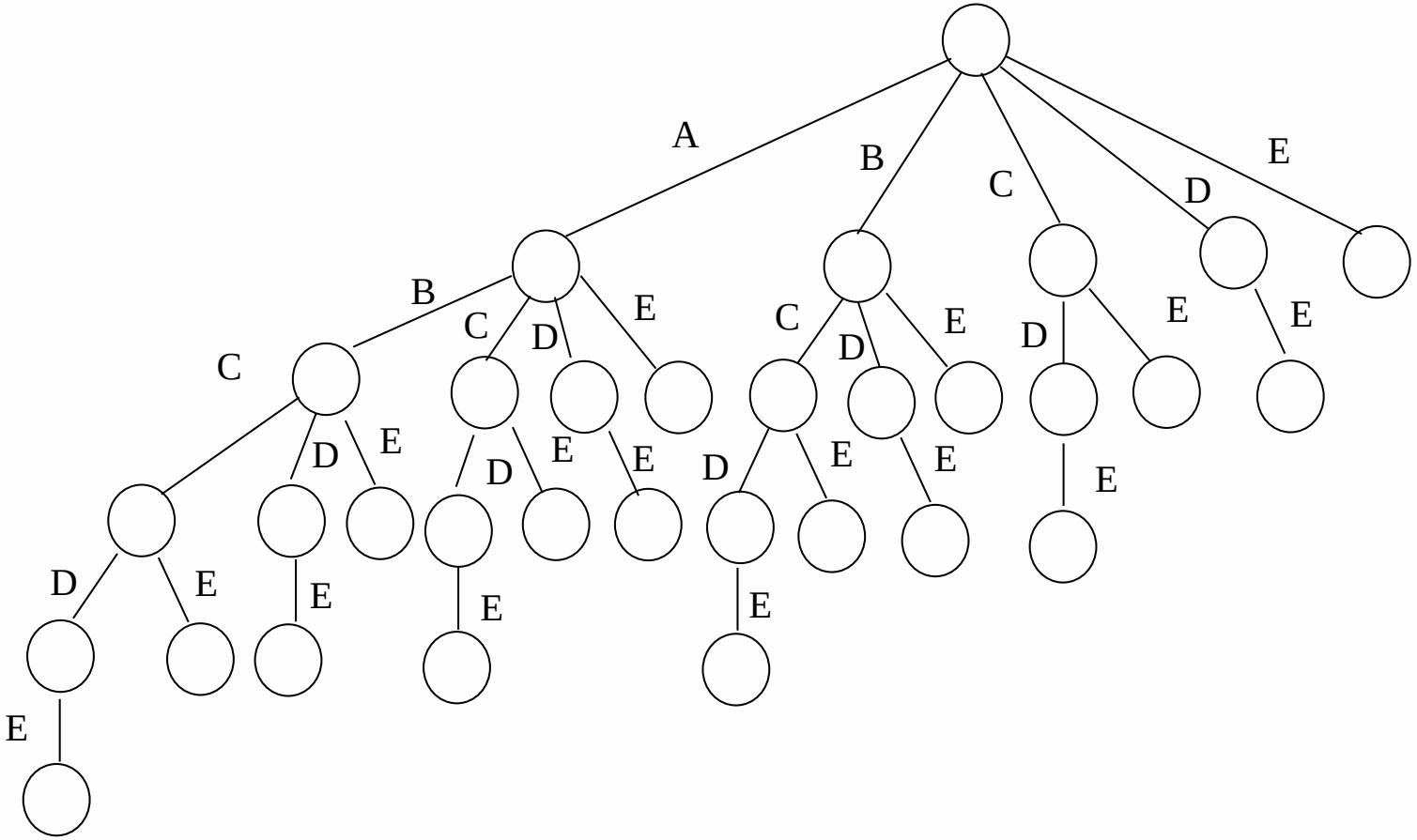
- Globaalisti optimaalinen, jos palauttaa parhaat K mallia hakuavaruudesta
  - Takuu: datasta ei mahdollista löytää näitä parempia annetun muotoisia malleja.
- Lokaalisti optimaalinen: Löydetyt mallit parhaita jossain hakuavaruuden osassa, mutta eivät välttämättä koko hakuavaruudessa.
  - algoritmi ei tutki koko avaruutta, vaan valikoi tutkittavan alueen heuristisesti
  - algoritmi voi “juuttua” lokaaliin optimiin

Heuristiikka

- Intuitiivisesti: apumenetelmä tai periaate, jota soveltamalla ongelma ratkeaa nopeammin mutta tulokset silti melko hyviä/useimmiten hyviä.
- yleensä ovat “alioptimaalisia” (suboptimal) eli eivät takaa globaalia optimia (parhaan mallin löytymistä)
- Esim. **ahne heuristiikka** (greedy heuristic): Edetään haussa sillä hetkellä parhaalta näyttävään suuntaan.
  - Tyypillistä ihmisille! Tyyliin “Väsyttää ja masentaa, hirveästi rästitöitä, paras pitää lökeripäivä.” (optimoi mielihyvää lyhyellä tähtäimellä)
- joskus heuristiikan virheellisyys merkitsee vain hidasta hakua, ei oikeiden tulosten hukkaamista (riippuu mihin heuristiikkaa käytetään)

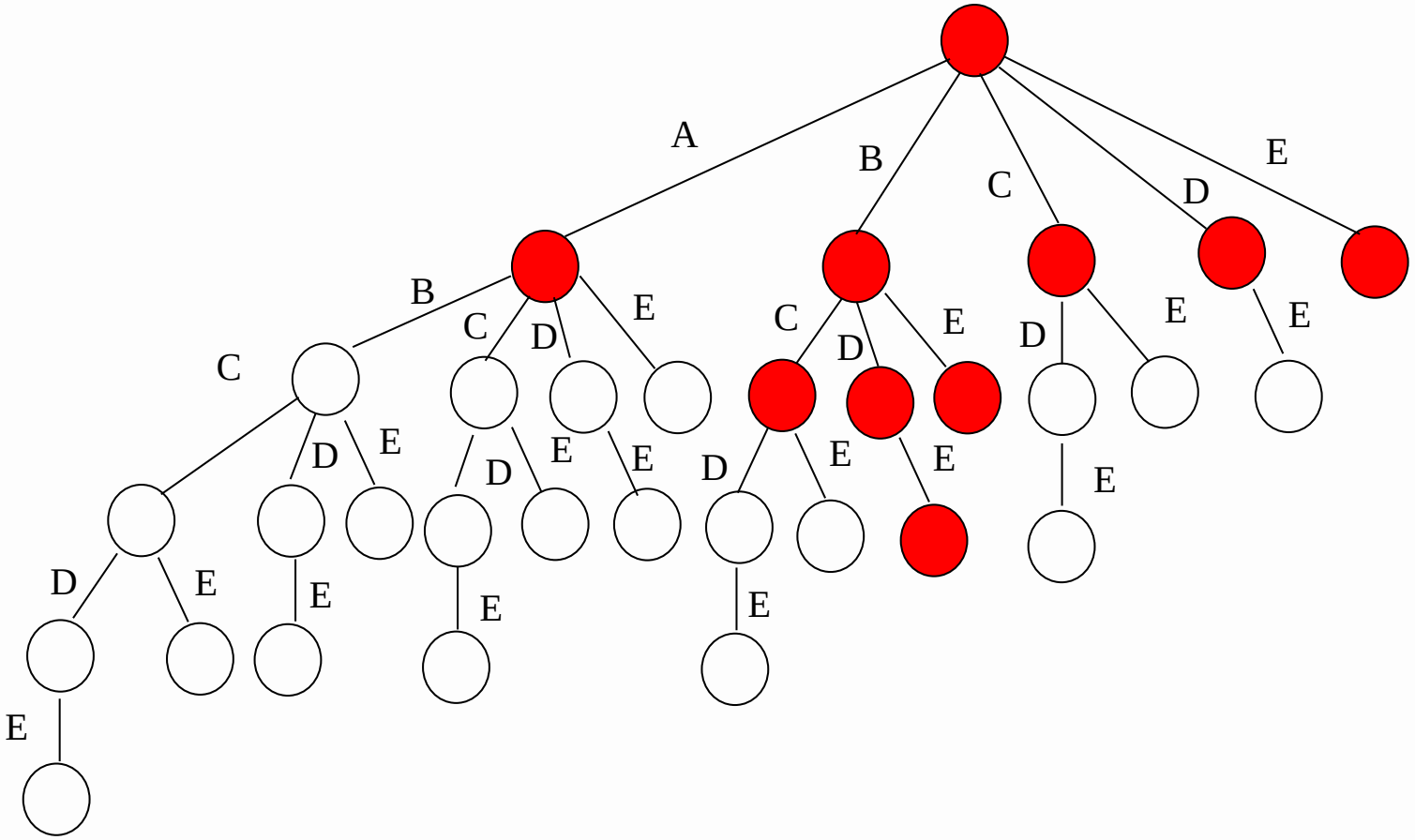
Esimerkki: Luokittelusääntöjen hakuavaruus

Olk. $R = \{A, B, C, D, E\}$ ja luokka-attribuutti F. Hakuavaruus $\mathcal{P}(R)$ voidaan esittää luettelointipuuna (enumeration tree).



Parhaan säännön ahne haku

Jos tasoittain haku ylhäältä alas ja ahne haku tuottaisi $BDE \rightarrow F$:n. Tutkisi vain 9 solmua 32:sta, mutta voi mennä metsään!



Varoituksen sana!

Aina menetelmää/sen hakualgoritmia valitessasi

- Epäile alioptimaalista heuristiikkaa!
- Ota selvää, mikä heuristiikka on ja milloin se voi hukata parhaat mallit.
- Miten todennäköistä on, että saadaan huonoja tuloksia?
- Hälytyskellojen paras soida, jos kuulet sanan “ahne heuristiikka” (vaikka joskus harvoin turha hälytys)
- Ongelma: Menetelmien kehittäjät eivät mielellään kerro, ettei menetelmä ole globaalisti optimaalinen...

Mallin ylisovittuminen (overfitting)

Intuitiivisesti: Malli/hahmo h kuvaa mallinnettavan datan d liian yksityiskohtaisesti, mukaan lukien satunnaisvaihtelun, eikä siksi yleisty d:n ulkopuolelle.

Täsmällisemmin: h ylisovittunut, jos $\exists h'$ s.e. $M(h, d) > M(h', d)$ mutta $M_{true}(h) < M_{true}(h')$, missä M_{true} on mallin "oikea" hyvyys.

- Oikeaa hyvyyttä M_{true} ei voi mitata! Koetettava arvioida heuristisesti.
- Oletus: M hyvyyden mukaan kasvava (jos ei, niin vaihda $>$ ja $<$)

Occamin partaveitsi –periaate

Tärkeä havainto: Ylisovittuminen edellyttää monimutkaista mallia.

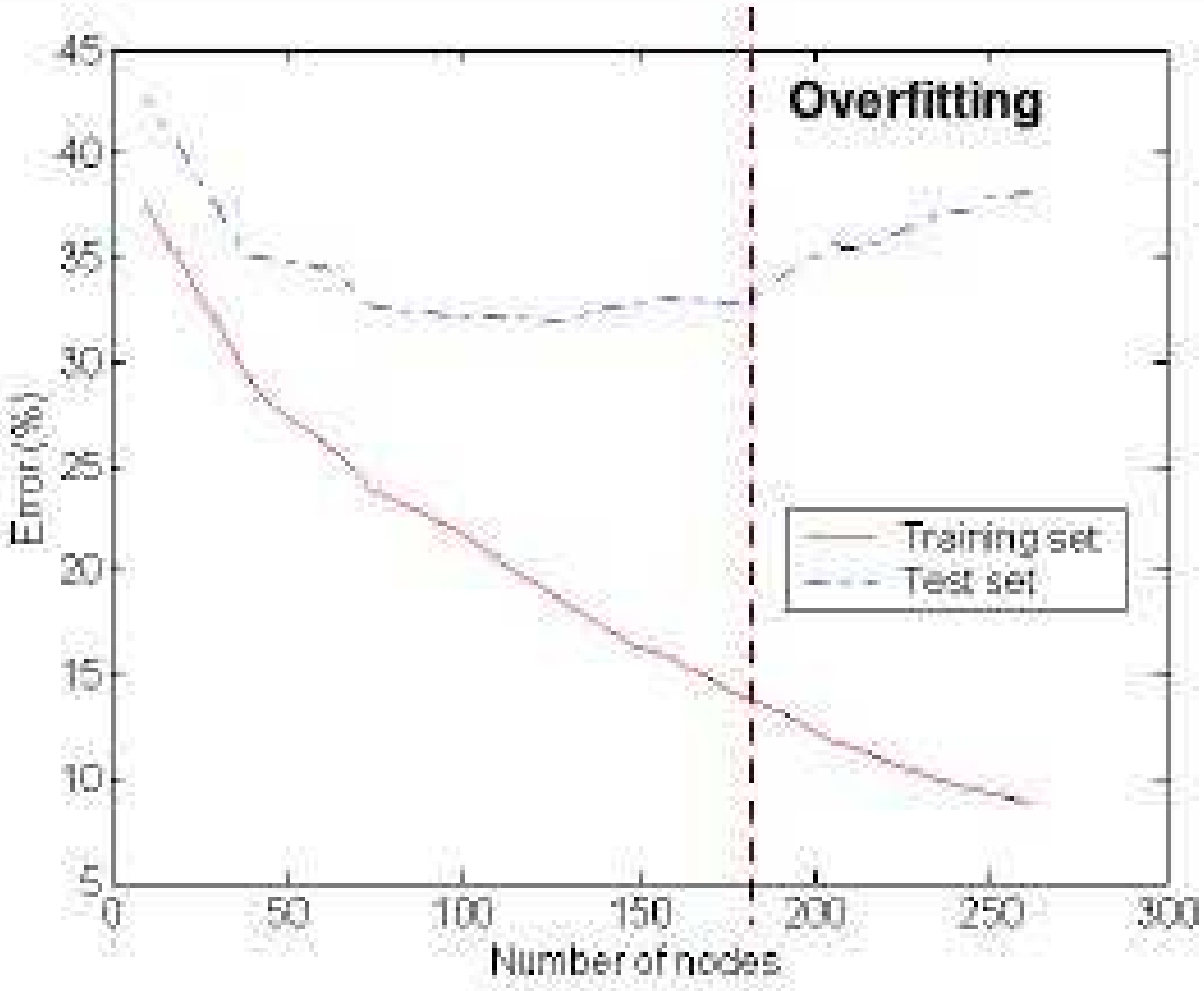
- monimutkaisella mallilla voi kuvata yksityiskohdat → tarkka kuvaus mutta harvoille (tai vain yhdelle) datajoukoille
- yksinkertaisella mallilla ei voi kuvata kaikkia yksityiskohtia → kuvaa ylimalkaisemmin mutta useampia datajoukkoja
- ⇒ ylisovittumista voi estää kontrolloimalla mallin monimutkaisuutta!
- = Occamin partaveitsi (Occam's Razor) –periaate: (Yhtä hyvistä) malleista suosi yksinkertaisinta.
- tai rankaise monimutkaisuudesta

Alisovittuminen (underfitting)

Intuitio: Jos malli on liian yksinkertainen, se kuvaa minkä tahansa datan liian ylimalkaisesti.

- Helppo havaita! Hyvyysmitta saa huonon arvon myös opetusdatassa.
- Huom! Jos data oikeasti satunnaista, ei siitä voi muodostaa parempaa mallia!

Esimerkki: Päätöspuun ylisovittuminen



Mallin monimutkaisuus suhteutettava datan määrään!

Perusperiaate: mitä enemmän datarivejä, sitä monimutkaisemman mallin voi muodostaa.

- Peukalosääntö: oltava vähintään 5–10 datariviä jokaista malliparametria kohden
- Jos jokin hahmo esiintyy < 5 rivillä, ei mitään takeita, etteikö sattumaa
- $\Rightarrow$ Jos vähän dataa, käytettävä yksinkertaisia mallinnusmenetelmiä ja etsittävä yksinkertaisia malleja.

Mallin monimutkaisuus vs. datan määrä

- Poikkeus: tukivektorikoneet (support vector machines, SVM) hyviä välttämään ylisovittumisen, ainakin teoriassa (käytännössä valittava malliparametrit huolella!)

Tilastollisten riippuvuuksien analyysi

- Tilastollisen riippuuden määritelmä/tulkinnat
- Riippuvuussääntöjen haku kategorisesta (tai diskretoidusta) datasta
- Riippuvuuksien haku numeerisesta datasta
- Ekskursio: Bayesläinen riippuvuuksien mallinnus (Bayes-verkoilla)

Tilastollinen riippuvuus: Monta tulkintaa!

Määritelmä: Tapahtumat X ja Y ovat tilastollisesti **riippumattomia**, jos $P(XY) = P(X)P(Y)$.

Mutta: Milloin muuttujat tai muuttuja-arvokombinaatiot ovat tilastollisesti **riippuvia**?

(termi "**korrelaatio**" viittaa yleensä kahden numeerisen muuttujan väliseen lineaariseen riippuvuuteen, mutta käytetään väljästi)

Tilastollinen riippuvuus: Monta tulkintaa!

- Tilastollinen riippuvuus edellyttää siis että $P(XY) \neq P(X)P(Y)$. Mutta: pieni poikkeama $P(X)P(Y)$:stä johtuu todennäköisesti sattumasta! Miten arvioida, onko riippuvuus aitoa? $\rightarrow$ Mitat riippuvuuden voimakkuuden ja merkitsevyyden arviointiin.
- Jos A ja B binäärimuuttujia, riittää tutkia yhtä arvokombinaatiota: $P(A = 1B = 1) \neq P(A = 1)P(B = 1) \Leftrightarrow \forall a \in \{0, 1\} \forall b \in \{0, 1\} P(A = aB = b) \neq P(A = a)P(B = b)$. Entä jos A ja B moniarvoisia muuttujia? $\rightarrow$ muuttuja- ja arvoperustaiset tulkinnat
- Miten määritellä kolmen muuttujan, A, B ja C, välinen riippuvuus? $\rightarrow$ ainakin 3 vaihtoehtoista tulkintaa.

Tilastollinen riippuvuus: 3 tulkintaa

Olk. A, B, C binäärisiä. Merk. $\neg A \equiv (A = 0)$ ja $A \equiv (A = 1)$

1. Riippuvuussääntö $AB \to C$: oltava $\delta = P(ABC) - P(AB)P(C) > 0$ ja δ riittävän suuri.
2. Kokonaistodennäköisyysmalli: Lasketaan $\delta_1 = P(ABC) - P(AB)P(C)$, $\delta_2 = P(A\neg BC) - P(A\neg B)P(C)$, $\delta_3 = P(\neg ABC) - P(\neg AB)P(C)$ ja $\delta_4 = P(\neg A\neg BC) - P(\neg A\neg B)P(C)$. Jos $\delta_1 = \delta_2 = \delta_3 = \delta_4 = 0$, ei ainakaan riippuvuutta. Muuten päätellään jollain laskukaavalla d_i:stä ($i = 1, \ldots, 4$)
3. ABC on “korreloiva joukko” jos $\rho = P(ABC) - P(A)P(B)P(C) > 0$ ja ρ riittävän suuri. (Yl. olt. $\rho > 0$ eli positiivinen riippuvuus)

1. Riippuvuussäännöt

Yleisessä muodossa $X \to A = a$, $X \subsetneq R$, $A \in R \setminus X$ ja $a \in \{0, 1\}$.

- Yleensä ei siis sallita negaatioita ehto-osassa X (ei esim. $A_1 \neg A_3 \to A_5$. Helppo kiertää: luo uudet binäärimuuttujat negaatioille! (esim. $A_i = B_{2i}$ ja $\neg A_i = B_{2i+1}$)
- $X \to A_i$ (tai $X \to A_i = 1$) merkitsee positiivista riippuvuutta ($P(XA_i) > P(X)P(A_i)$)
- $X \to \neg A_i$ (tai $X \to A_i = 0$) merkitsee negatiivista riippuvuutta X:n ja A_i:n välillä ($P(XA_i) < P(X)P(A_i)$ eli $P(X\neg A_i) > P(X)P(\neg A_i)$)

1. Riippuvuussäännöt: Esimerkkejä

korkea LDL → ateroskleroosi
korkea HDL → ¬ ateroskleroosi
tupakoi → ateroskleroosi
paljon tupakkaa → alzheimer
kahvi, ei tupakoi → ateroskleroosi
kohtuudella alkoholia → ¬ sydänsairaus
miestyypin kaljuus -> sydänsairaus
ApoE-e4 -> aivoinfarkti
ApoE-e4, kohtuudella alkoholia → alzheimer
ApoE-e4, kohtuudella tupakkaa → ¬ alzheimer
ABCA1-R219K → ¬ alzheimer
ABCA1-R219K, nainen → alzheimer

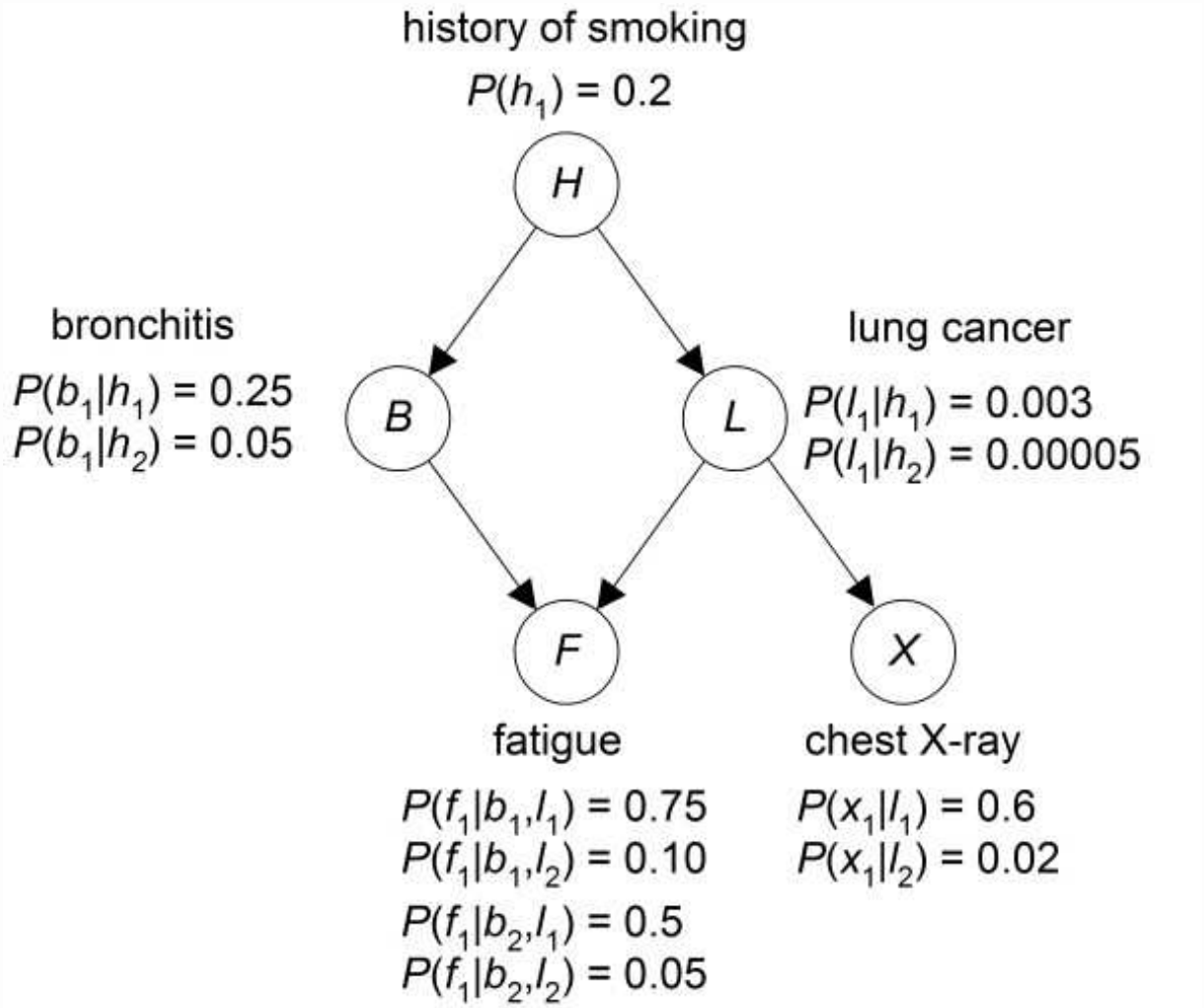
1. Riippuvuussäännöt

- lokaaleja hahmoja, eivät globaaleja malleja
- erittäin tehokkaat algoritmit! (silti globaalisti optimaalisia)
- tuhansille tai jopa kymmenille tuhansille binäärimuuttujille (riippuu datan jakaumasta, miten tehokkaita)
- ei-binäärimuuttujat edellyttävät binärisointia

2. Kokonaistodennäköisyysmalli

- esittävät kokonaistodennäköisyysjakauman
  $P(A_1 = a_1 \ldots A_k = a_k), \forall a_i \in \{0, 1\}$
- globaaleja malleja
- **paljon** raskaampia oppia $\rightarrow$ esivalittava joukko hyviä muuttujia, silti vain lokaalisti optimaalisia
- voi esittää havainnollisesti Bayes-verkkona

2. Kokonaistodennäköisyysmalli: Esimerkki



Lähde: Jiang. et al. Learning genetic epistasis using Bayesian network scoring criteria, Bioinformatics 12:89 2011.

3. Korreloiva joukko

Yleisessä muodossa $X \subseteq R$, missä $P(X) > \Pi_{A_i \in X} P(A_i)$

- kätevä kun ei selkeää säännön seurausosaa
- esim. esiintymisdata (ocurrence data)
  - Data: kasvilistoja eri havaintopaikoista. Hahmo: lajit jotka esiintyvät poikkeuksellisen usein yhdessä.
- monia heuristisia hahmoja ja niille hakualgoritmeja, mutta vähän hyviä (tilastollisesti mielekkäitä)
- Älä sekoita säännöllisiin joukkoihin (frequent sets)! Koostuu riittävän usein yhdessä esiintyvistä attribuuteista, joiden välillä ei välttämättä tilastollista riippuvuutta.

Luento 4

- Riippuvuussäännöt (tarkemmin)
- Riippuvuuksien haku numeerisesta datasta
- Ekskursiona Bayes-verkot

Riippuvuussäännöt

- Muuttuja- ja arvoperustainen tulkinta
- Riippuvuuden vahvuuden ja merkityksellisyyden mittaus
- Säännön redundanssi
- Tilastollisen riippuvuuden epämonotonisuus ja Simpsonin paradoksi
- Ei-binääristen muuttujien käsittely
- Varoitus assosiaatiosäännöistä

Muuttuja- vs. arvoperustainen tulkinta

Sääntö $X \rightarrow A$. (Merk. myös vastaavia muuttujia X:llä ja A:lla.)

1. Muuttujaperustainen (variable-based) tulkinta: **binäärimuuttujien** X ja A välillä on vahva positiivinen riippuvuus.
   - Sama asia kuin $\neg X \rightarrow \neg A$
   - Yhtä hyviä kuin negatiivinen riippuvuus X:n ja $\neg A$:n välillä (tai $\neg X$:n ja A:n)
2. Arvoperustainen (value-based): **arvokombinaatioiden** $X = 1$ ja $A = 1$ välillä on vahva positiivinen riippuvuus.
   - eri asia kuin $\neg X \rightarrow \neg A$, joka voi olla heikko!

Riippuvuuden vahvuus: Tavallisimmat mitat

Olk. $X \rightarrow A$

1. Muuttujaperustainen: Leverage

$$\delta(X, A) = P(XA) - P(X)P(A)$$

2. Arvoperustainen: Lift

$$\gamma(X, A) = \frac{P(XA)}{P(X)P(A)} = \frac{P(A|X)}{P(A)}$$

$P(A|X)$ on säännön “konfidenssi”

Huom! Kontingenssitaulu

	A	$\neg A$	kaikki
X	$fr(XA) = n[P(X)P(A) + \delta]$	$fr(X\neg A) = n[P(X)P(\neg A) - \delta]$	$fr(X)$
$\neg X$	$fr(\neg XA) = n[P(\neg X)P(A) - \delta]$	$fr(\neg X\neg A) = n[P\neg(X)P(\neg A) + \delta]$	$fr(\neg X)$
kaikki	$fr(A)$	$fr(\neg A)$	n

Kaikkien arvokombinaatioiden välillä siis sama $|\delta|$!
$\Leftrightarrow \gamma$ riippuu arvokombinaatiosta

Esim: Milloin arvoperustaisesta voisi olla iloa?

Olk. X=alleelikombinaatio, D=sairaus ja $P(X)$ pieni.

$P(D|X) = 1.0$ eli $\gamma(X, D) = P(D)^{-1}$ voi olla iso
$P(\neg D|X) = 0$ eli $\gamma(X, \neg D) = 0$
$P(D|\neg X) \approx P(D)$ eli $\gamma(\neg X, D) \approx 1$
$P(\neg D|\neg X) \approx P(\neg D)$ eli $\gamma(\neg X, \neg D) \approx 1$

$\leftrightarrow \delta(X, D) = P(X)P(\neg D)$ pieni.

Ts. riippuvuus olisi vahva arvoperustaisessa tulkinnassa, mutta heikko muuttujaperustaisessa.

(Yleensä **muuttuja**perustaiset riippuvuudet “kestävämpiä”)

Merkitsevyysmitat

- Riippuvuuden vahvuus ei kerro kaikkea! Voisi johtua sattumasta $\Rightarrow$ arvioitava tilastollinen merkitsevyys
- Idea: Oletetaan että X ja A riippumattomia. Mikä on todennäköisyys p, että olisimme silloin havainneet vähintään näin vahvan riippuvuuden?
- Jos p on pieni, riippuvuus tuskin johtui sattumasta.
- Tarkka p luotettava mitta varsinkin muuttujaperustaisille säännöille! (Riippuvuudet pätevät suurin piirtein yhtä vahvoina myös tulevaisuudessa.)
- Muita tunnettuja hyvyysmittoja: χ^2 ja mutual information (MI) + paljon heuristisia mittoja!

Esimerkkejä mitoista

$$p_F = \frac{\sum_{i=0}^{\infty} \binom{fr(X)}{fr(XA)+i} \binom{fr(\neg X)}{fr(\neg X \neg A)+i}}{\binom{n}{fr(A)}}$$

$$MI = \log \frac{P(XA)^{P(XA)} P(X\neg A)^{P(X\neg A)} P(\neg XA)^{P(\neg XA)} P(\neg X\neg A)^{P(\neg X\neg A)}}{P(X)^{P(X)} P(\neg X)^{P(\neg X)} P(A)^{P(A)} P(\neg A)^{P(\neg A)}}$$

$$\chi^2 = \frac{n(P(XA) - P(X)P(A))^2}{P(X)P(\neg X)P(A)P(\neg A)}$$

Kannattaa suosia p_F:ää!

Säännön redundanssi

Intuitio: Spesifimpi sääntö $XQ \rightarrow A$ on redundantti, jos se ei lisää mitään uutta informaatiota yleisempään sääntöön $X \rightarrow A$.

esim. Jos tunnetaan $stressi \rightarrow verenpaine$,
$stressi, suklaa \rightarrow verenpaine$ lisää tuskin informaatiota.

- Arvoperustaiset säännöt: Oltava $P(A|XQ) > P(A|X)$, jotta $XQ \rightarrow A$ olisi ei-redundantti. Lisäksi konfidenssin parannuksen oltava tilastollisesti merkitsevä

Säännön redundanssi

- Muuttujaperustaiset säännöt: Huomioitava myös $P(\neg A|\neg(XQ))$ vs. $P(\neg A|\neg X)$.
  - Jos $P(A|XQ) > P(A|X)$, on usein $P(\neg A|\neg(XQ)) < P(\neg A|\neg X)$.
  - Punnittava näitä: Onko $P(A|XQ)$:n tuottama parannus merkitsevämpi kuin $P(\neg A|\neg X)$:n tuottama?
  - Käytännössä selviää tutkimalla onko $M(XQ \rightarrow A) > M(X \rightarrow A)$ (kunhan M fiksu mitta)

Analogia: Suuret makeat omenat

Tarjolla on vain punaisia ja virheitä omenoita, joista osa suuria ja osa pieniä.
Tunnetaan omenoiden makeutta indikoiva sääntö $punainen \rightarrow makea$.
Milloin $punainen, suuri \rightarrow makea$ on ei-redundantti?

- “suuruuden” lisättävä makeuden todennäköisyyttä
- Entä jos pienet punaiset omenat ovat toisinaan makeita?
- toimii, jos pieniä punaisia makeita omenoita tosi vähän tai pienet punaiset ovat happamia

Tilastollisen riippuvuuden epämonotonisuus

Määr. Ominaisuus O on monotoninen (osajoukkorelaatiolle), mikäli kaikilla $Z \supseteq X$ pätee $O(X) \Rightarrow O(Z)$ ja antimonotoninen, mikäli kaikilla $X \subseteq Z$ pätee $O(Z) \Rightarrow O(X)$. Muuten O on epämonotoninen.

- esim. ominaisuus $fr(Y) \geq min_{fr}$ on antimonotoninen
- ominaisuus $fr(YC) = 0$ on monotoninen
- Hakua helpottavia ominaisuuksia!
- Mutta: Kumpikaan ei päde tilastolliselle riippuvuudelle...

Epämonotonisuus: Merkitys

- $AB \to C$ voi ilmaista merkitsevää riippuvuutta, vaikka A ja C sekä B ja C riippumattomia toisistaan.
- Pahimmassa tapauksessa datajoukon ainut riippuvuus koskee kaikkia attribuutteja $A_1 \ldots A_k$ (esim. $A_1 \ldots A_{k-1} \to A_k$)

$\Rightarrow$ 1) Ahne heuristiikka ei toimi!

$\Rightarrow$ 2) Kahden välisten riippuvuuksien tutkiminen ei paljasta kaikkea!

Simpsonin paradoksi

ABCA1-R219K $\rightarrow$ ¬alzheimer
ABCA1-R219K, nainen $\rightarrow$ alzheimer

(Jopa ABCA1-R219K $\rightarrow$ alzheimer tai ABCA1-R219K $\perp\!\!\!\perp$ alzheimer)

> Positiivinen riippuvuus muuttuu negatiiviseksi, kun sääntöä spesifoidaan, tai päinvastoin.

Riippuvuuden suunta riippuu siitä, minkä osaryhmän edustajia datassa oli!

$\Rightarrow$ Lääkkeeksi edustava data ja riittävän monimutkaisien riippuvuuksien analyysi.

Riippuvuussääntöjen rajoitukset

- perusmuodossa vain binääridatalle
- yleensä ei sallita negaatioita ehto-osassa → luotava uudet attr. negaatioille
- monet ohjelmat eivät salli edes seurausosassa (Kingfisher poikkeus)
- soveltuvat sellaisenaan esiintymisdatan analyysiin!
  - Kasvi- ja eläinlajien esiintymisdata
  - Geenien (alleelien) esiintymisdata
  - Ostoskoridata

Ei-binääristen muuttujien käsittely

Perusstrategia:

- Kategorisen muuttujan kaikille arvoille luodaan uudet binääriattribuutit
  - esim. 5 pyyntipaikkaa $\rightarrow$ luodaan binääriattribuutit $paikka1, \ldots, paikka5$
  - ei hukkaa mitään riippuvuuksia!
- Numeeriset muuttujat diskretoidaan etukäteen
  - Saattaa hukata riippuvuuksia – ei takaa enää globaalia optimia!
  - Ongelma: Eri diskretointi voi olla optimaalinen eri riippuvuuksissa $\rightarrow$ olisi parempi diskretoida dynaamisesti, haun aikana (onnistuu, mutta nykymenetelmillä tehotonta)

Lisää attribuuttien muodostusvinkkejä

Hämäläinen: Thorough analysis of log data with dependency rules: Practical solutions and theoretical challenges. (kannattaa tutkia!)

`http://www.cs.uef.fi/~whamalai/research.html`

Ja algoritmeista kiinnostuneille (hyvin yleistajuisesti):

Hämäläinen: Globaali optimonti haastaa lokaalit menetelmät: luotettavampia riippuvuuksia tehosta tinkimättä.

Varoituksen sana: Assosiaatiosäännöt

- Esitetään usein samannäköisesti $X \rightarrow A$ tai $X \Rightarrow A$
- tilastotieteessä “assosiaatio”=kategoristen muuttujien välinen riippuvuus
- Tiedonlouhinnassa ei yleensä merkitse sitä!
- X ja A voivat olla tilastollisesti riippumattomia tai voi olla $P(XA) < P(X)P(A)$ (vaikka käyttäjä luulee päinvastaista)
- Assosiaatiosääntö kertoo vain, että kombinaatio XA on riittävän frekventti, ts. $P(XA) > min_{fr}$ jollain raja-arvolla $min_{fr} \in [0, 1]$

Varoituksen sana: Assosiaatiosäännöt

- Periaatteessa voi karsia jälkikäteen, jolloin jäljelle jäävät tilastollisilla mitoilla parhaat frekventit säännöt
- Ei silti korvaa haun aikana hukattuja sääntöjä!
- algoritmit yleensä niin tehottomia, että oltava suuri min_{fr} (jopa 0.6)
- jos tehokkaampia, käyttävät muita järjettömiä heuristiikkoja (esim. suljettujen joukkojen heuristiikka suosii monimutkaisia ylisovittuneita sääntöjä)
- Poikkeus: Geoff Webbin MagnumOpus etsii tilastollisilla mitoilla ja toimii jopa ilman minimifrekvenssiä

Jos käytät assosiaatiosääntöjä

- tarkista huolella, millaisia hahmoja tuottavat!
- parhaimmillaan $\approx$ riippuvuussääntöjä, pahimmillaan vain harhaanjohtavia
- Luokittelusäännöt (classification rules), joissa seurausosa luokka-attribuutti, perustuvat yleensä tilastolliseen riippuvuuteen
  - tutki silti onko lisäkriteerejä tai vaatiiko min_{fr}:ää
- Pieni min_{fr} (esim. $min_{fr} = \frac{5}{n}$) ei haittaa!

Numeeristen (ja ordinaalisten) muuttujien riippuvuudet

- Kahden muuttujan riippuvuudet
  - systemaattinen haku onnistuu
- Monen muuttujan riippuvuudet
  - ML-menetelmät: yhden–muutaman mallin oppiminen

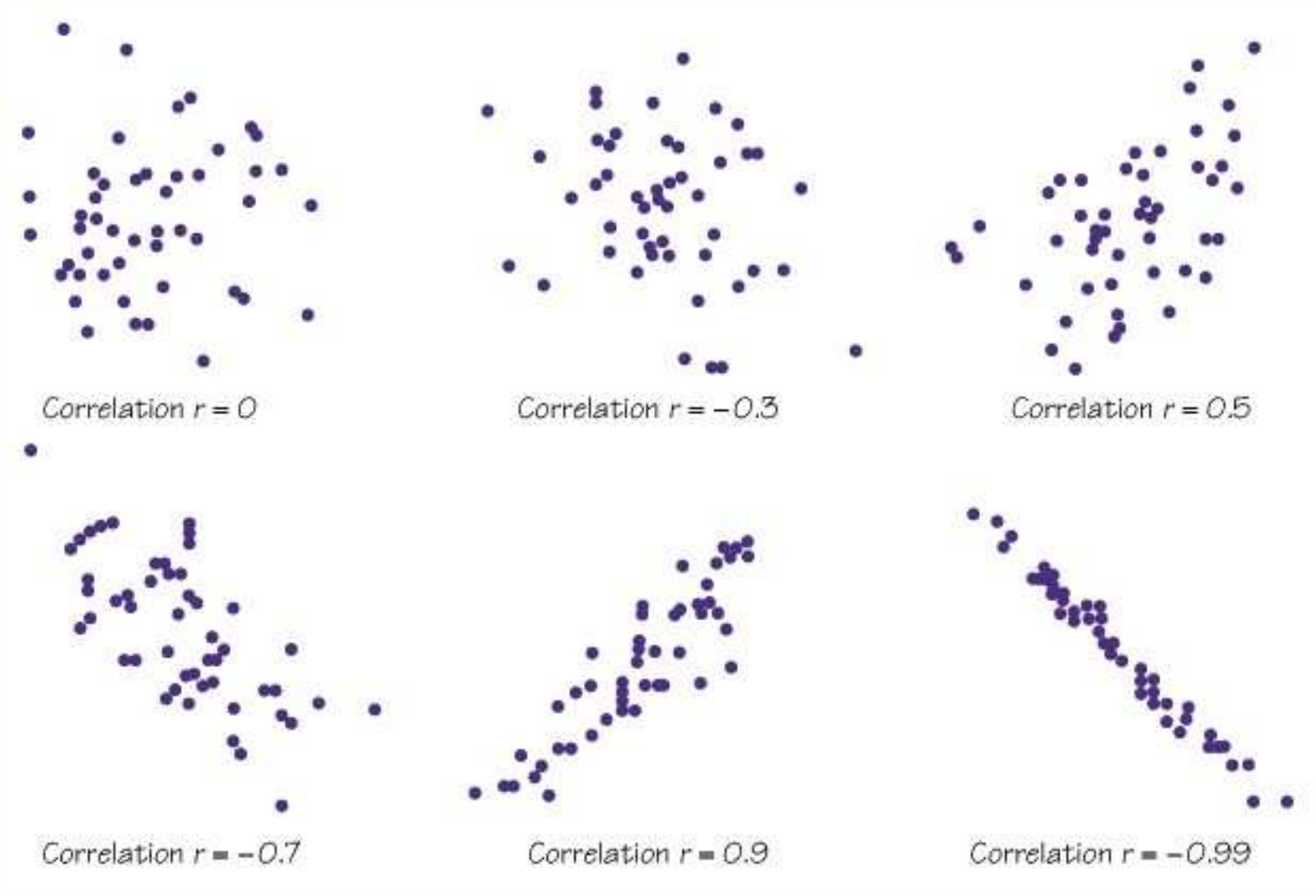
Muistutus: ordinaalisen muuttujan arvoilla järjestys, mutta eivät “oikeita” numeroita.
Esim. vatsahaavalla 4 arvoa: puuttuu, lievä, keskivaikea, vaikea.

Pearsonin korrelaatiokerroin

$$r(X,Y) = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

- numeerisille muuttujille X ja Y
- paljastaa lineaarisen riippuvuuden (pos. tai neg.)
- jos X ja Y riippumattomia, $r \approx 0$
- Voi olla riippuvuus, vaikka $r = 0$!
  - Jos X ja Y normaalijakautuneita, $r = 0$ merkitsee riippumattomuutta
- Varo: Hyvin herkkä kohinalle ja ulkopuolisille (outliers)!

Esimerkkejä



Lähde: `http://knottwiki.wikispaces.com/`

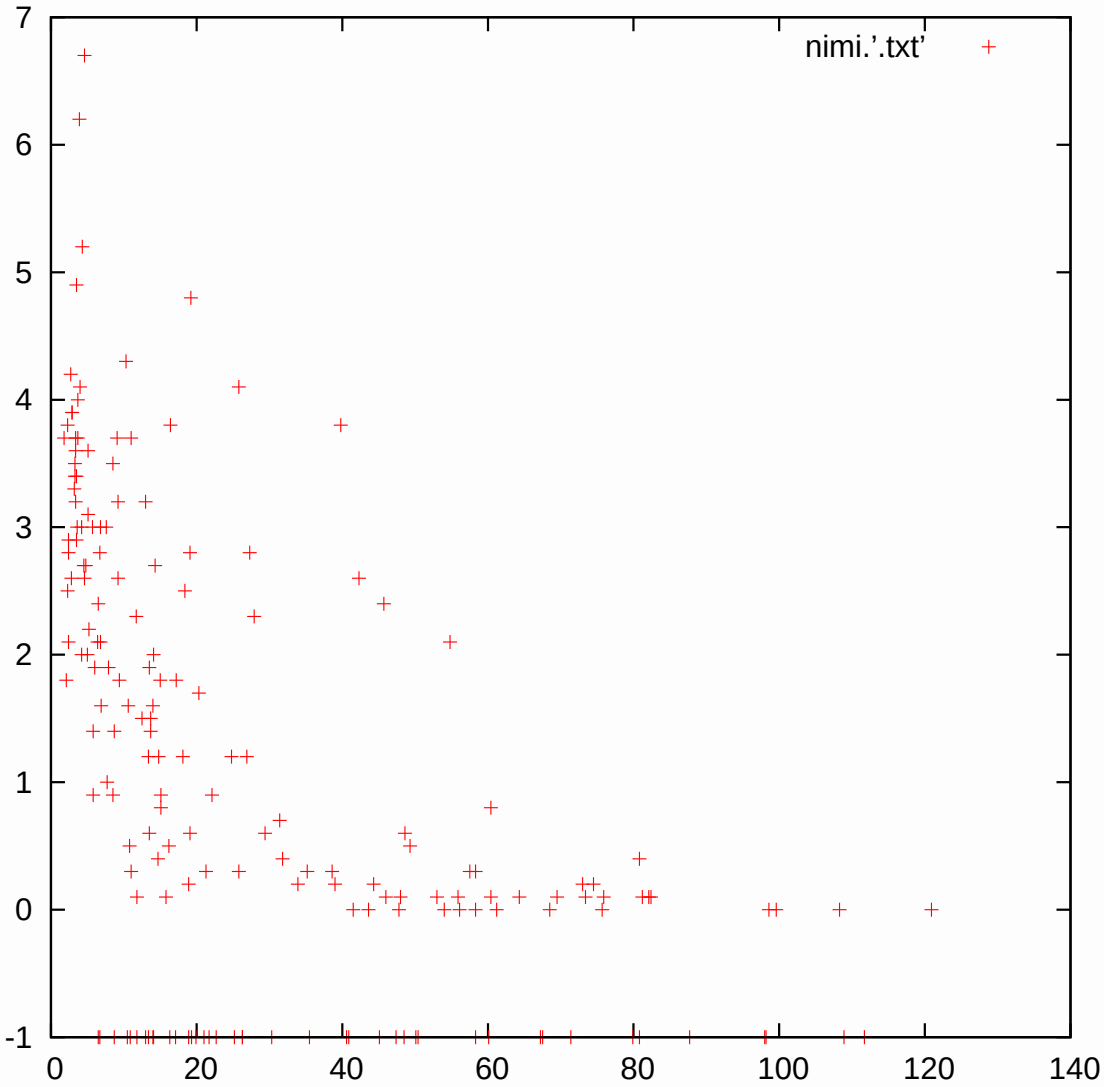
Järjestyskorrelaatiokertoimet (rank correlation coefficients)

- tunnetuimpia Kendallin τ ja Spearmanin ρ
- hyviä ordinaalimuuttujille!
- periaatteessa numeeristen muuttujien monotonisen riippuvuuden tunnistamiseen
- **Mutta:** varsinkin Spearman herkkä kohinalle/yksilölliselle vaihtelulle
- eivätkä osaa hyödyntää tietoa etäisyyksistä...

$\rightarrow$ pieni diskretointi (**moneen** osaväliin) voisi auttaa?

Esimerkki: Infant Mortality–Physicians

$r = -0.53$, $\tau = -0.36$, $\rho = 0.03$



Kahden muuttujan mv. riippuvuus

Ehdotettuja mittoja esim.
1) Korrelaatiosuhde (correlation ratio) $\eta(X, Y)$

- X diskretoidaan ja verrataan Y:n varianssia X:n osaväleillä vs. koko datassa
- jos lineaarinen riippuvuus, $\eta^2 = r^2$
- η – $|r|$ kertoo epälineaarisuuden asteesta
- Ongelma: riippuu X:n diskretoinnista! (Jos paljon osavälejä, $\eta = 1$)
- Huomaa: $\eta(X, Y) \neq \eta(Y, X)$

Kahden muuttujan mv. riippuvuus

2) Mutual information $MI(X, Y)$

- Diskretoidaan X ja Y K:hon yhtä suureen osaväliin
- Lasketaan sitten

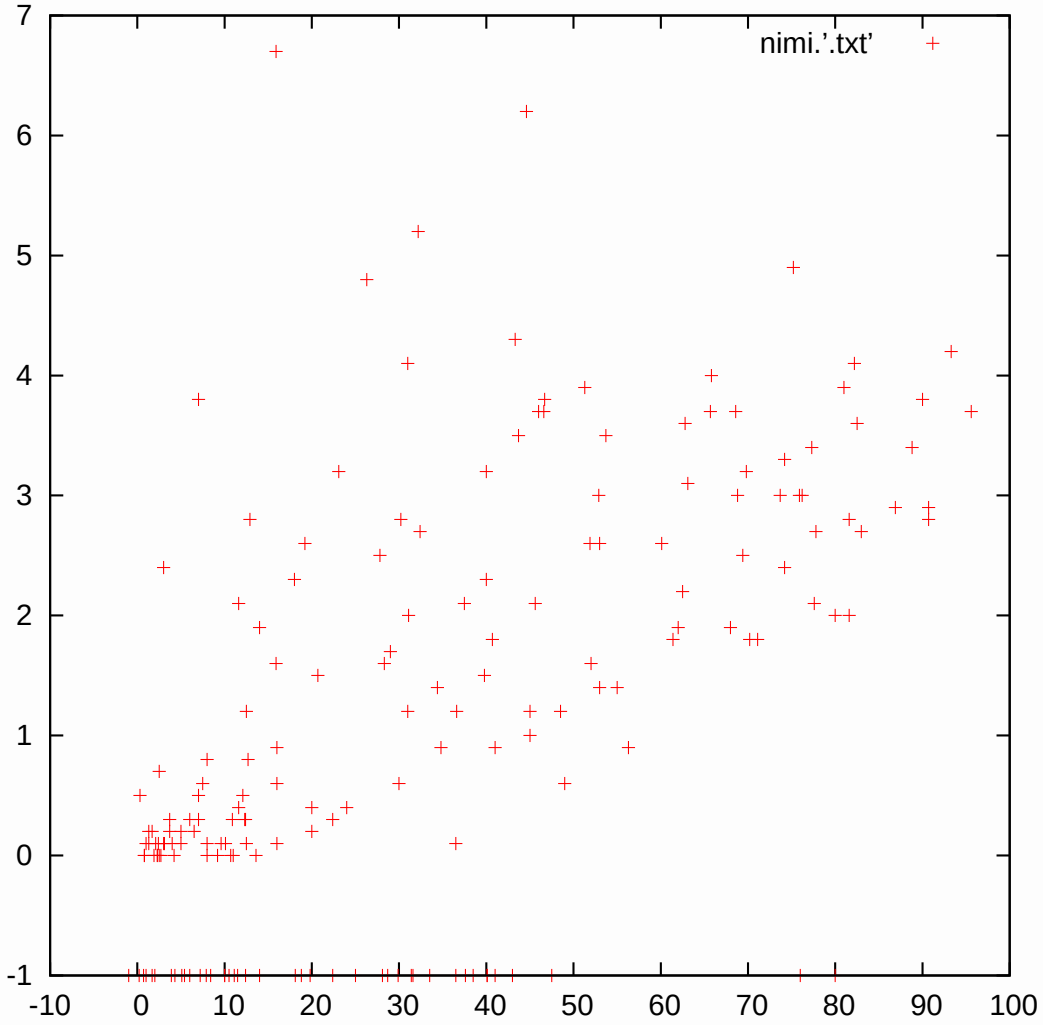
$$MI(X, Y) = \sum_{i=1}^{K} \sum_{j=1}^{K} P(X = x_i, Y = y_j) \log \frac{P(X = x_i, Y = y_j)}{P(X = x_i)P(Y = y_j)}$$

- + mahd. normalisointi

Pian päässemme testaamaan, miten hyvin toimii!

Joskus binarisointi+riippuvuussäännöt löytävät!

Internet users – Physicians



Riippuvuussäännöt

mostInternetUsers → **mostPhysicians**
fr=62 (0.3370), cf=0.816, M=-3.639e+01
leastInternetUsers → ¬ **mostPhysicians**
fr=84 (0.4565), cf=0.785, M=-3.499e+01
leastPhysicians → **leastInternetUsers**
fr=53 (0.2880), cf=0.930, M=-2.536e+01
mostInternetUsers → ¬ **leastPhysicians**
fr=72 (0.3913), cf=0.947, M=-2.473e+01

Monen muuttujan riippuvuudet: lineaariregressio (Multiple linear regression)

$$Y = \beta_1 X_1 + \ldots + \beta_k X_k + \alpha$$

- kuvaa Y:n **lineaarista** riippuvuutta muuttujista $X_1, \ldots, X_k$
- $X_1, \ldots, X_k$ selittävät muuttujat (explanatory variables)
- $\beta_1, \ldots, \beta_k$ regressiokertoimet (regression coefficients)
  - voivat kertoa myös muuttujien tärkeydestä
- koska yksinkertainen, on hyvin resistantti ylisovittumista vastaan!
  - Suosi erityisesti, jos vähän dataa

Rajoitukset

- riippuvuuden oltava lineaarinen, mutta pieni poikkeama ei vaarallista
  - Jos ennustava mallinnus, sietää suurtakin poikkeamaa!
- ongelma datan (suuret) aukot: onko lineaarinen aukkojen kohdalla?
- hyvin herkkä ulkopuolisille (outliers)!
- selittävien muuttujien oltava keskenään riippumattomia

Ongelma: Jos X_i:t korreloivat keskenään

- = “multicollinearity”
- β_i:t epätarkkoja–järjettömiä $\rightarrow$ eivät sovi ainakaan datan deskriptiiviseen mallinnukseen!
- mitä pienempi data ja enemmän kohinaa, sitä pahempi ongelma
- peukalosääntö: oltava $r(X_i, X_j) < \min\{r(X_i, Y), r(X_j, Y)\}$ $(\forall i, j)$
- periaatteessa korrelaatiot voi poistaa PCA:lla

Deskriptiivinen analyysi: Arvioi mallin validius!

1. Arvoi lineaarisuus laskemalla

$$r^2 = \frac{Var(Y) - MSE}{Var(Y)}$$

   - =multiple correlation coefficient
   - yleistää Pearsonin korrelaatiokertoimen monelle muuttujalle!
   - mitä enemmän $r^2 \to 1$, sitä lineaarisempi

2. Tutki residuaalit (virheet) $\epsilon_i = y_i - (\beta_1 x_{11} + \ldots + \beta_k x_{1k} + \alpha)$ (olt. riippumattomia, normaalijakautuneita, ka. 0 ja sama varianssi)
3. Tutki miten hyvin sopii dataan (F-testi)

Parhaan lineaariregressiomallin haku

- Valittava muuttujat X_i ja niiden parametrit β_i ja α
- yhden mallin sovitus dataan kevyttä $\mathcal{O}(n)$ (mutta muuttujien valinta pahimmillaan $\mathcal{O}(2^k)$)
- silti globaalisti optimaalisia menetelmiä vain pienidimensioiselle datalle $\rightarrow$ voisi varmasti kehittää tehokkaampiakin!
- valitettavan usein ahne haku
  - Aloitetaan yksinkertaisista malleista ja täydennetään aina parasta
    Tutki kaikki $\{Y = \beta_i X_i + \alpha_i\} \rightarrow$ esim. $\beta_3 X_3 + \alpha_3$ paras
    Tutki kaikki $\{Y = \beta_3 X_3 + \beta_i X_i + \alpha_i\}$ $(i \neq 3)$ jne.
  - tai aloita $Y = \beta_1 X_1 + \ldots \beta_k X_k + \alpha$:sta ja poista muuttujia 1 kerrallaan

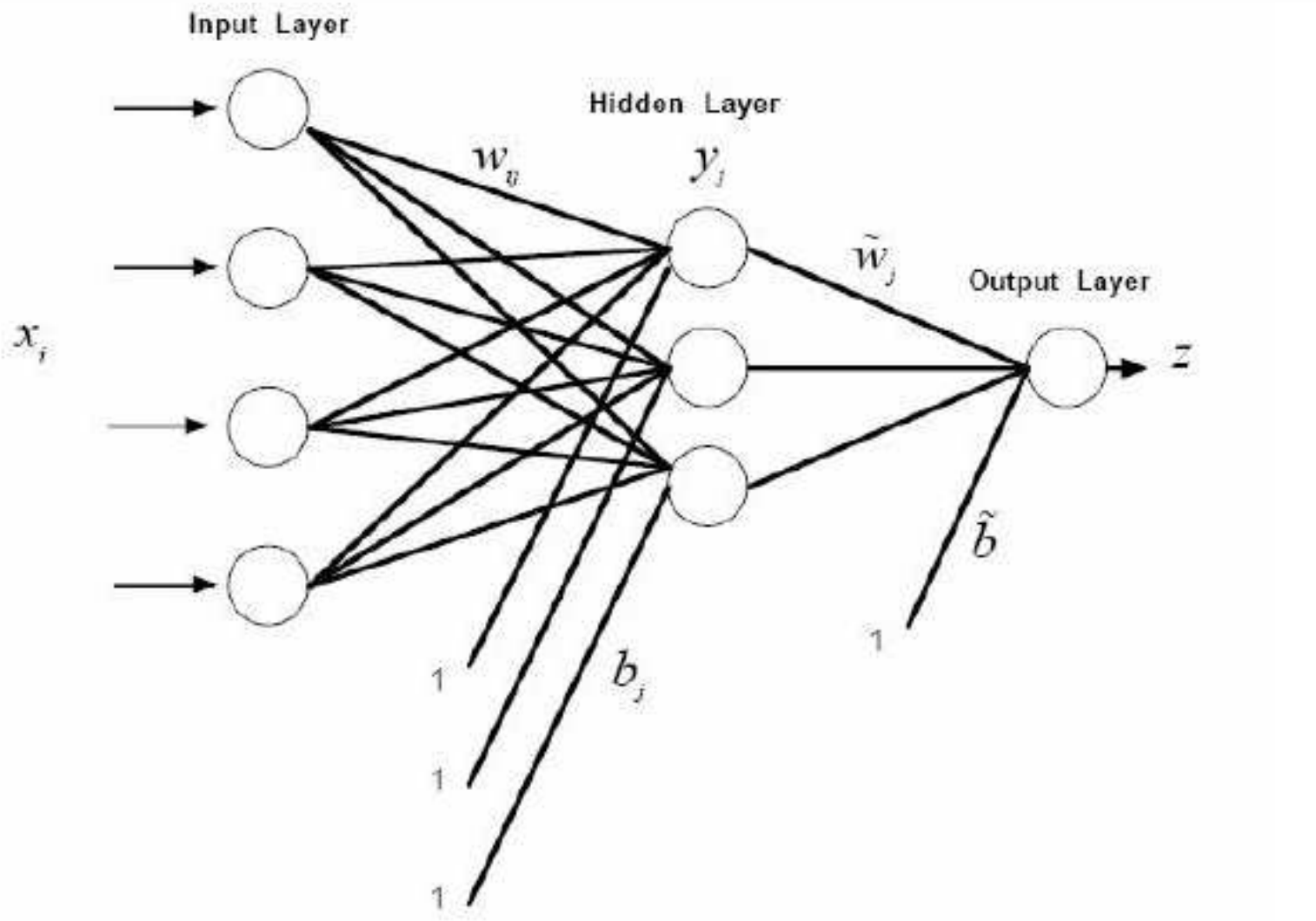
Laajennoksia

- $Y = \beta_1 f_1(X_1) + \ldots + \beta_k f_k(X_k) + \alpha$, missä f_i:t "pehemeitä" funktioita (esim. $\log$ tai $\sqrt{}$)
  - voi auttaa jos riippuvuus ei-lineaarinen
  - silti resistentti ylisovittumiselle!
  - hyvät funktiot löytyvät yleensä vain kokeilemalla
- joskus kannattaa muuntaa Y:tä, esim. $\log(Y)$ tai $\sqrt{}(Y)$
- logistinen regressio: Y on binäärinen tai ordinaalinen
- Y voi olla paloittain lineaarinen eri X_i:den yhdistelmien suhteen (piecewise linear) $\rightarrow$ malli kootaan yksinkertaisista lokaaleista hahmoista (splines)

Laajennoksia (jatkuu)

- korkeamman asteen termit, kuten $\beta_{i1}X_i$, $\beta_{i2}X_i^2$, $\beta_{i3}X_i^3$
- muuttujien vuorovaikutusta kuvaavat termit (intercation terms), kuten $\beta_{ij}X_iX_j$, $\beta_{ijl}X_iX_jX_l$, $\beta_{ijlm}X_iX_jX_lX_m$
- kaikki mahdollisia vuorovaikutustermejä yht. $2^k - 1$ kpl!
- monimutkaiset mallit, helposti ylisovittuvia!
- neuroverkoilla voi kuvata mielivaltaisia riippuvuuksia $Y = f(X_1, \ldots, X_k)$! (yleinen regressiomalli)

Yleinen regressio neuroverkolla



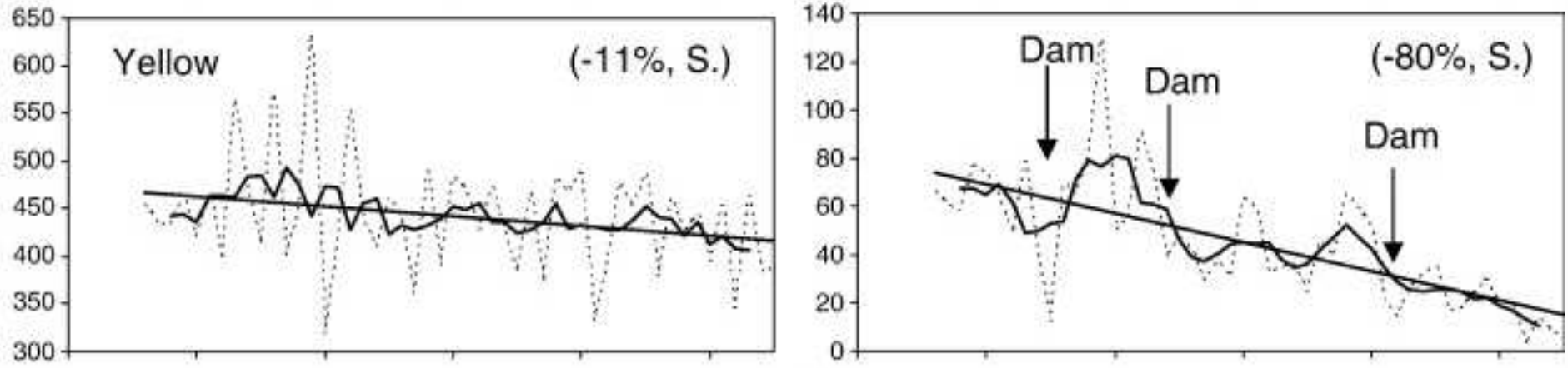
Lähde: `web.unbc.ca/~ytang/Chapter6.pdf`

Yleinen regressio neuroverkolla: Muista!

- ylisovittuu erittäin helposti!
- oppimisalgoritmi vaatii paljon ja edustavaa dataa jotta luotettavia tuloksia
- mutta ei jaksa käsitellä liian suuria datamääriä...
- eivät koskaan takaa globaalia optimia!
- Peukalosääntö: Valitse oletusarvoisesti vain 3-tasoinen verkko (vain 1 piilotaso)
  - riittää minkä tahansa funktion esittämiseen
  - auttaa ylisovittumista vastaan
- “Neuroverkko työkaluna on yhtä hyvä kuin käyttäjänsä”

Esimerkki: Trendien kuvaus lineaariregressiolla

Sademäärät ja joen vesimäärän muutokset Keltaisenjoen alueella 1945–2000



Lähde: Xu et al.: Temporal trend of precipitation and runoff in major Chinese Rivers since 1951, Global and Panetary Change 73, pp. 219–232, 2010

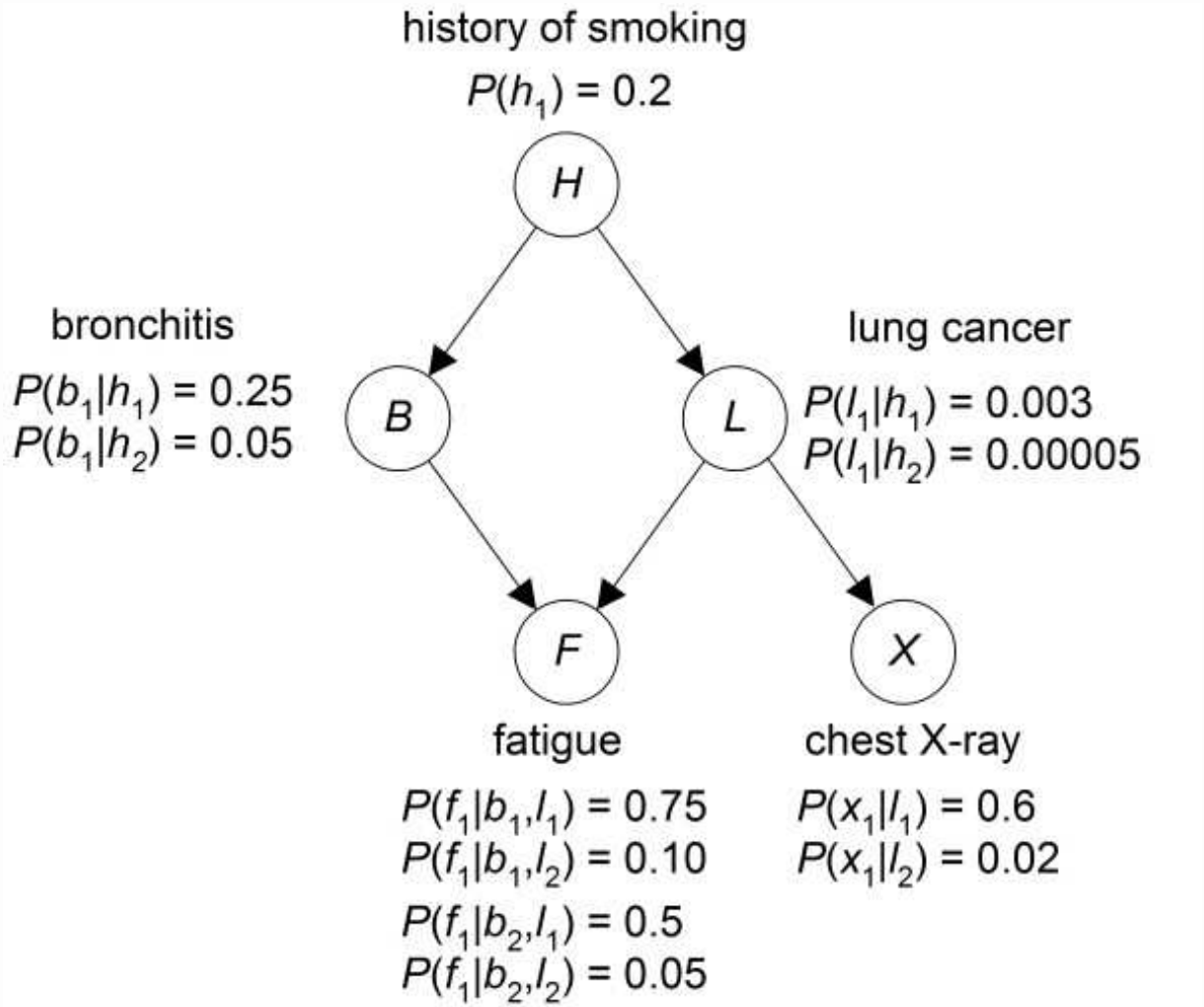
Ekskursio: Bayes-verkot

- esittää datan riippuvuusrakenteen ja implisiittisesti kokonaistodennäköisyysjakauman
- koostuu verkon rakenteesta ja solmujen todennäköisyystauluista
- Perusidea: solmut vastaavat muuttujia ja kaaret esittävät riippuvuuksia
- Jos kaksi muuttujajoukkoa X ja Y ehdollisesti riippumattomia annettuna kolmas Z, niin verkossa kaikki X:n ja Y:n solmut jonkin Z-solmun erottamia

Ekskursio: Bayes-verkot

- Juurisolmuille A_i annetaan prioritodennäköisyydet $P(A_i = a_i)$ (kaikilla A_i:n arvoilla a_i)
- Muille solmuille B_i ehdolliset todennäköisyydet $P(B_i = b_i | X = \overline{x})$ (X=solmun vanhemmat ja $\overline{x}$=niiden arvokombinaatio)

Esimerkki



Lähde: Jiang. et al. Learning genetic epistasis using Bayesian network scoring criteria, Bioinformatics 12:89 2011.

Ekskursio: Bayes-verkot

- Havainnollinen esitys riippuvuuksista (apua jopa kausaalisuhteiden keksimiseen)
- Rakenteen oppiminen laskennallisesti hankalaa!
- Ei siis takeita globaalista optimista (kuten ei yleensä)
- DM:ssä ei kuitenkaan järkeä määrittää rakennetta ad hoc

Luento 5: Luokittelu

- Määritelmä
- Käyttötarkoitukset
- Ongelmien ja luokittelijoiden tyypittely
  → auttaa valitsemaan oikean menetelmän
- Tärkeimmät menetelmät
  - Päätöspuut ja luokittelusäännöt
  - Bayes-luokittelijat
  - Lähimmän naapurin menetelmät (Nearest neighbour classifiers)
  - Neuroverkkoluokittelijat
  - Tukivektorikoneet (support vector machines, SVM)

Luokitteluongelma

Annettu joukko selittäviä muuttujia $R = \{A_1, \ldots, A_k\}$ ja luokkamuuttuja C, jonka arvot ovat mahdolliset luokat $Dom(C) = \{c_1, \ldots, c_L\}$.

Muodosta malli M joka määrittää jokaiselle datapisteelle $x_i \in \mathcal{D}$ sen luokan c_i.

- $\mathcal{D} = Dom(A_1) \times \ldots \times Dom(A_k)$
- Perustapauksessa malli on siis kuvaus
  $M : \mathcal{S} \rightarrow Dom(C)$
  - Olettaa että luokan määräytyminen determinististä! (voi olla epärealistista)
- Malli voi kertoa vain luokkajakauman $P(C = c_j | x_i)$

Luokittelun hyvyys

Luokittelun tulos		Oikeasti C	Oikeasti $\neg C$	Oikeasti Σ
	C	TP=true positive $fr(luokittelu = C \wedge oik = C)$	FP=false positive $fr(luokittelu = C \wedge oik = \neg C)$	TP+FP $fr(luokittelu = C)$
	$\neg C$	FN=false negative $fr(luokittelu = \neg C \wedge oik = C)$	TN=true negative $fr(luokittelu = \neg C \wedge oik = \neg C)$	FN+TN $fr(luokittelu = \neg C)$
	Σ	TP+FN $fr(oik = C)$	FP+TN $fr(oik = \neg C)$	n

$$error = \frac{FP + FN}{n} \qquad accuracy = \frac{TP + TN}{n}$$

Luokittelun hyvyys

$$precision = \frac{TP}{TP + FP} = P(oik = C|luokittelu = C)$$

$$sensitivity = \frac{TP}{TP + FN} = P(luokittelu = C|oik = C)$$

Kaikki lasketaan testidatasta! (tai ristiinvalidoinnilla)

Muista ylisovittuminen ja Occamin partaveitsi!

Käyttötarkoituksia

1. Luokittelu itsetarkoituksena
   esim. Mikä sairaus potilaalla on? Mitä lajia ötökkä on? Tulenko minä saamaan keuhkosyövän?
2. Hahmon tunnistus muun päättelyn avuksi (piirteitä korkeamman tason mallille)
   Mitä Mansikki tekee? Mitä ope sanoi? Mitä kirjainta kuva esittää?
3. Datan kuvaus ja ymmärtäminen
   Mitkä tekijät altistavat vatsahaavalle? Mitkä suojelevat siltä? Mitkä geenit vaikuttavat keuhkosyövän syntyyn?

Ongelmien ja menetelmien tyypittelyä

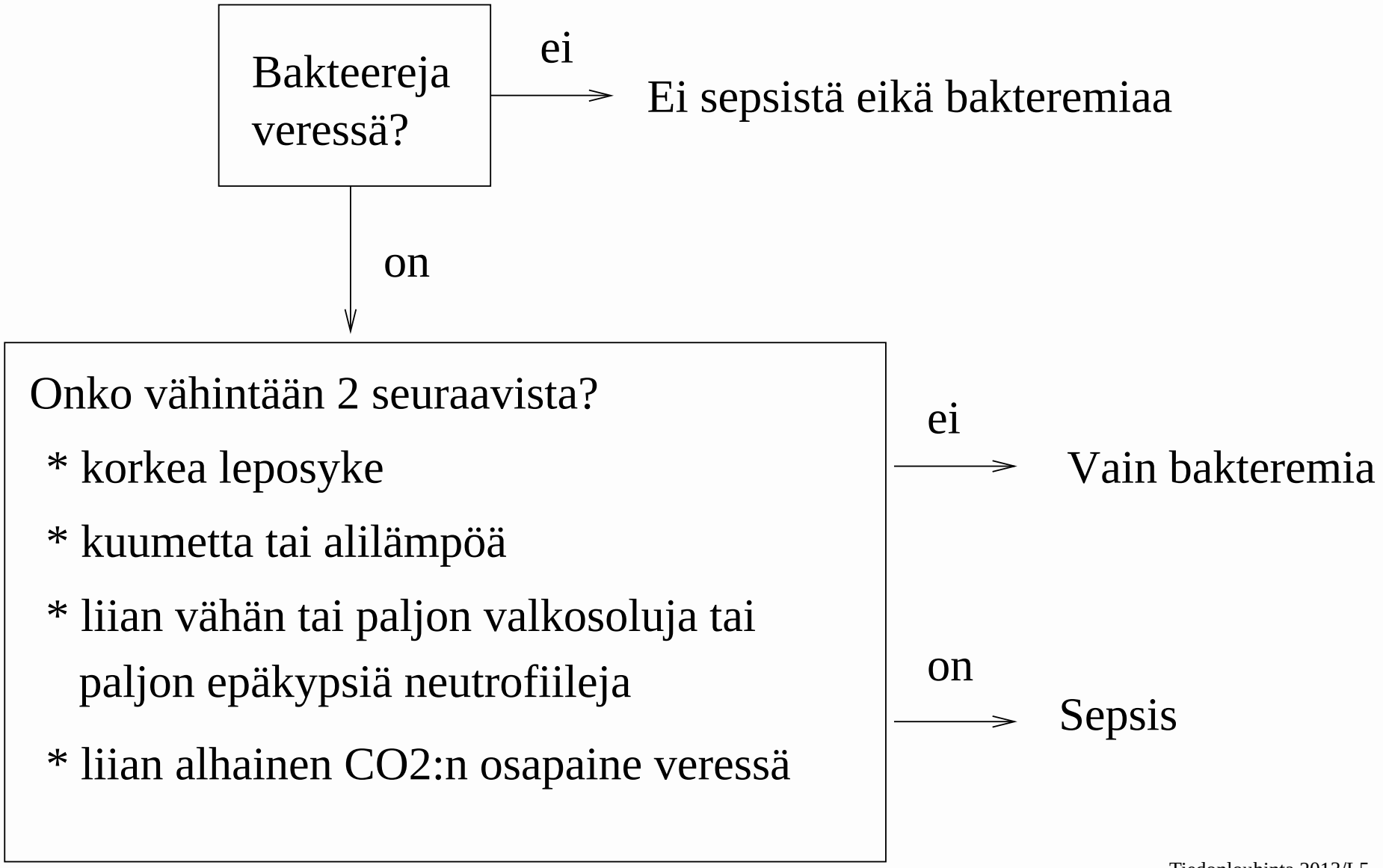
- Ovatko luokat toisistaan erottuvia? (annetuilla muuttujilla)
  → konsistentti ja epäkonsistentti data
- Millaisia luokkien väliset rajat ovat?
  → lineaariset ja epälineaariset luokkarajat
- Haluammeko vastaukseksi 1-käsitteisen luokan vai luokkien todennäköisyysjakauman?
  → erottelevat ja probabilistiset luokittelijat
- Tarvitsemmeko eksplisiittistä mallia vai riittääkö vain luokittelun tulos?
  → laiska vai ahkera (eager) menetelmä?

Konsistentti ja epäkonsistentti data

Data on epäkonsistenttia, jos siitä löytyy vähintään kaksi alkiota, joilla on samat selittävien muuttujien A_i arvot mutta eri luokat.

- Vaikka annettu data olisikin konsistenttia, voimmeko silti olettaa, että kaikki tuleva datakin on konsistenttia?
- Ts. **määräytyykö luokka deterministisesti** selittävien muuttujien arvoista?
- Konsistentissa datassa mahdollista opetusvirhe=0
- $\leftrightarrow$ Epäkonsistentissa aina luokitteluvirhettä
- Konsistenssi tyypillistä ihmisten luomille luokitteluille
- Mutta “oikeassa” elämässä harvinaista!

Esimerkki: Deterministinen luokittelu



Esimerkkejä epädeterministisistä luokitteluongelmista

- Annettuna potilaan oireet ja testitulokset, mikä sairaus hänellä on?
- Annettuna opiskelijan harjoituspisteet ja esitietokyselyn vastaukset, läpäiseekö hän kurssin?
- Annettuna henkilön perimä (geenit), saako hän ateroskleroosin (diabeteksen, Alzheimerin taudin, tms.)?
- Annettuna Mansikin syömistiedot, käyttäytyminen ja kantoaika, poikiiko Mansikki 12h kuluessa?

Erottelevat ja probabilistiset luokittelijat

1. **Erotteleva** (discriminative) luokittelija: määrittää kuvauksen $\mathcal{D} \rightarrow Dom(C)$
   - pakottaa kaikki alkiot johonkin luokkaan
2. **Probabilistinen** luokittelija: määrittää kuvauksen $\mathcal{D} \rightarrow [0,1]^L$, missä $M(x_i) = [P(C = c_1|x_i), \ldots, P(C = c_L|x_i)]$ (eli luokkien todennäköisyysjakauma)
   - todennäköisyysjakauma sisältää enemmän informaatiota $\rightarrow$ apua tulosten tulkinnassa
   - haluttaessa voi valita todennäköisimmän luokan "oikeaksi" luokaksi

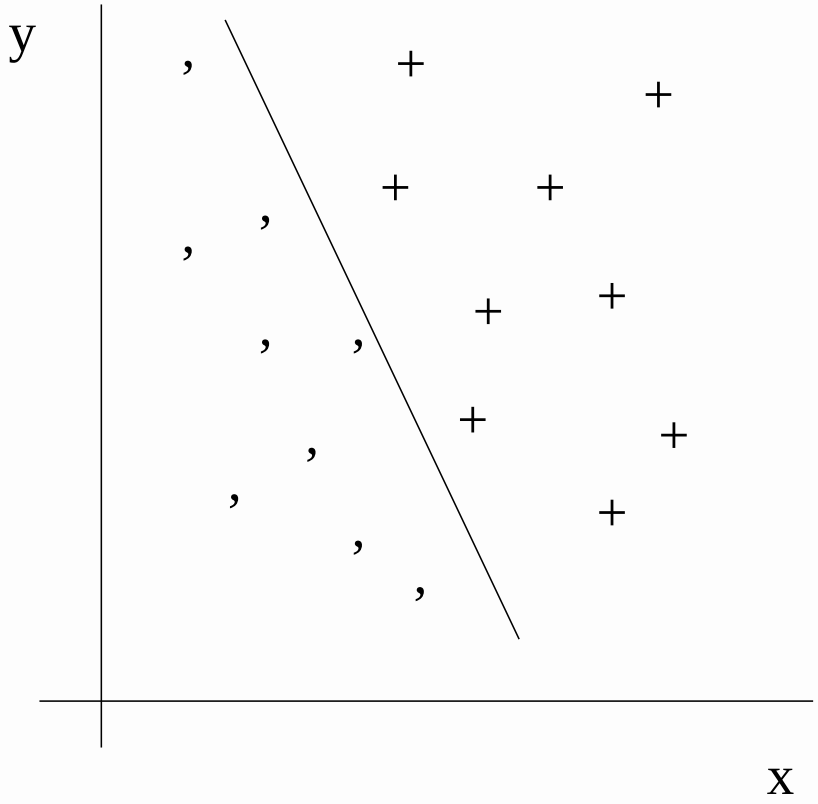
Probabilistinen luokittelu älykkäissä järjestelmissä

- Järjestelmän pitäisi valita kussakin tilanteessa (tilanneluokassa $T = t_i$) optimaalinen toiminto $A = a_j$
- Tunnetaan tilanneluokkien todennäköisyysjakauma $P(T = t_i|x)$ (x=senhetkiset muuttujien arvot)
- Määritetty toimintojen “oikeellisuuksien” ehdolliset todennäköisyysjakaumat $P(A = a_j|T = t_i)$ (tn että a_j oikea toiminto tilanteessa t_i)
- Toiminnon “oikeellisuuden” todennäköisyys, annettuna x: $P(A = a_j|x) = \sum_{i=1}^{L} P(T = t_i|x) \cdot P(A = a_j|T = t_i)$
- Esim. Pitäisikö kutsua eläinlääkäri ($A = a_1$) vai antaako soodaa ($A = a_2$), jos Mansikki on passiivinen, syönyt pikkuisen, mutta ei märehtinyt? (x)

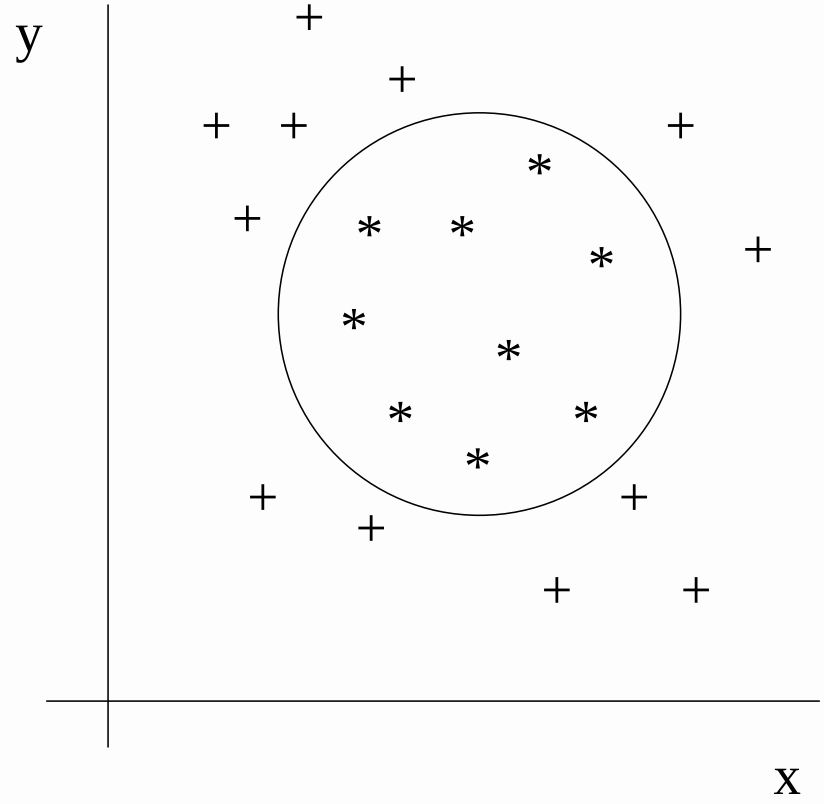
Luokkarajan lineaarisuus/epälineaarisuus

Voiko luokat erottaa toisistaan suoralla tai hypertasolla?
(erottelufunktio lineaarinen $f(X) = \beta_1 X_1 + \ldots \beta_k X_k + \alpha$)

Linearly separable:



Linearly inseparable:



Lineaarinen luokittelija

- olettaa että luokkaraja (class/decision boundary) lineaarinen
- yksinkertainen, joten kestävämpi ylisovittumista vastaan
- pieni poikkeama lineaarisuuden oletuksesta ei haittaa (mallin robustius kompensoi heikkoa ilmaisuvoimaa (representational power))
  $\rightarrow$ kannattaa suosia!
- mutta huono jos luokkarajat oikeasti monimutkaisia!

Epälineaarinen luokittelija

- epälineaarinen luokkaraja
- aste-eroja ilmaisuvoimassa: miten monimutkaisia luokkarajoja pystyy esittämään?
- Ongelma: mitä ilmaisuvoimaisempi menetelmä, sitä monimutkaisempia malleja, sitä suurempi ylisovittumisen vaara ja sitä enemmän dataa tarvitaan $\rightarrow$ monimutkaisuutta kannattaa rajoittaa

Laiska ja ahkera luokittelija

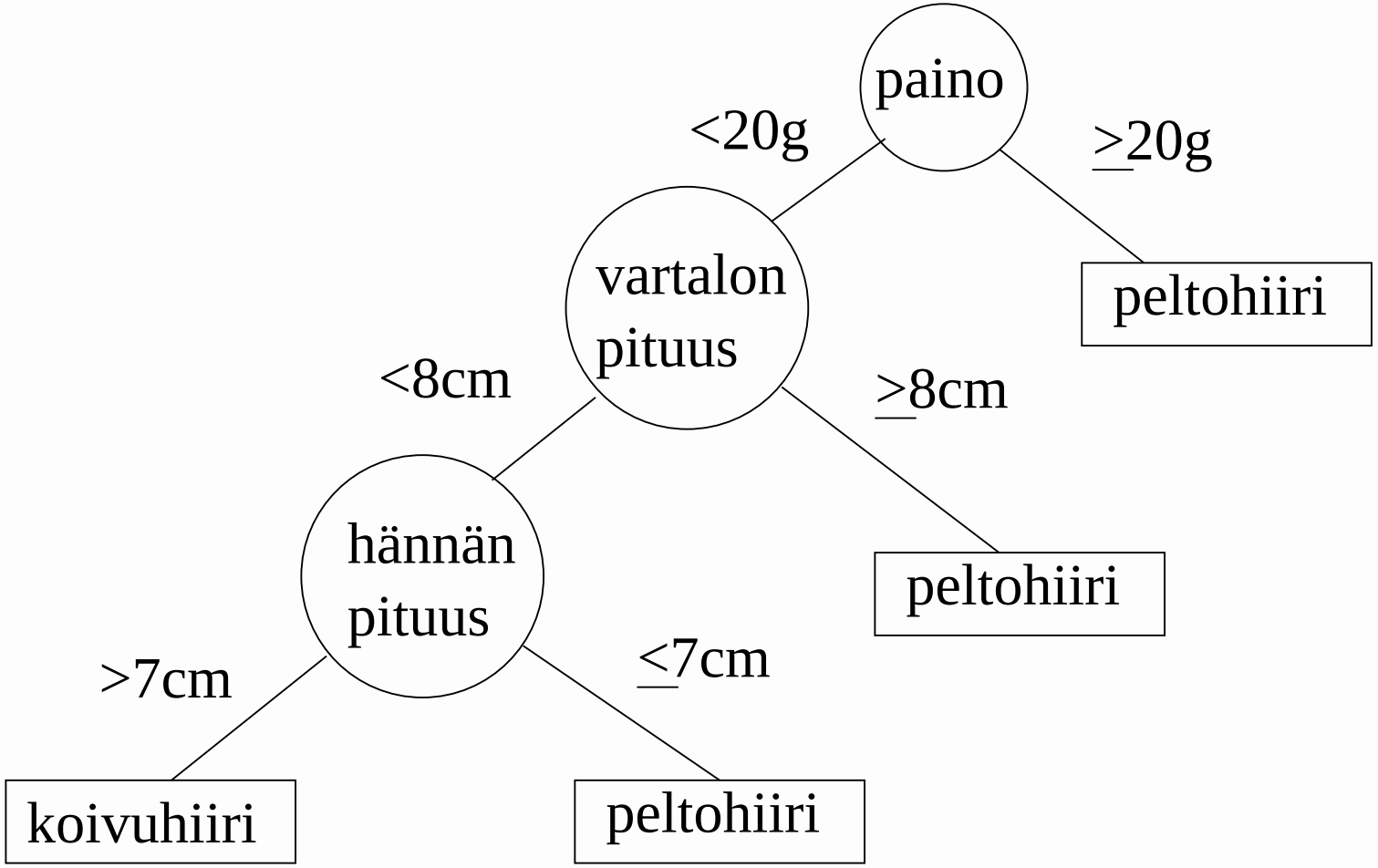
1. Ahkera luokittelija (eager learner): Muodostaa opetusdatasta eksplisiittisen mallin (luokittelijan), jolla luokittelee tulevat alkiot
   - (yleensä) vain yksi globaali malli
   - hitaampi oppia, nopea luokittelemaan
2. Laiska luokittelija (lazy learner): Kerää vain opetusdatan talteen ja suorittaa luokittelun vertaamalla opetusdataan
   - instance-based learning (case-based learning sama monimutkaisemmille objekteille)
   - esittää implisiittisesti monia lokaaleja malleja
   - ei oppimista, mutta hitaampi luokittelemaan

Tärkeimmät menetelmät

1. Päätöspuut ja luokittelusäännöt
2. Bayes-luokittelijat
3. Lähimpien naapurien menetelmät (Nearest neighbour classifiers)
4. Neuroverkkoluokittelijat
5. Tukivektorikoneet (support vector machines, SVM)

1. Päätöspuut

Luokittelijan voi esittää puuna, jonka sisäsolmut esittävät muuttujia koskevia ehtoja ja lehtisolmut luokkia.



Esitys luokittelusääntöinä

Luetellaan päätöspuun kaikki juuri–lehti-polut:

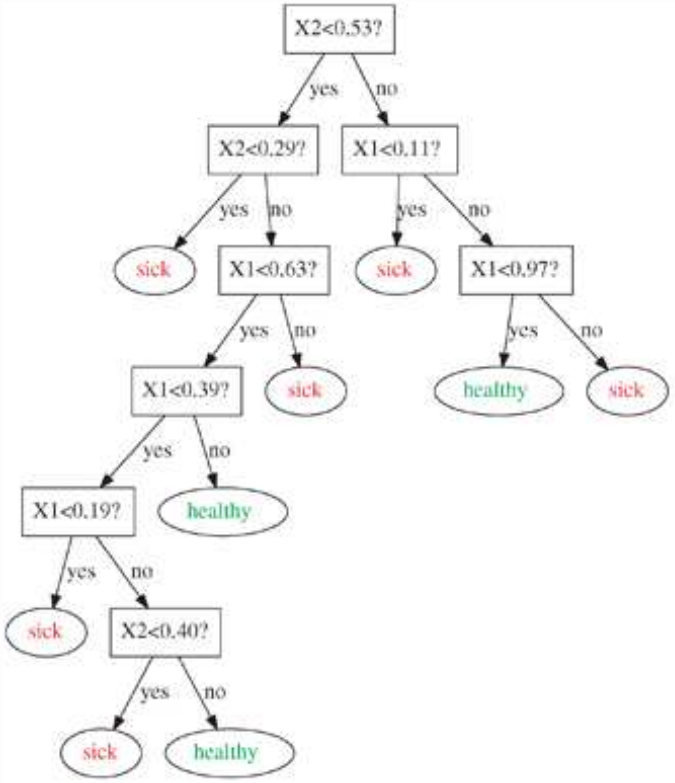
Jos paino $\geq$ 20g, niin peltohiiri.
Jos paino $<$20g ja vartalonpituus $\geq$ 8cm, niin peltohiiri.
Jos paino $<$20g ja vartalonpituus $<$ 8cm ja hännänpituus $>$7cm, niin koivuhiiri.
Jos paino $<$20g ja vartalonpituus $<$ 8cm ja hännänpituus $\leq$7cm, niin peltohiiri.

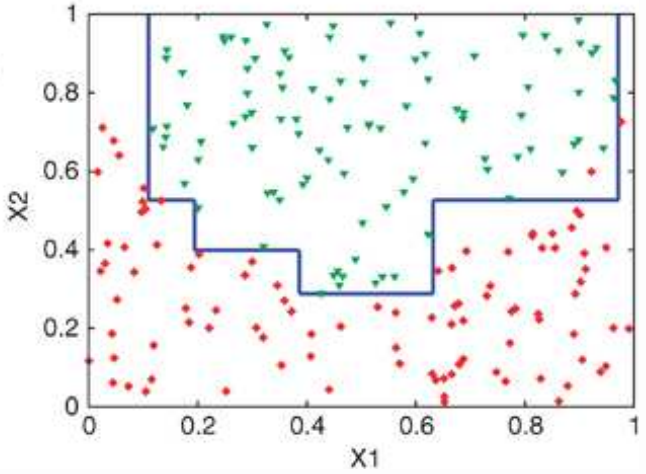
- Näihin voi lisätä ehdolliset todennäköisyydet, mutta tuloksena saatava malli usein epävakaa
- Toisaalta datasta voi etsiä parhaat luokittelusäännöt ja rakentaa niistä luokittelijan (hyviksi kehuttuja)

Etuja

- \+ voi olla sekaisin kategorisia ja (diskretoimattomia) numeerisia muuttujia
- \+ havainnollinen malli, joka auttaa ymmärtämään dataa
- \+ oppimisalgoritmi valitsee samalla muuttujat
  - etukäteen valitut ja/tai hyvin diskretoidut muuttujat parantavat silti!
- \+ ilmaisuvoimainen (voi esittää monimutkaisia luokkarajoja)
- \+ varsin robusti (sietää pientä kohinaa ja puuttuvia arvoja)
- \+ nopea luokittelu (kun malli valmis)
- \+ sopii noviiseille (ei hankalia parametreja asetettavana)

Päätöspuun luokkarajat ovat paloittain muuttuja-akselien suuntaisia





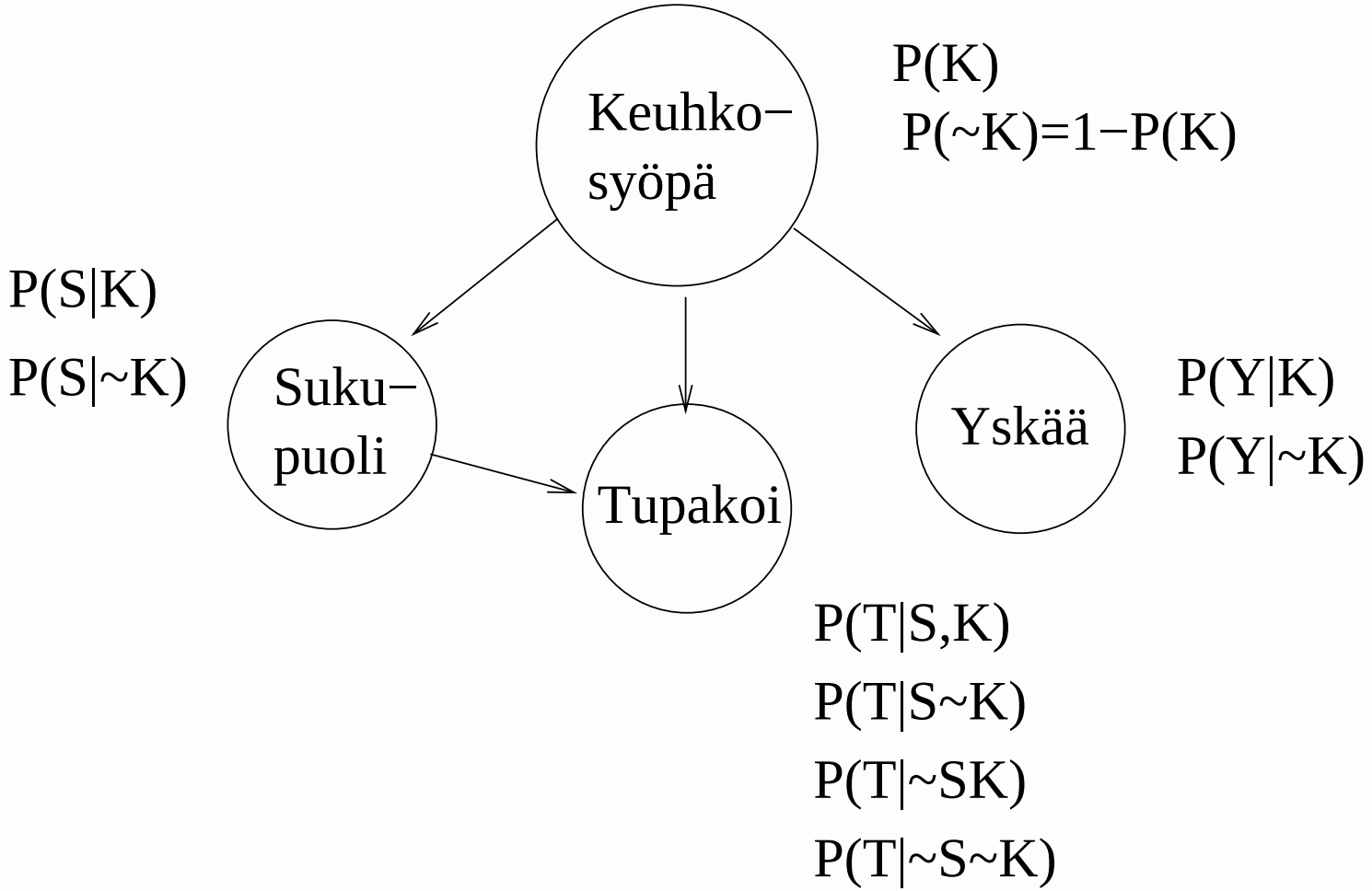
Lähde: Geurts et al: Supervised learning with decision tree-based methods in computational and systems biology. Mol. BioSyst., 2009,5, 1593-1605.

Haittoja/rajoituksia

- Globaalisti optimaalinen oppiminen “mahdotonta” (*NP*-kova ongelma) → heuristiset algoritmit
  - vanha id3: ahne puun muodostus (valitsee aina parhaan muuttujan ja sen arvoille testin) – huono!
  - parempi versio c4.5 (Wekassa J48): karsii puuta lopuksi
  - vielä uudempi c5.0: nopeampi, pienemmät puut + muita algoritmeja

  → voi tuottaa suboptimaalisen luokittelijan!
- melko herkkä ylisovittumaan, varsinkin jos dataa vähän
- deterministinen luokkaoletus
- jotkut luokkarajat hankalia esittää → turhan monimutkainen puu

2. Bayes-verkkoluokittelija



Bayes-verkon idea

- esittää datan riippuvuusrakenteen ja implisiittisesti kokonaistodennäköisyysjakauman
- Jos kaksi muuttujajoukkoa X ja Y ehdollisesti riippumattomia annettuna kolmas Z, niin verkossa kaikki X:n ja Y:n solmut jonkin Z-solmun erottamia
- Juurisolmuille A_i annetaan prioritodennäköisyydet $P(A_i = a_i)$ (kaikilla A_i:n arvoilla a_i)
  - tähän voi integroida asiantuntijatietämystä (domain knowledge)
- muille solmuille B_i ehdolliset todennäköisyydet $P(B_i = b_i|X = \overline{x})$ (X=solmun vanhemmat ja $\overline{x}$=niiden arvokombinaatio)

Todennäköisyyksien päivitys

Idea: Solmujen todennäköisyyksiä päivitetään **Bayesin säännöllä**, kun saadaan uutta tietoa (kiinnitetään muuttuja-arvoja):

$$P(A = a_i | X = x) = \frac{P(A = a_i)P(X = x | X = x)}{P(X = x)}$$

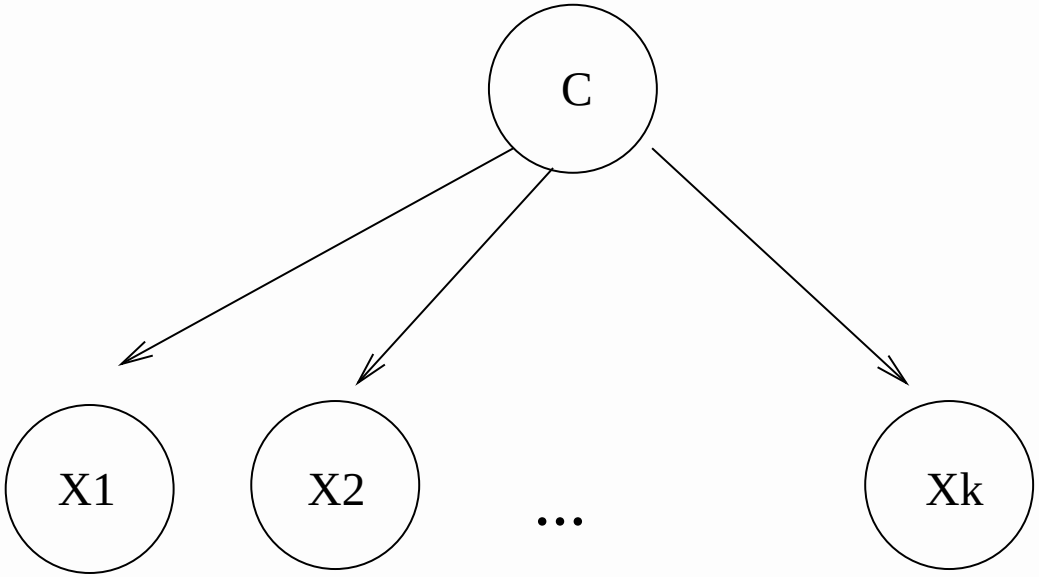
- hyödyntää aina kaiken saatavilla olevan tiedon!
- mutta voi olla raskasta (*NP*-kovaa)
- myös mallin (verkon rakenne + todennäköisyydet) oppiminen *NP*-kovaa

Bayes-verkkoluokittelija

- Puurakenteinen → päättely kevyempää!
- Juuressa luokkamuuttuja, muut muuttujat sen alla
- Luokkajakaumaa päivitetään Bayesin säännöllä, kun muiden muuttujien arvoja kiinnitetään (→ sallii puuttuvat arvot!)
- Yksinkertainen Naiivi-Bayes-luokittelija suosituin

Naiivi-Bayes-luokittelija

Vain 2-tasoinen eikä lehtien välillä kaaria



Naiivi-Bayes-oletus: X_i:t toisistaan ehdollisesti riippumattomia, annettuna C eli

$$P(X_1, \ldots, X_k|C) = \Pi_{i=1}^{k} P(X_i|C)$$

- Usein epärealistinen mutta voi silti toimia hyvin!

Naiivi-Bayesin edut

- \+ robusti kohinalle
- \+ hyvin resistentti ylisovittumiselle
- \+ voi oppia pienestäkin datamäärästä
- \+ ei tarvitse oppia rakennetta → helppo oppia
- \+ voi hyödyntää kaiken informaation, mutta sallii puuttuvat arvot
- \+ sekadatallekin menetelmiä (yleensä diskretoivat numeeriset muuttujat)
- \+ sopii hyvin noviiseille

Naiivi-Bayesin edut

- Naivi-Bayes-oletus usein epärealistinen! $\rightarrow$ jos vahvoja X_i:den välisiä riippuvuuksia, luokittelu kärsii
- pieni esitysvoima
  - jos kategorisia muuttujia, vain lineaariset luokkarajat

Yleisen Bayes-luokittelijan edut ja haitat

- \+ hyvin informatiivinen!
- \+ voi esittää epädeterminismiä
- \+ voi olla sekaisin kategorisia ja numeerisia muuttujia
- \+ hyvin ilmaisuvoimainen
- – rakenteen oppiminen vaikeaa!
  $\rightarrow$ suboptimaaliset menetelmät
- – vaatii paljon dataa
- – herkempi ylisovitumiselle

3. K:n lähimmmän naapurin menetelmä (K-Nearest neighbour classifier)

Idea: Luokitellaan piste x siihen luokkaan, mihin suurin osa sen K:sta lähimmästä naapuripisteestä kuuluu.

- Tarvitaan etäisyys- tai samankaltaisuusmitta d
  - $d(x_1, x_2)$ kertoo pisteiden x_1 ja x_2 etäisyyden tai samankaltaisuuden asteen
  - numeeriselle datalle esim. euklidinen etäisyys
    $d(x_1, x_2) = \sqrt{(a_{11} - a_{21})^2 + \ldots + (a_{1k} - a_{2k})^2}$
  - merkkijonoille esim. minimum edit distance
  - binaaridatalle esim. Hamming-etäisyys
  - kategorinen ja sekadata hankalimpia!

K:n lähimmän naapurin menetelmä

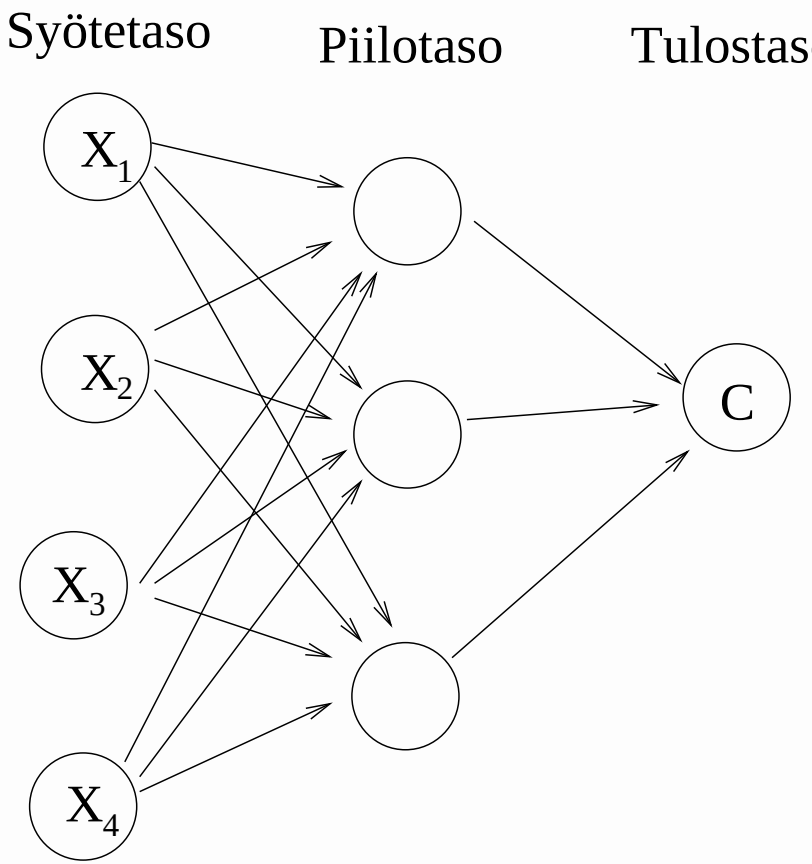
- lisäksi määritettävä naapuruston koko K
  - jos liian pieni, tulee epävakaa luokittelija
  - jos liian suuri, voi naapuruston yleisin luokka poiketa oikeasta
  - pienidimensioiselle datalle 5–10 hyvä
- ei suositella suuridimensioiselle (> 20 muuttujaa) datalle, varsinkin jos harvaa
- eräs (hyvä) variaatio: huomioidaan kaikki naapurit, mutta painotetaan naapurin luokkaa sen etäisyydellä

K:n naapurin edut ja haitat

- \+ ei oppimista
- \+ voi olla sekaisin numeerisia ja kategorisia muuttujia (kunhan keksii d:n)
- \+ varsin robusti kohinalle ja puuttuville arvoille
- \+ hyvin ilmaisuvoimainen – luokkarajat voivat olla monimutkaisia
- \+ luokittelija paranee käytössä (kun kertyy lisää dataa)
- – hyvän d:n määritys kriittistä
- – ei sovi suuridimensioiselle datalle
- – luokittelu voi olla hidasta (jos paljon dataa)

4. Neuroverkkoluokittelijat

Tyypillisesti monitasoinen perseptroni (multilayer perceptron, MLP)
↔ perseptronissa ei piilotasoja



Verkon topologian valinta

- 0 piilotasoa $\rightarrow$ perseptroni – voi esittää vain lineaarisia luokkarajoja (mutta yksinkertaisuus voi kompensoida ilmaisuvoiman puutetta)
- 1 piilotaso eli 3-tasoinen verkko: pystyy esittämään mitä tahansa luokkarajoja (epälineaarisia funktioita)
- Ei siis kannata turhaan käyttää useampia tasoja! (vain jos tarpeen)
- hyvin erottuville luokille riittää pienempi määrä piilosolmuja
- Peukalosääntö (solmujen lkm:n valintaan): painoja tulisi olla noin $\frac{n}{10}$ (n=opetusjoukon koko)

Muut parametrit

- aktivaatiofunktio (yl. sigmoid-funktio)
- painojen alkuarvot (Suositus: satunnaiset pienet posit. arvot)
- painojen päivityssääntö (kannattaa lisätä momentum-termi)
- oppimisalgoritmi
  - useimmiten back-propagation (perustuu gradienttihakuun)
  - hienommat algoritmit voivat säätää verkon topologiaa ja valita piirteitä

Muut parametrit

- minimoitava virhefunktio
  - yleensä SSE
  - ylisovittumiseen auttaa lisätermi, joka rankaisee suurista painoista
- lopetuskriteeri (testijoukko tai ristiinvalidointi hyvä ratkaisu)

$\Rightarrow$ Säätö vaatii asiantuntemusta!

Erittäin vahva työkalu oikeissa käsissä, mutta ylisovittumisen vaara hyvin suuri.

“Neuroverkko työkaluna on yhtä hyvä kuin käyttäjänsä”

Neuroverkon edut ja haitat

+ erittäin ilmaisuvoimainen
+ robusti kohinalle (kunhan ei ylisovitu)
+ voi arvioida myös luokkien todennäköisyyksiä
– erittäin herkkä ylisovittumaan
– vaatii paljon dataa
– vain numeeriselle datalle (ehkä ordinaaliselle)
– mallia vaikea tulkita ja tarkistaa
– oppimisalgoritmit suboptimaalisia, mutta silti hitaita
– vaatii asiantuntemusta (paljon parametrivalintoja)

5. Tukivektorikoneet (support vector machines, SVM)

Perusidea: Kuvataan data suurempaan ulottuvuuteen, jolloin luokkaraja muuttuu lineaariseksi. Sitten etsitään “levein hypertaso” (lineaarinen luokkarajavyöhyke), joka erottaa luokat.

- **tukivektorit**=vaikeimmin luokiteltavia pisteitä → määrittävät luokkarajan
- yleensä sallitaan joitain luokitteluvirheitä (“pehmeä rajavyöhyke”, “soft margin”)
- kuvaus suurempaan dimensioon tehdään implisiittisesti, ydinfunktioilla (kernel functions)

Valittavat parametrit

- kernel-funktio ja sen parametrit
  - tässä voi hyödyntää esitietämystä
  - muuten radial basis funktio hyvä oletusarvo (jos epälineaarinen luokkaraja)
- pehmeän marginaalin parametri (=regularisointiparametri, regularization parameter)
- oppimisalgoritmi
  - globaalisti optimaalisia!
  - Mutta ei muuttujien valintaa.

SVM:n edut ja haitat

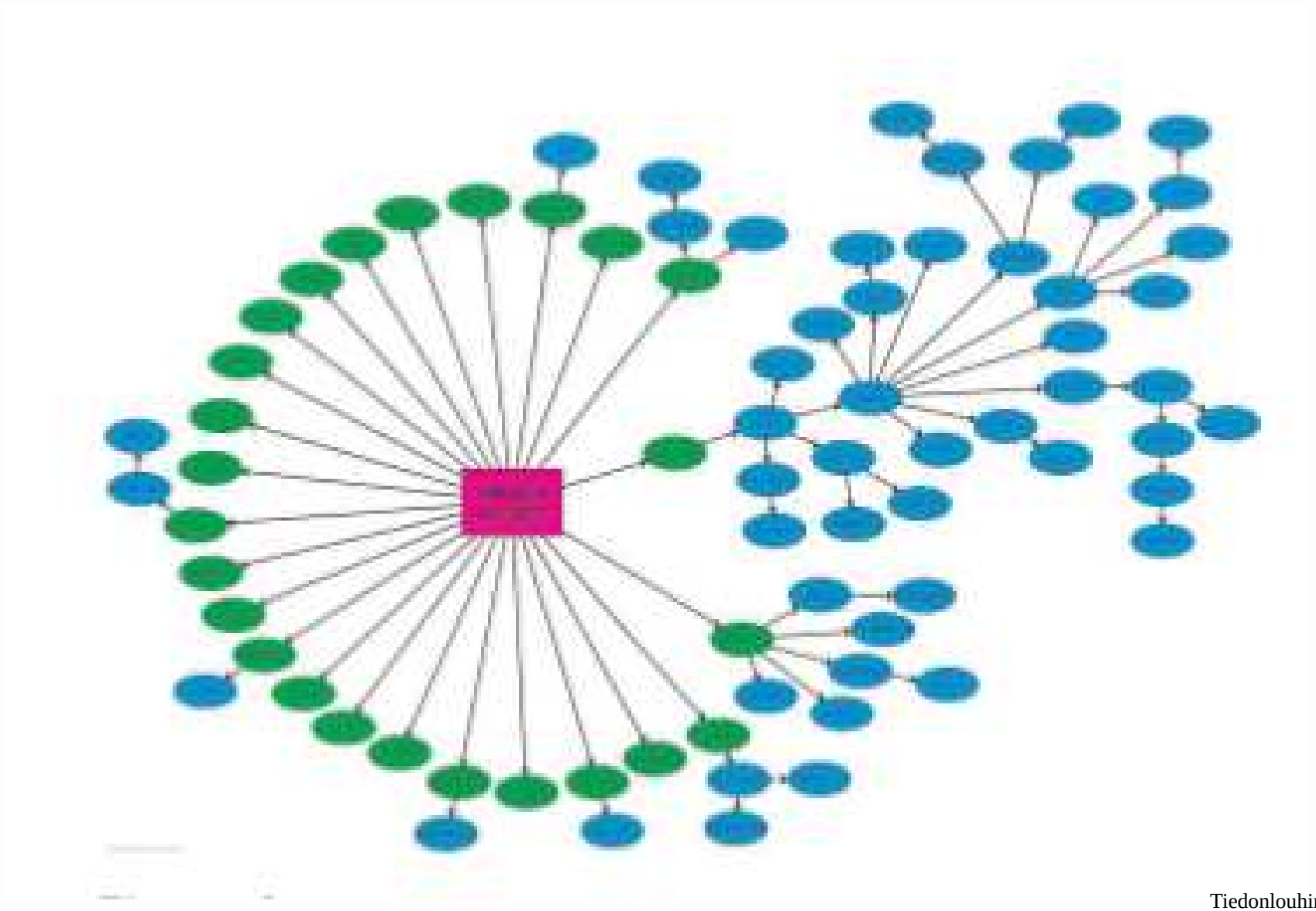
+ erittäin ilmaisuvoimainen
+ resistantti ylisovittumiselle
+ toimii vaikka muuttujia olisi enemmän kuin datarivejä!
+ globaalisti optimaalinen malli (mutta perusmenetelmä ei valitse muuttujia)
– vain numeeriselle (ehkä ordinaaliselle) datalle
– (perusmenetelmä vain 2:n luokalle (on laajennoksia))
– mallia vaikea tulkita
– parametrien valinta voi olla vaikeaa (automaattisissa menetelmissä suuri ylisovittumisen vaara)
– ulkopuoliset (outliers) saattavat olla ongelma?

Esimerkki 1: Bayes-luokittelija keuhkosyöpätyypin tunnistukseen geenien perusteella

- Keuhkosyövän yleisimpien tyyppien, AC:n (adenocarcinoma) ja SCC:n (squamous cell carcinoma), vaikea mutta tärkeä ongelma
- Bayes-luokittelijalla onnistuttiin löytämään 25 geeniä, jotka ennustavat syöpätyypin 95% tarkkuudella opetusjoukon ulkopuolisella datalla
- Bayes-verkolla voi hyvin esittää myös geenien väliset säätelysuhteet (hyvä kuvaava malli)

Lähde: Chang and Ramoni: Transcriptional network classifiers. BMC Bioinformatics 2009, 10(Suppl 9):S1.

Bayes-luokittelija keuhkosyöpätyypin tunnistukseen



Esimerkki 2: Leukemiatyypin ennustaminen SVM:illä

- Datana 2308 geenin tiedot kahta leukemiatyyppiä, AML:aa (acute myeloid leukemia) ja ALL:aa (acute lymphoblastic leukemia), sairastavilta (vain 38+34 riviä dataa)
- Vertailtiin tavallista SVM:ää ja kolmea muuttujien valintaa suorittavaa SVM:ää
- Paras malli (DrSVM) valitsi 78 geeniä ja testivirhe oli 0

Lähde: Zhu and Zou: Variable selection for the linear support vector machine. Computational Intelligence (SCI) 35, 35-59 (2007).

Luento 6: Klusterointi

- Klusterointiongelma
  - etäisyys-/erilaisuusfunktiot
  - dimensionaalisuuden kirous (curse of dimensionality)
- Menetelmien pääluokat
  1. Osittavat
  2. Hierarkiset
  3. Malliperustaiset (tiheysfunktioihin perustuvat)
- Mikon esitys tasapainoisesta klusteroinnista

Klusterointiongelma

Intuitio: Halutaan löytää datan ryhmittely K:hon klusteriin $c_1, \ldots, c_K$, siten että klusterin sisäiset pisteet ovat lähekkäisiä/samankaltaisia mutta klusterien välillä on eroja/etäisyyksiä.

Tarvitaan ainakin:

- K=klusterien lkm
- d=pisteiden etäisyys- tai erilaisuusfunktio
- D=klusterien etäisyys- tai erilaisuusfunktio
- kriteeri jolla arvioida klusteroinnin hyvyyttä (score function) (ei tarvita aina)
- klusterointimenetelmä

Usein suotavia ominaisuuksia

- Ei rajoittavia oletuksia klusterin muodosta (eri klusterit voivat olla erimuotoisia)
- Samalla olisi hyvä tunnistaa ulkopuoliset
- Jos datassa ei selvästi erottuvia ryhmiä, niin pitäisi esittää jotenkin havainnollisesti klusterien päällekäisyys
- Klusterien lukumäärä haluttaisiin määrittää datasta
- Tasapainoisuus joskus suotavaa (esim. diskretointi), mutta dataa ei pidä väkisin pakottaa siihen (esim. ulkopuoliset, vähemmistöt)
- Lisäinformaationa mahdollisesti klusterien väliset suhteet (klusterihierarkia) tai datan tiheysjakauma

Usein suotavia ominaisuuksia

- Jos datassa ei oikeasti ole ryhmiä, sekin olisi tärkeää saada selville!

Vaara: Klusterointi voi luoda dataan (katsojan silmään) ryhmittelyn, vaikkei sitä olisikaan!

Etäisyysfunktio numeeriselle datalle

Merk. datapiste $x = (x_1, \ldots, x_k)$ (kaikkien k:n muuttujan arvot)

- yleisin d on L_p-metriikka eli Minkowski-etäisyys (perhe etäisyysmittoja):

$$L_p(x, y) = \left[\sum_{i=1}^{k} (x_i - y_i)^p \right]^{1/p}$$

- $p \in \mathbb{R}^+$ kertoo mistä mitasta on kyse
- kun $p = 2$, euklidinen etäisyys
- kun $p = 1$, Manhattan-etäisyys
- robustimpi mitta, kun pieni p

Etäisyysfunktio numeeriselle datalle

- L_p:t toimivat hyvin, kun klusterit tiiviitä ja hyvin erottuvia
- mutta eivät toimi, jos muuttujat eri skaalassa!
- Mitä jos yksi muuttuja grammoja, toinen milligrammoja ja kolmas senttejä??

Dimensionaalisuuden kirous (curse of dimensionality)

Data-avaruuden tilavuus kasvaa eksponentiaalisesti dimension funktiotan $\rightarrow$ avaruudesta tulee hyvin harva. Kaikki naapuripisteet kaukana eikä data riittäisi enää minkään mallin määritykseen.

- muitakin merkityksiä (kaikenlaiset suuresta dimensiosta seuraavat vaikeudet)
- klusteroinnin erityisongelma: ei enää selkeitä tihentymiä
- etäisyysmitta menettää merkityksenä, kun kaikkien pisteiden etäisyydet jokseenkin yhtä suuria
- euklidinen etäisyys L_2 kärsii erityisesti $\rightarrow$ L_1 (tai $L_{0.5}$ tms. $p < 1$) parempi

Dimensionaalisuuden kirous (jatk.)

- epärelevantit muuttujat alkavat helposti dominoida mittaa
  Mitä jos $k = 20$ mutta vain 2 muuttujaa relevantteja?

Huom! aikavaativuuden kannalta datarivien lkm n kriittisempi ongelma.

- n pistettä voi jakaa K:hon klusteriin $\frac{K^n}{K!}$ eri tavalla
- yleensä klusterointiogelmat *NP*-kovia, mutta riippuu tietysti klusteroinnin määritelmästä!

1. Osittavat menetelmät (partitioning)

Idea: Siirrellään datapisteitä klusterista toiseen, kunnes hyvyysfunktio ei enää parane.

- Klusterien esitys=centroidit (“keskipisteet”)
  - centroidin muuttuja-arvot yleensä keskiarvoja tai mediaaneja klusterien pisteiden muuttuja-arvoista
- tunnetuin menetelmä K**-means**

K-means

Minimoitava funktio $SSE = \sum_{j=1}^{k} \sum_{x_i \in c_j} d^2(x_i, centroid(c_j))$

- yleensä $d = L_2$ (euklidinen etäisyys)
- Oletus: klusterit erillisiä, tiiviitä hyperpalloja
- kärsii epärelevanteista muuttujista (erit. suuridimensioisessa datassa) → voi tulla virheellisiä klustereita
- alkucentroidit valitaan satunnaisesti → eri ajokerroilla eri tuloksia → kannattaa ajaa useita kertoja!
- jos juuttuu lokaaliin optimiin tai oikeat klusterit eivät oletuksen mukaisia, tulee epäintuitiivisia tuloksia (usein huono “oikean elämän” datalla)

Etuja ja haittoja

- \+ varsin tehokas
- \+ helppo toteuttaa
- – hyvin rajoittunut oletus klustereista
- – voi tuottaa epäintuitiivisia tai selvästi vääriä klusterointeja
- – L_2 hyvin herkkä ulkopuolisille $\rightarrow$ k-medians parempi
- – L_2 ei sovi suuridimensioiselle datalle $\rightarrow$ esim. L_1 parempi
- – vaikea soveltaa kategoriselle tai sekadatalle
- – parametrien valinta kriittistä

Muita osittavia menetelmiä

Muuttamalla klusteroinnin hyvyysfunktiota, voi tunnistaa myös hyperellipsoidit tai tuottaa samankokoisia tai suurinpiirtein smanamuotoisia klustereita.

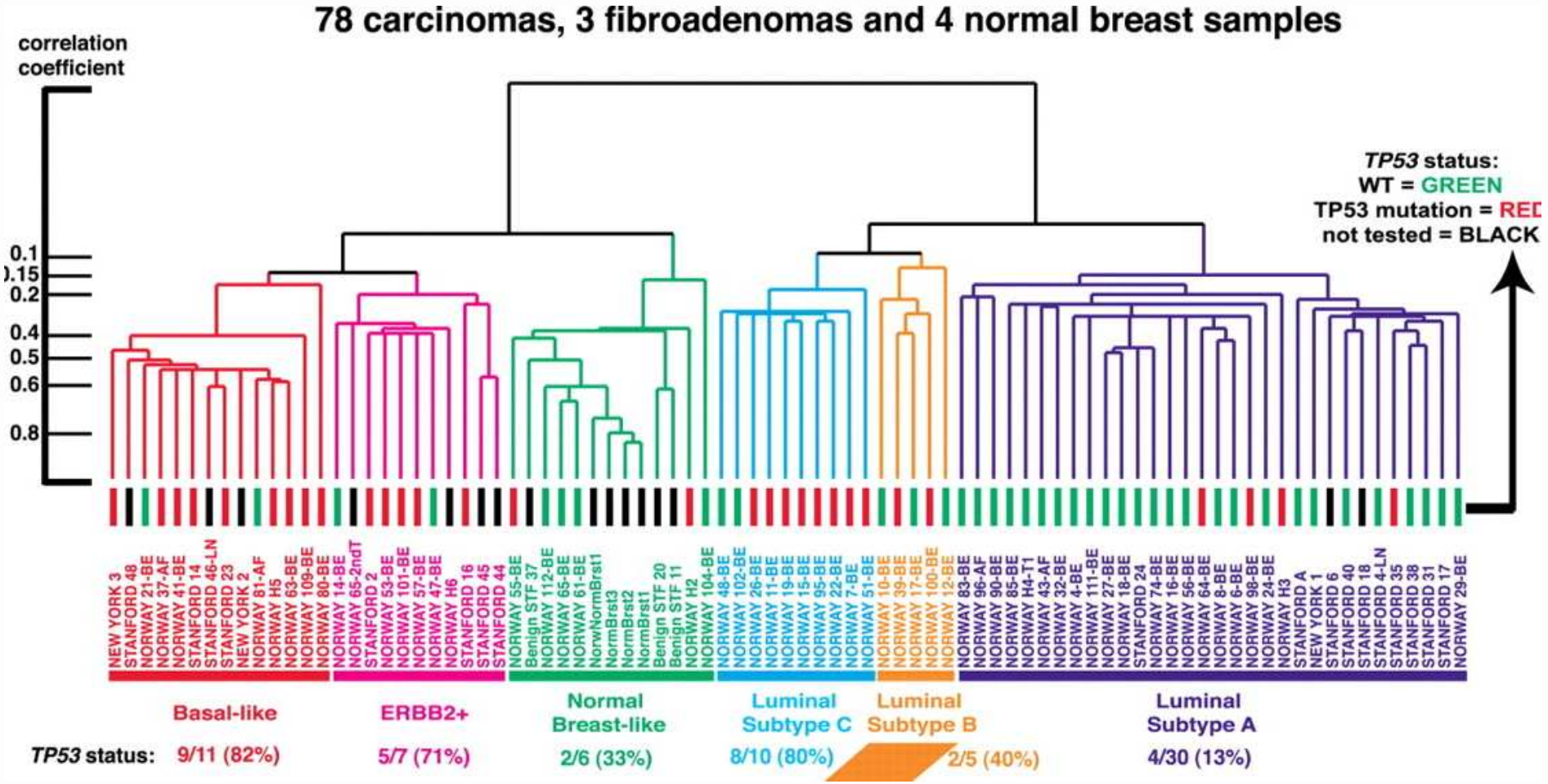
$\rightarrow$ Smyth et al, luku 9.4.1!

2. Hierarkiset menetelmät

Idea: Yhdistetään (tai ositetaan) klustereita hierarkisesti, kunnes jäljellä sopiva määrä klustereita.

- tuloksena saadaan **dendrogrammi**, joka kuvaa klusterien hierarkian
- dendrogrammin voi myös rakentaa loppuun asti (lehdissä 1-alkioiset klusterit ja juuressa koko datajoukko) ja valita sopivan lukumäärän jälkikäteen

Dedrogrammeilla voi kuvata lajien geneettisiä sukulaissuhteita!



Lähde: Sörlie et al: Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. PNAS vol. 98 no. 19, 2001.

Dedrogrammin rakennus

- kaksi rakennustapaa:
  1. Yhdistävät (agglomerative) menetelmät rakentavat dendrogrammia lehdistä juureen (yleisempi ja tehokkaampi)
  2. Jakavat (divisive) menetelmät rakentavat juuresta lehtiin
- ei eksplisiittistä hyvyysmittaa
- Monia vaihtoehtoisia mittoja klusterien etäisyydelle $D(c_1, c_2)$ $\rightarrow$ **hyvin erilaisia tuloksia!**

Klusterien etäisyysmittoja $D(c_1, c_2)$

1. Single-link: $D(c_1, c_2) = \min_{x,y}\{d(x, y) \mid x \in c_1,\ y \in c_2\}$ (lähimpien pisteiden etäisyys)
2. Complete-link: $D(c_1, c_2) = \max_{x,y}\{d(x, y) \mid x \in c_1,\ y \in c_2\}$ (kauimpien pisteiden etäisyys)
3. Centroid-mitta: $D(c_1, c_2) = d(centroid(c_1), centroid(c_2))$
4. Average-link: $D(c_1, c_2) = \frac{\sum_{x \in c_1} \sum_{y \in c_2} d(x,y)}{|c_1||c_2|}$ (kaikkien pisteparien keskimääräinen etäisyys)
5. Minimum variance: $D(c_1, c_2) = \frac{|c_1||c_2| d^2(centroid(c_1), centroid(c_2))}{|c_1|+|c_2|}$ (varianssin kasvu jos yhdistetään)

Klusterien etäisyysmitat: Vertailua

Mitta	Klusterin muoto	Huomioita
Single link	pitkulaisia, epäsäännöllisiä jopa sisäkkäisiä	sopii erillisille epäsymmetrisille Epävakaa! (kohina)
Complete link	pieniä, tiiviitä	Ei löydä pitkulaisia Ulkopuoliset sotkevat.
Centroid-mitta	tiiviitä, erillisiä hyperpalloja	Joskus huonoja tuloksia. Tehokas laskea Aika robusti.

Vertailua (jatkuu)

Mitta	Klusterin muoto	Huomioita
Average link	melko tiiviit, suuremmat voivat olla harvempia	Kehuttu! Robusti
Minimum Variance	samanmuotoisia, jokseenkin samankokoisia, symmetrisiä melko tiiviitä	Herkkä ulkopuolisille! Tehokas laskea

Etuja ja haittoja

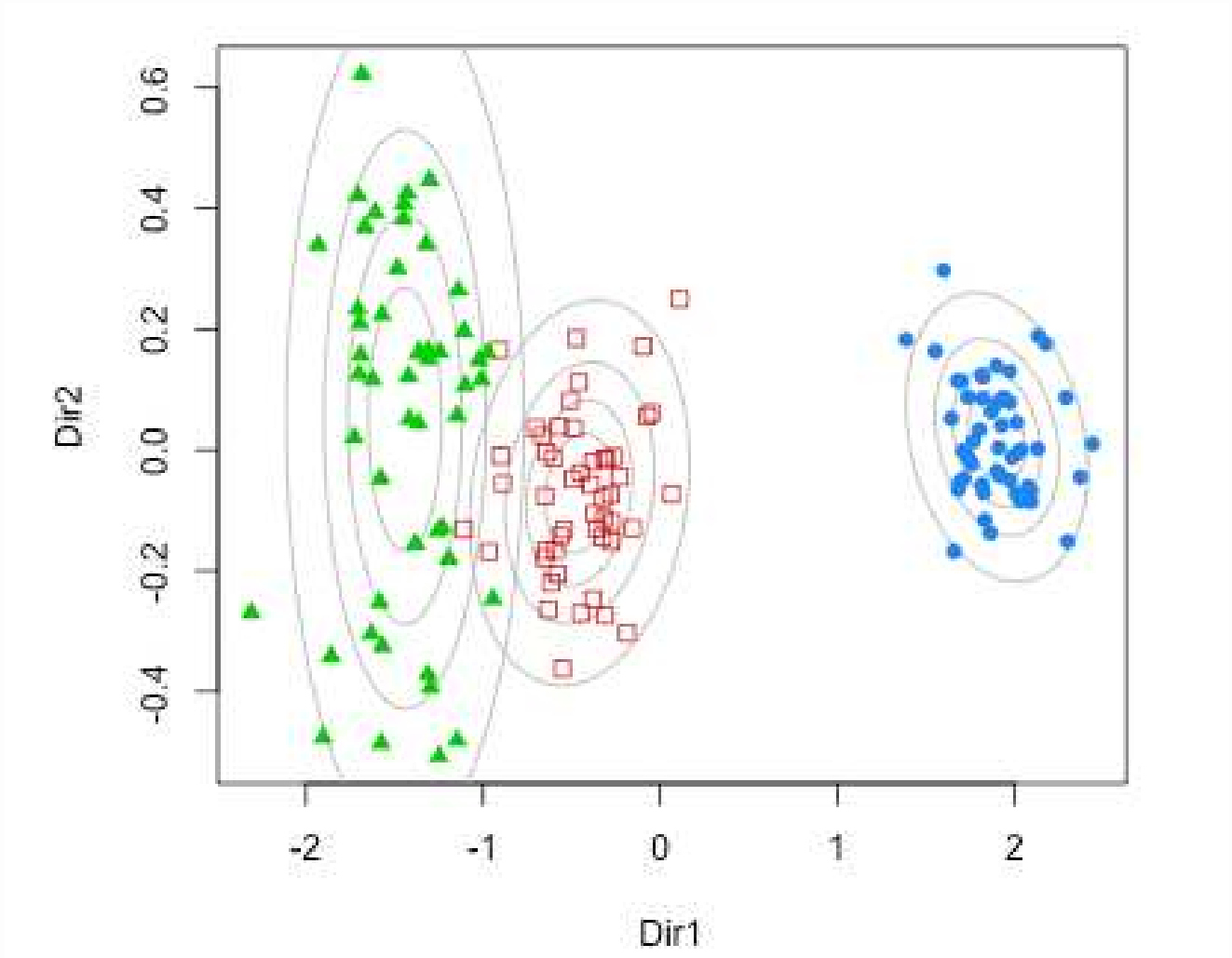
+ käytännössä toimivat yleensä hyvin (joskin riippuu mitasta!)
+ dendrogrammi lisää informaatiota
+ klusterien lkm:n voi valita dendrogrammista
+ sovellettavissa myös ei-numeeriselle datalle, kunhan klusterien etäisyydelle mitta
– useimmilla mitoilla tulos riippuu datan järjestyksestä! (em:ista complete-, centroid-, average-link ja min. var)
– skaalautuvat huonosti suurille datoille (kun paljon rivejä)
– menetelmät eivät osaa korjata aiempia tasoja

3. Malliperustaiset (tiheysfunktioihin perustuvat) klusterointimenetelmät

Idea: Määritetään koko datan tiheysjakauma, jolla data uskottavin. Samalla saadaan klusteroinnin probabilistinen kuvaus.

- sama idea kuin probabilistisessa luokittelussa!
- datapistettä ei määrätä deterministisesti vain yhteen klusteriin, vaan se voi kuulua eri todennäköisyyksillä eri klustereihin
- erittäin informatiivinen (enemmän kuin vain klusterointi)
- mutta vaikeaa tehdä hyvin...

Tuloksena saatu datan “tiheyskartta”



Lähde: mclust, R-paketti tällaiseen!

`http://www.stat.washington.edu/mclust/`

[Extra] Malliperustaisen klusteroinnin idea tarkemmin

Jokaiselle klusterille c_i ($i = 1, \ldots, K$) määritetään

- tiheysfunktio $P(x|c_i, \theta_i)$
  - kertoo todennäköisyyden, että piste x kuuluu klusteriin c_i
  - θ_i sisältää kaikki parametrit (esim. keskiarvo ja keskihajonta, jos normaalijakauma) – nämä siis **tuntemattomia!**
- prioritodennäköisyys $P(c_i)$ (klusterin todennäköisyys, jos dataa ei tunnettaisi)

[Extra] Datan uskottavuus annettuna klusterointi

- Merk. $\theta = (\theta_1, \ldots, \theta_K)$
- Datapisteen x (posteriori) todennäköisyys koko mallissa on

$$P(x|\theta) = \sum_{i=1}^{K} P(c_i)P(x|c_i, \theta_i)$$

- Jos todennäköisyys $P(x|\theta)$ kovin pieni, piste ei sovi malliin ja se voidaan tulkita sen ulkopuoliseksi
- Koko datan uskottavuus annettuna klusterointi on silloin

$$P(data|\theta) = \Pi_{j=1}^{n} P(x_j|\theta)$$

(eli kaikkien datapisteiden todennäköisyyksien tulo)

[Extra] Miten ratkaista θ?

- Maximum likelihood-periaate: Valitaan sellainen θ, joka maksimoi $P(data|\theta)$:n (eli data on uskottavin)
- vaihtoehtoisesti voisi mallintaa bayesläisittäin:

$$P(\theta|data) = \frac{P(data|\theta)P(\theta)}{P(data)}$$

- Käytännössä ratkaistaan yleensä iteratiivisesti EM-algoritmilla

Millainen todennäköisyysjakauma?

- Data yleensä moniulotteista, joten tarvitaan monen muuttujan tiheysfunktio!
- Yksinkertaisuuden vuoksi oletetaan yleensä, että **muuttujat ovat toisistaan riippumattomia**
- Epärealistista, mutta silloin yhteinen tiheysfunktio helppo esittää (yhden muuttujan tiheysfunktioiden tulona)
- Muuttujat voisi tehdä ensin riippumattomiksi (ainakin korreloimattomiksi) PCA:lla
- Numeerisille muuttujille oletetaan yleensä normaalijakauma

Etuja

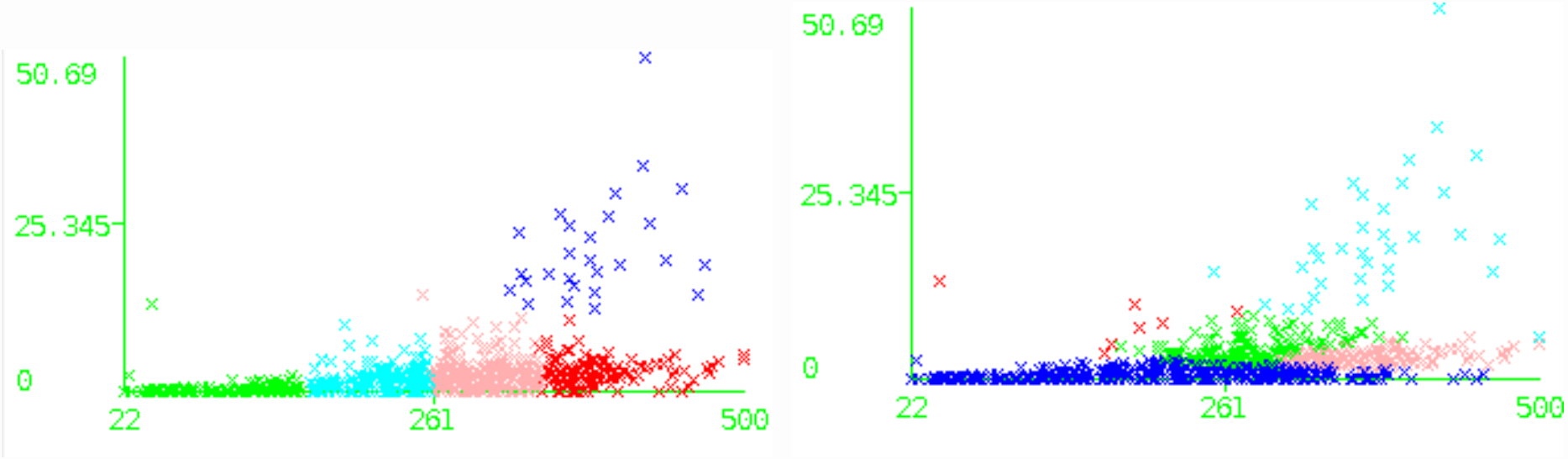
- \+ hyvin informatiivinen
- \+ voi hyvin erilaisia klustereita, myös päällekkäisiä
- \+ erittäin joustava (joka klusteri voisi noudattaa jopa erityyppistä jakaumaa)
- \+ tunnistaa samalla ulkopuoliset (ainakin periaatteessa)
- \+ toimii myös kategorisella datalla (mutta vaikeampaa)

Haittoja

- tulokset riippuvat vahvasti oletetusta jakaumasta ja K:sta
- vaikea tietää ennalta “oikeita” jakaumatyyppejä!
- jos ol. normaalijakauma, ei robusti ulkopuolisille
- muuttujien riippumattomuuden oletus rajoittava! $\rightarrow$ voi tulla huonoja tuloksia (ellei hoida esim. PCA:lla)
- vaikea käsitellä sekaisin numeerisia ja kategorisia muuttujia
- algoritmit hitaita
- EM-algoritmi suboptimaalinen ja voi joskus tuottaa huonoja tuloksia

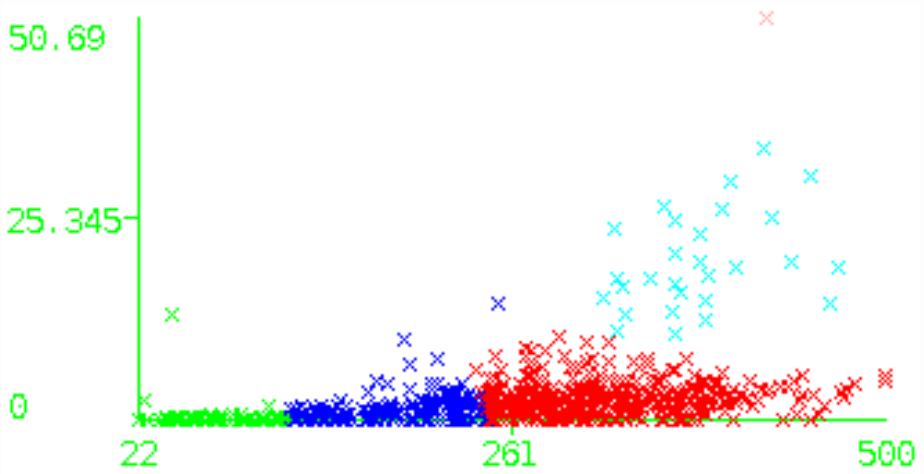
Esimerkkejä: Wekan K-means ja EM

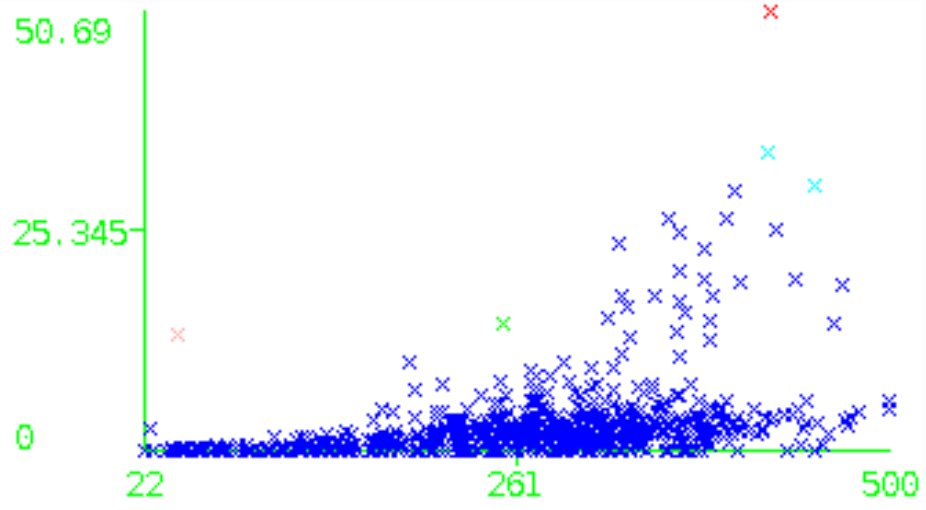
(Muuttujat rotan paino/g ja gon-rasva/mg, $K = 5$)

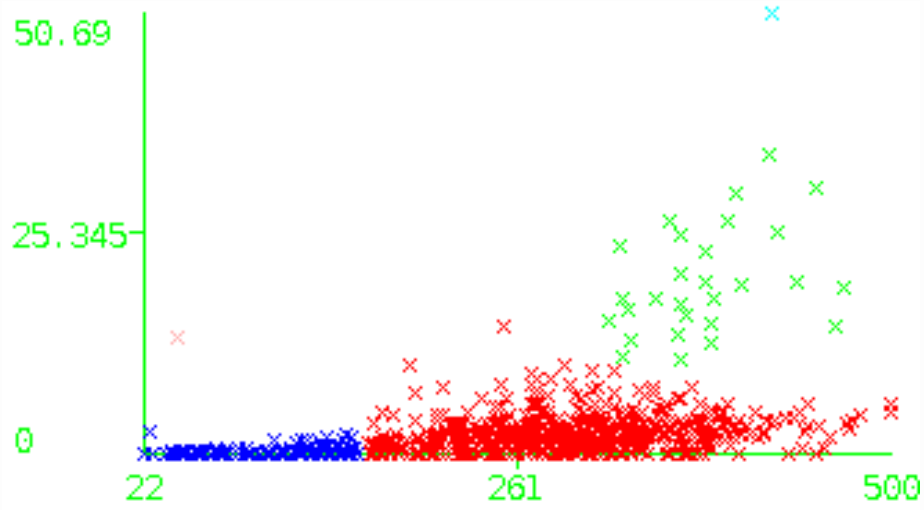


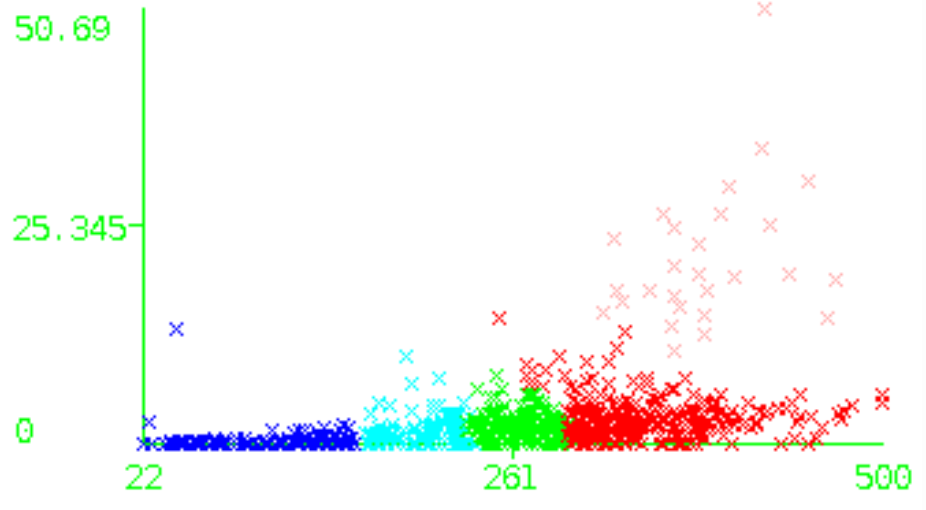
Onko EM-klusterointi järkevä?? (ei kuvassa)

Complete-, single-, average-link ja min. variance









Yleistä asiaa

1. Haluaisitteko työkalujen demosession?
2. Haluaisitteko tehdä itse alustavan arvostelun harjoitustyöllenne?
   Kohdittain pisteet ja töiden jakautuminen ryhmässä, esim.
   - Datan ymmärrys (+perusanalyysi muuttujien arvoista)
   - Esiprosessointi
   - Mallinnus (voi jakaa osiin, jos useita menetelmiä)
   - Tulosten tulkinta
   - Raportointi

Luento 7: ≈ Aputekniikoita harjoitustöihin

1. Tekstidatan kaivelu
   - yleiskuva menetelmistä (DM:n lähialoista)
   - tekstin klusterointi sanojen perusteella
   - HITS-algoritmi (alkujaan internetin kaiveluun)
2. Temporaalisen datan mallinnus
   - erityisesti aikasarjadata!
   - trendien ja periodisuuden tunnistus
   - piirteiden eristys kiihtyvyysdatasta

Tekstidatan kaivelu

Tällä kurssilla: Miten tekstidatasta saisi piirteitä tai ryhmittelyjä helposti? VS.

- Tiedon eristäminen tekstistä (information extraction from text, IET): Halutaan eristää tekstistä automaattisesti vastaukset kysymyksiin tai tiettyjen muuttujien arvot.
- Luonnollisen kielen prosessointi: halutaan ymmärtää tekstin sisältö (merkitys)
- “Tiedon haku” (Information retrieval): halutaan löytää suuresta tietokannasta (esim. internet) kyselyn kannalta relevantit dokumentit

[EXTRA] Esimerkki (IET): Tiivistä tiedot uutisartikkelista

19 March – a bomb went off this morning near a power tower in San Salvador leaving a large part of the city without energy, but no casualties have been reported. According to unofficial sources, the bomb – allegedly detonated by urban guerrilla commandos – blew up a power tower in the northwestern part of San Salvador at 0650 (1250 GMT)

[EXTRA] Esimerkki (IET): Tiivistä tiedot

- Missä? (El Salvador)
- Milloin? (19 Mar)
- Miten? (bombing)
- Ketkä? (urban querilla commandos)
- Kohde? (power tower)
- Uhrit? (no injury or death)

[EXTRA] Monia vaiheita, ongelmaspesifejä menetelmiä

- Erota virkkeet (välimerkkien perusteella)
- Erota sanat (+mahd. luvut)
- Esitä sanan vartalo (stemming) → saman sanan esiintymät
- Tunnista substantiivilausekkeet (noun phrase) “Tämä virke sisältää kaksi substantiivilauseketta.”
- (Lisähankaluudet: yhdyssanat, erisnimet, pronominien viittaussuhteet)
- Tunnista semanttiset suhteet (yleensä vain tietyn tyyppiset relaatiot) “SVM:ien keksijä Vladimir Vapnik” eli Keksijä(SVM,Vapnik)

→ **Mitä saisi selville pelkästään sanoista?**

Tiedon kaivuu annettuna vain sanat

- dokumettien klusterointi sanojen perusteella
- samankaltaisten dokumenttien haku (implisiittinen klusterointi: samanlaiset vs. erilaiset)
- sanojen väliset riippuvuussuhteet →
  - käsitekartat
  - moniosaisten ilmaisujen tunnistus

Sanavartaloiden erottelu

- Mitä sanoja/sanaryhmiä (englannin “yhdyssanat”) tekstissä esiintyy?
- suomen kielen erityisongelma: taivutus – Mikä on sanan vartalo?
  - sanastosta tarkistus (jos saatavilla)
  - heuristiset “stemmerit” (esim. Wekaan saa Snowball-paketissa, mutta suomenkielinen huono)
- Idea: Löytyisivätkö vartalot sanoja klusteroimalla?
  - etäisyysmittana minimum edit distance
  - esim. d(internetissä, internetin)=3 vs. d(kissa,viisas)=3 $\rightarrow$ toimii paremmin pitkillä sanoilla

Dokumenttien klusterointi sanojen perusteella

Ongelma: Klusteroi dokumentit $D = \{d_1, \ldots, d_n\}$, joissa esiintyvät sanat (sanavartalot) $w_1, \ldots, w_k$.

- Karsitaan pois usein toistuvat epäinformatiiviset sanat (stopwords), kuten apunsanat (ja, tai, sekä, hän, se)
- Karsitaan liian harvinaiset sanat (parhaita tosin sellaiset, jotka esiintyvät usein mutta vain osassa dokumentteja)
- Dokumentti esitetään vektorina $(fr(w_1), \ldots fr(w_k))$, missä $fr(w_i)$=sanan w_i frekvenssi kyseisessä dokumentissa

Dokumenttien klusterointi (jatk.)

- Tai $(fr(w_1) \cdot idf(w_i), \ldots, fr(w_k) \cdot idf(w_k))$, missä idf (inverse document frequency) painottaa sanan yleisyydellä

$$idf(w_i) = \log \frac{|D|}{|\{d \in D | w_i \in d\}|}$$

- Lopuksi kannattaa normalisoida vektorit jakamalla vektorin pituudella (jolloin saadaan $\|d_i\| = 1$)

Dokumenttien klusterointi (jatk.)

- Yleisin etäisyysmitta kosinimitta

$$cos(d_1, d_2) = \frac{d_1 \cdot d_2}{\|d_1\| \, \|d_2\|}$$

- dokumenttiklusterin centroidi on vektori $c = (c_1, \ldots, c_k)$, missä c_i on keskiarvo dokumenttivektorien vastaavista painoista
- centroidien etäisyys tai centroidin ja dokumentin etäisyys lasketaan samalla kosinimitalla
- Klusteroidaan halutulla menetelmällä (usein K-means, koska nopea)
- Ongelma: Jos pieniä dokumentteja, vähän yhteisiä sanoja ja toimii huonosti!

Pienien tekstipätkien klusterointi

- Dokumentin esityksessä riittää k:n bitin vektori: esiintyykö sana dokumentissa vai ei?
- Vaihtoehtoinen mitta Jackard:

$$Jackard(d_1, d_2) = 1 - \frac{\#(d_1\text{:ssä ja } d_2\text{:ssa esiintyvät sanat})}{\#(d_1\text{:ssä tai } d_2\text{:ssa esiintyvät sanat})} =$$

$$1 - \frac{|\{w_i | w_i \in d_1 \wedge w_i \in d_2\}|}{|\{w_i | w_i \in d_1 \vee w_i \in d_2\}|}$$

Pienien tekstipätkien klusterointi

Samankaltaisuusverkon ositusmenetelmät (spektraaliklusterointi sukulaisineen) saattavat tuottaa parempia tuloksia!

- **Affinity propagation** – uusi kehuttu menetelmä!
- samankaltaisuusmitaksi dokumenttien välinen Mutual Information
- HITS-algoritmi – helppo ja monikäyttöinen

HITS-algoritmi

Hyperlink-Induced Topic Search (=Hubs and Authorities)

- Alkuperäinen idea web-dokumentien hakemiseen:
  - Hub=sivu joka viittaa moniin autoratiivisiin sivuihin
  - Authority=autoratiivinen sivu johon monet hubit viittaavat
- pävitetään sivujen hub- ja authority-arvoja (h- ja a-vektorit), kunnes systeemi kovergoi
- samalla lasketaan (huomaamatta) vastaavien matriisien prinsipaaliominaisvektorit

[EXTRA] Hits-algoritmi

Olk. $G = (V, E)$ verkko, jonka jokaiseen solmuun v liittyy kaksi painoa $h(v)$ ja $a(v)$.

1. Alusta h ja a tasaisesti s.e. $\sum_i h(v_i)^2 = \sum_i a(v_i)^2 = 1$
2. Kunnes h- ja a-vektorit konvergoivat, päivitä kaikilla v_i:
   - $h(v_i) = \sum_{(v_i,w)\in E} a(w)$ (v_i:n osoittamien auktoriteettien arvojen summa)
   - $a(v_i) = \sum_{(w,v_i)\in E} h(w)$ (v_i:hin osoittavien hubien arvojen summa)
   - Normalisoi painot: $h(v_i) = \frac{h(v_i)}{H}$ ja $a(v_i) = \frac{a(v_i)}{A}$, missä $H = \sqrt{\sum_i h(v_i)^2}$ ja $A = \sqrt{\sum_i a(v_i)^2}$
3. Lopussa parhailla auktoriteeteilla on suurimmat a-arvot ja parhailla hubeilla suurimmat h-arvot

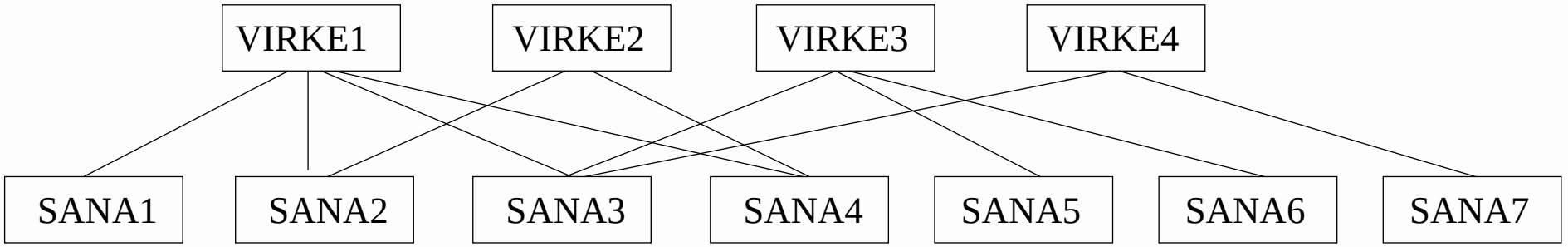
Samankaltaisten sivujen haku

Ongelma: Annettu sivu p, etsi samankaltaisia sivuja.

- Muodosta verkko seuraavista sivuista: kaikki sivut joihin p osoittaa ja osa (tai kaikki), jotka osoittavat p:hen
- Alusta sivujen h ja a-arvot, esim. $h(p) = |(p, q)|$ ja $a(p) = |(q, p)|$ + normalisointi
- Suorita $HITS$
- Lopussa sivu jolla suurin a-arvo on samankaltaisin

Sovellus sanoille ja virkkeille

Annettuna avainsana tai -sanoja, muodostetaan niihin liittyvä 2-jakoinen verkko:



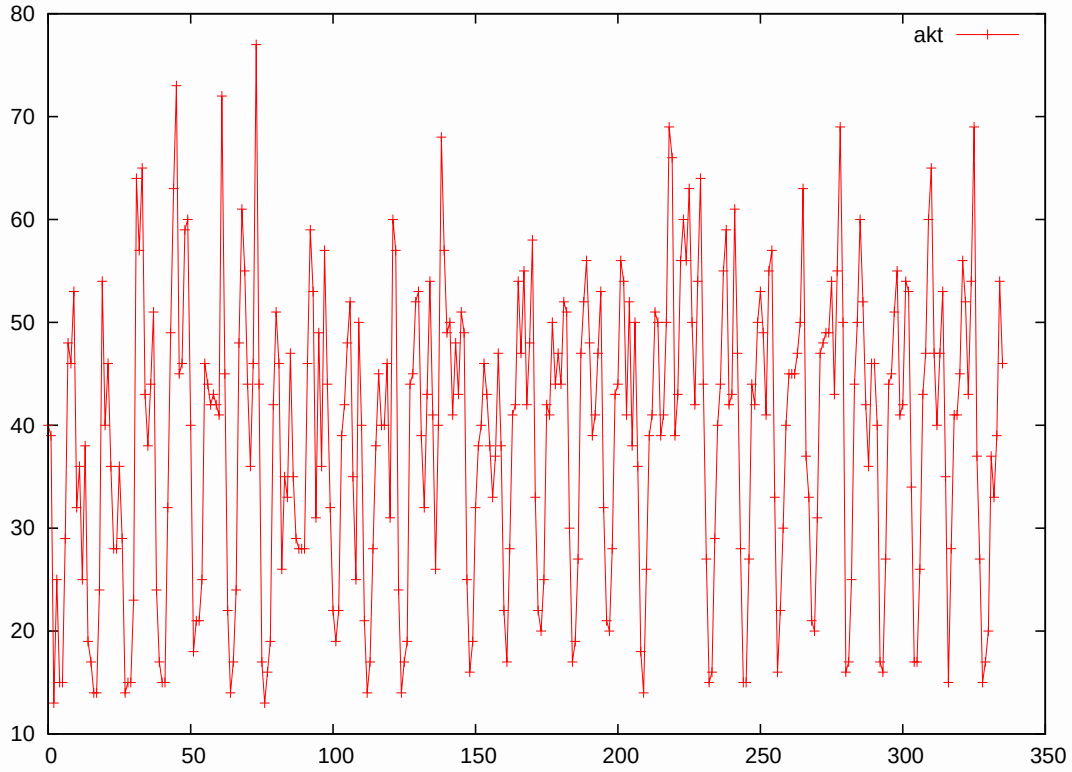
- Virkkeille lasketaan V-painoja ja sanoille S-painoja
- Alustetaan tasaisesti s.e. neliöiden summa 1
- Päivityssääntö: $V(v) = \sum_{w_i \in s} S(w_i)$ ja $S(w) = \sum_{w \in s_i} V(s_i)$ + normalisointi
- Lopussa suurimman S-painon saaneet sanat samankaltaisimpia (ja suurimman V-painon saaneet virkkeet)

Tuloksia

- Testattu suomenkielisillä kissa-aforismeilla ja -tarinoilla
- ei mitään vartalon etsintää tai stopwordin poistoa (olisi kannattanut)
- sanan “kissa” kanssa samankaltaisia mm. koira, ihminen (molemmat eri taivutusmuodoissa), runous, mukava, kotieläin
- sanan “rakastavat” kanssa samanlaisia mm. ruokkivat, pitävät, huolta, täytyy, jumalia
- homonyymeillä voi kiusata ohjelmaa (“Lassie palaa kotiin” vs. “Tuli palaa takassa”)
- lisäksi paljon roskaa! (erit. stopwordeja) → toimisi varmasti paremmin tekstin esiprosessoinnin jälkeen

Aikasarjadatan analyysiä

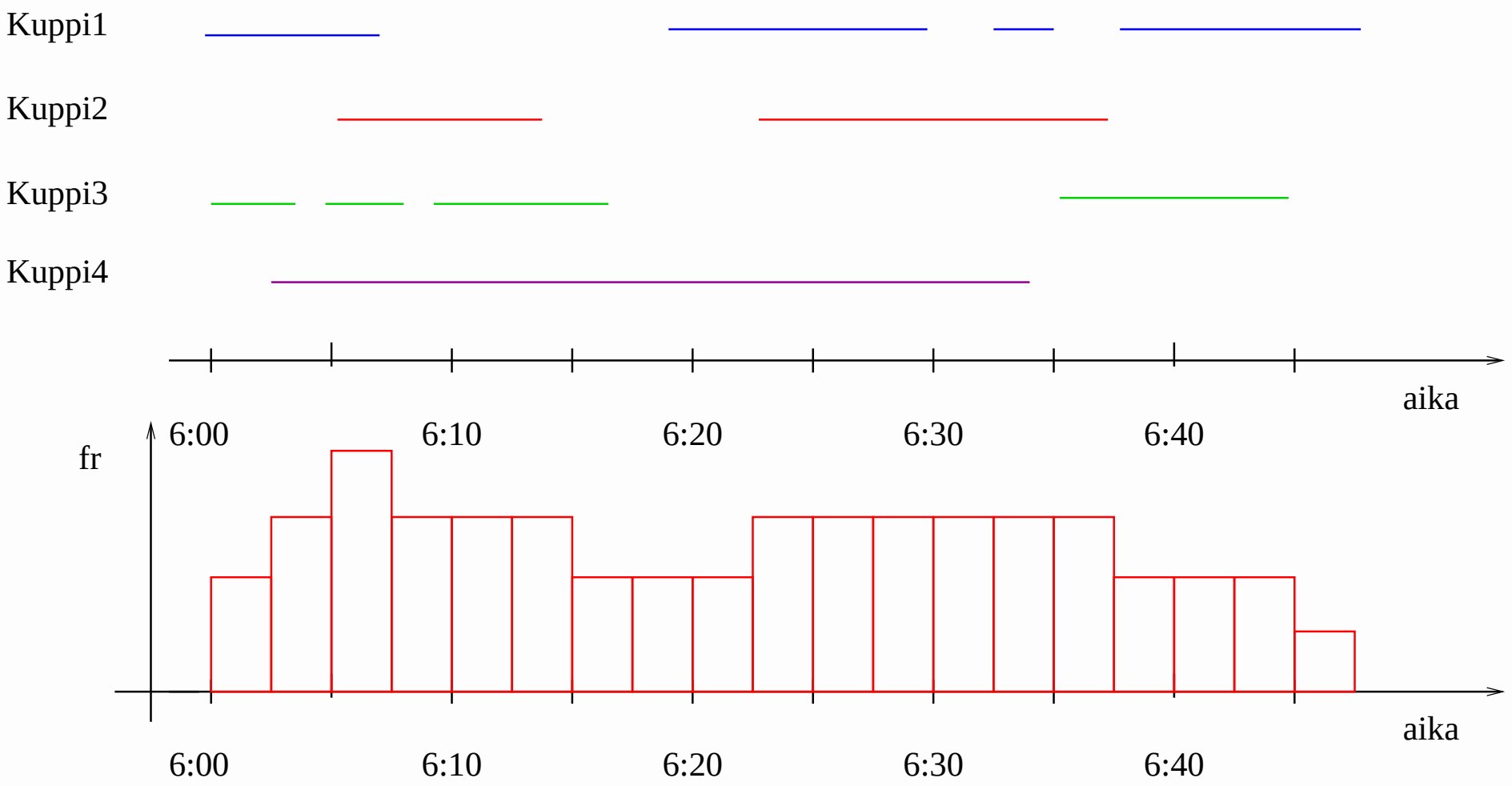
Aikasarja (time series): Jonkin muuuttujan X arvot eri ajanhetkillä. (Kuvassa lehmän aktiivisuus)



Aikasarjadata

- Yleensä X mitattu tasavälein jollain tiheydellä= ”resoluutiolla” (esim. 10 kertaa/s = 10Hz, tai kerran tunnissa)
  - resoluutio=erotustarkkuus
  - tyypillisesti mittaus on ko. mittausvälin keskiarvo
- Lokidatan tapahtumissa mittauksia vain kun jotain tapahtui
- Voi silti muodostaa tasavälisen aikasarjadatan, kun esim. summaa X:n arvot tietyllä aikaikkunalla (esim. minuuteittain, tunneittain)

Esimerkki: syöntifrekvenssi automaateittain



Aikasarjadataa koskevia ongelmia

- Onko nousevaa tai laskevaa **trendiä**?
- Löytyykö säännönmukaisuutta?
  - **periodisuus** (jaksollisuus tai syklisyys): sama hahmo toistuu tasaisin aikavälein uudestaan
  - esim. ulkolämpötilan vaihtelee vuoden sykleissä
  - elollisten käyttäytyminen vuorokauden sykleissä
  - tasaisen kävelyn kiihtyvyyksissä esim. sekunnin sykli
- Millaiset piirteet luonnehtivat luokan C aikasarjoja? → **piirteiden eristys**
- Löytyykö aikasarjasta s lyhyempi aikasarja v? → **aikasarjan sovitus** (subsequence matching)

Ensin kannattaa suodattaa kohinaa

- Liukuva keskiarvo muuntaa x_i:n x'_i:ksi:

$$x'_i = \sum_{j=-m}^{M} w_j x_{i+j}$$

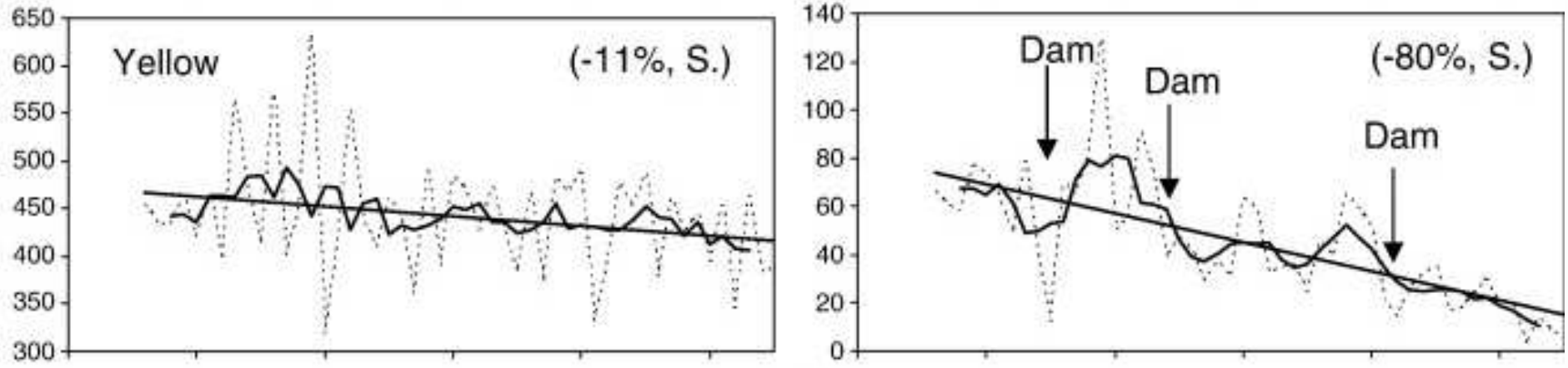
  eli painotettu keskiarvo pisteistä
  $x_{i-m}, x_{i-m+1}, \ldots, x_i, \ldots, x_{i+M}$
- voi olla m=M tai $M = 0$ (vain edelliset pisteet)
- w_j voi olla tasainen (jos pieni ympäristö) tai painottaa kaukaisia naapureita vähemmän
- mediaanisuodatin palauttaa tutkitun välin mediaanin $\rightarrow$ suodattaa häiriöpiikkejä (voivat olla kiintoisia!)
- joskus vaikea erottaa kohinaa “oikeasta” signaalista!

Trendi

- pitkän aikavälin muutos keskiarvossa tai mediaanissa
- viittaa yleensä lineaariseen muutokseen, mutta voi olla esim. polynomi (joskus jopa termi “syklinen trendi”)
- lineaarinen trendi löytyy helposti yhden muuttujan lineaariregressiolla!
- samalla idealla (pienimmän neliösumman minimointi) voi sovittaa muitakin funktioita
- trendi kannattaa poistaa ennen periodisuuden tutkimista!

Trendiesimerkki

Sademäärät ja joen vesimäärän muutokset Keltaisenjoen alueella 1945–2000



Lähde: Xu et al.: Temporal trend of precipitation and runoff in major Chinese Rivers since 1951, Global and Panetary Change 73, pp. 219–232, 2010

Periodisuuden tunnistus

Ongelma: Onko periodeja? Millä aikavälillä toistuvat?

- Helppo tekniikka: **analysoidaan autokorrelaatiot**
  - tavallisen Pearsonin korrelaation sovellus aikasarjalle
- Hieman hankalampaa: **spektraalianalyysi**
  - analysoidaan signaalin “spektriä” eli mistä taajuuskomponenteista se koostuu
  - esim. Fourier- ja Wavelet-analyysi
  - ei toimi aina (herkkä kohinalle tai ulkopuolisille?)

Autokorrelaatio ja korrelogrammit

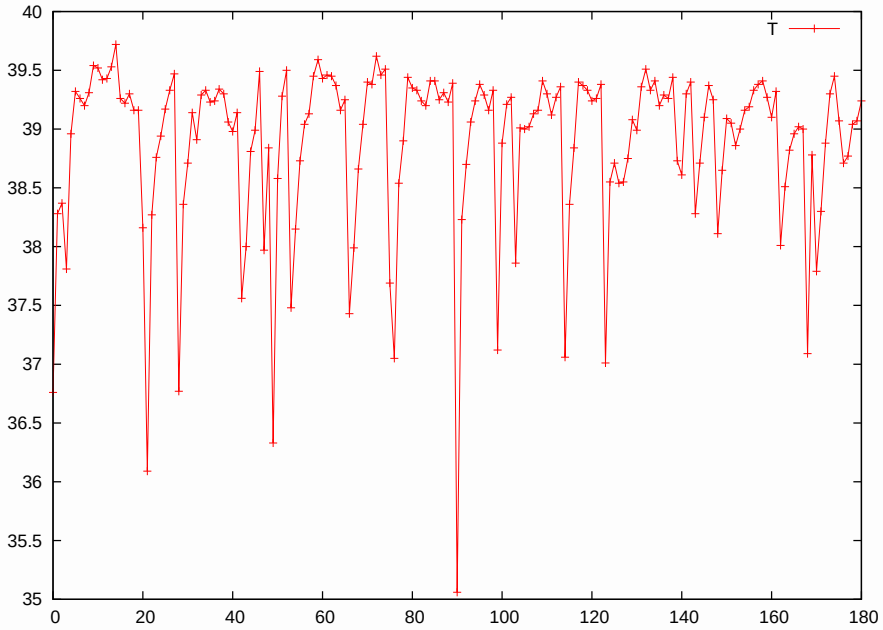
Olkoon aika sarja $x_1, \ldots, x_n$. Yksinkertaistus: Ol. että X:n keskiarvo ja varianssi suurinpiirtein vakiot. Estimaatit: $\mu_X = \frac{1}{n}\sum_i x_i$ ja $\sigma_X^2 = \frac{1}{n}\sum_i (x_i - \mu_X)^2$

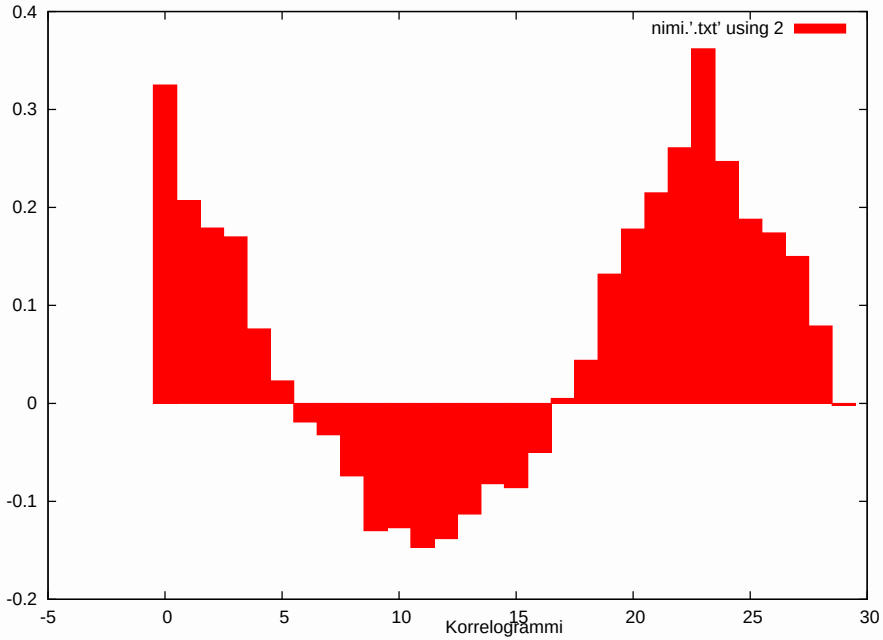
- Idea: Tutkitaan Δ:n päässä toisistaan olevien arvojen korrelaatiota

$$R(\Delta) = \frac{\sum_{i=1}^{n-\Delta}(x_i - \mu_X)(x_{i+\Delta-\mu_X})}{\sigma_X^2}$$

- kannattaa tutkia systemaattisesti eri Δ:n arvoilla (jos periodia ei arvata)
- tulokset voi esittää havainnollisesti **korrelogrammina**

Korrelogrammi: autokorrelaatiot ∆:n funktiona

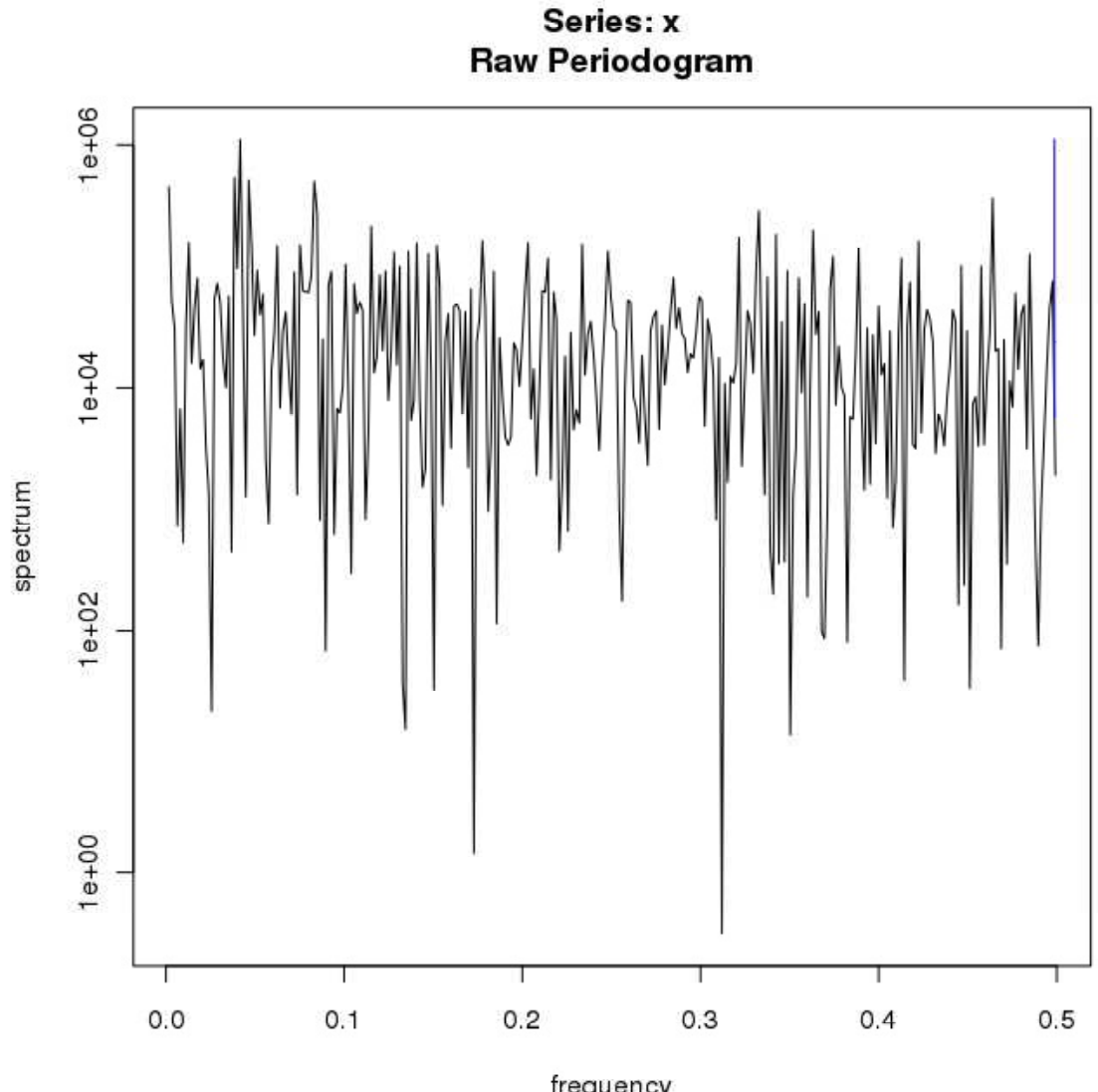




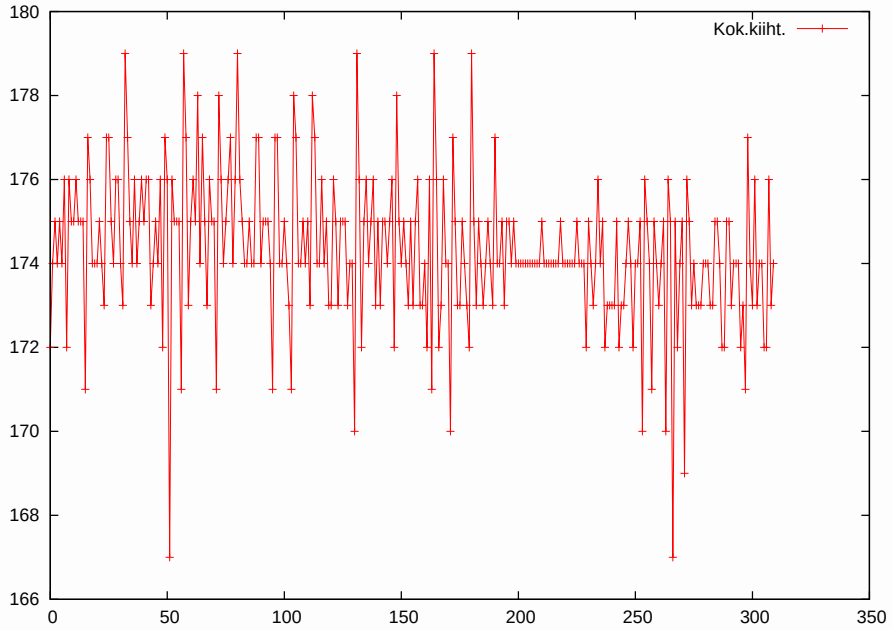
Vasemmalla pötsin lämpötila tunneittain ja oikealla korrelogrammi.

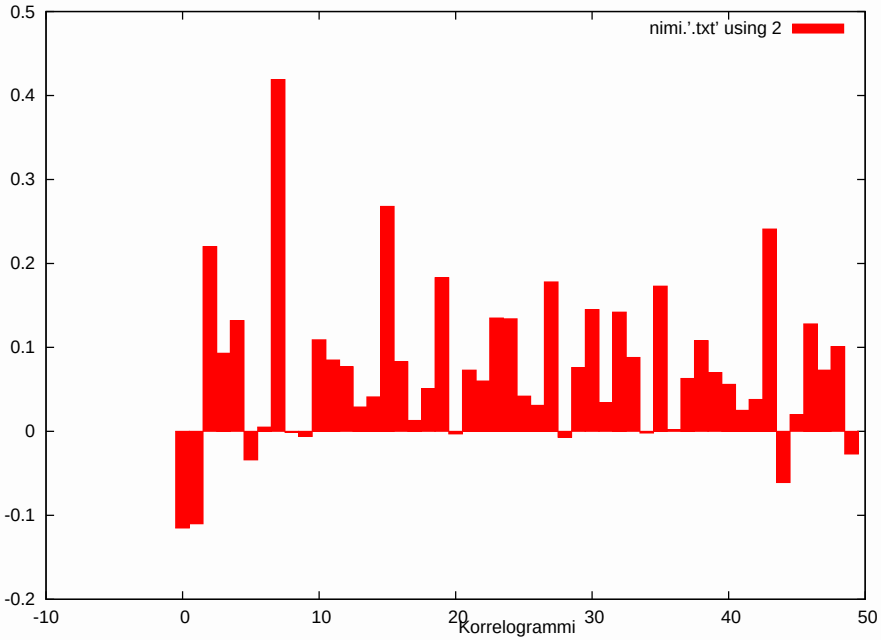
[Ekstra] Miten FT?

Ei selkeitä piikkejä – liian epäsäännöllinen data?

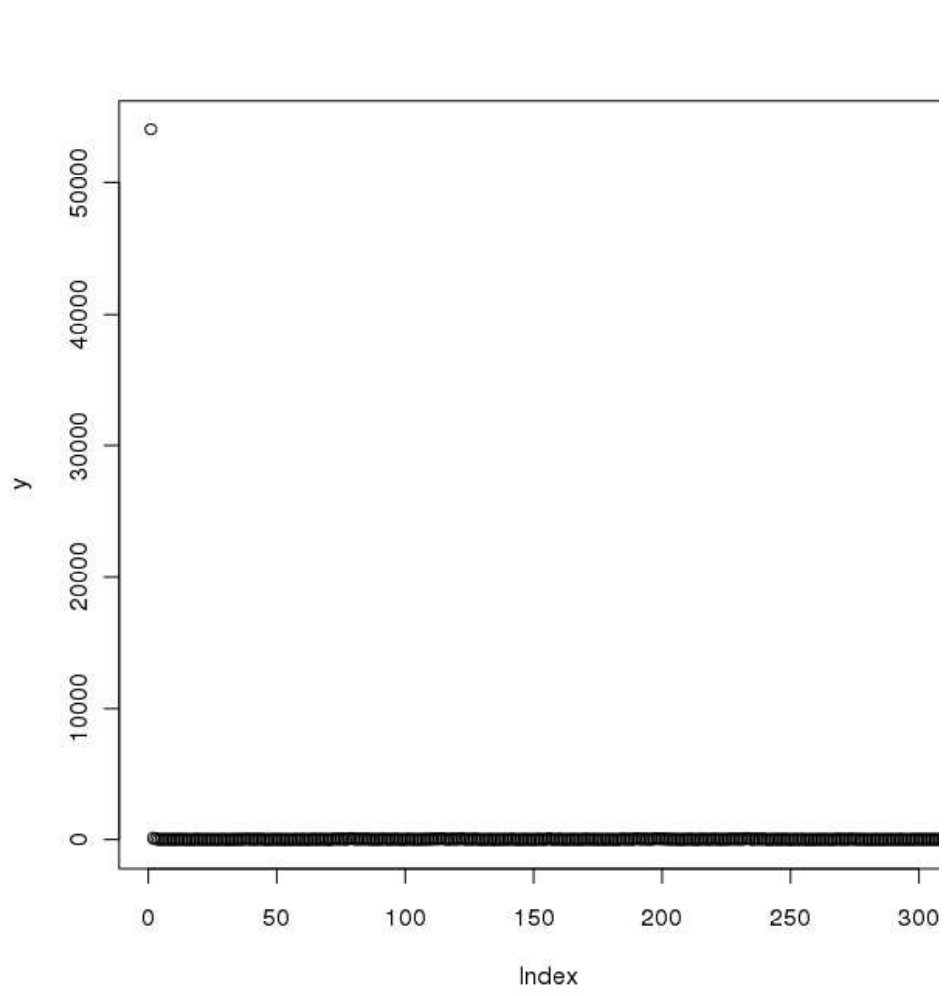


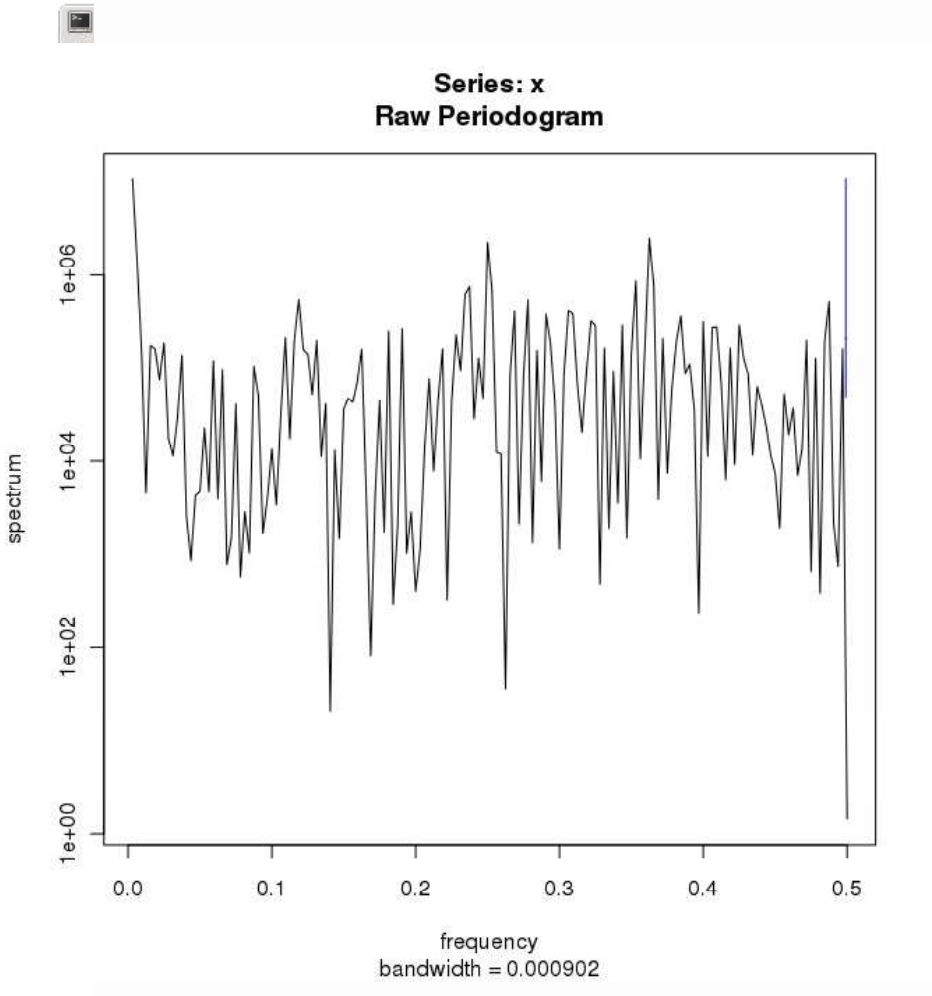
[Extra] Esim.2: Sama märehtimiselle (kiihtyvyysdata)





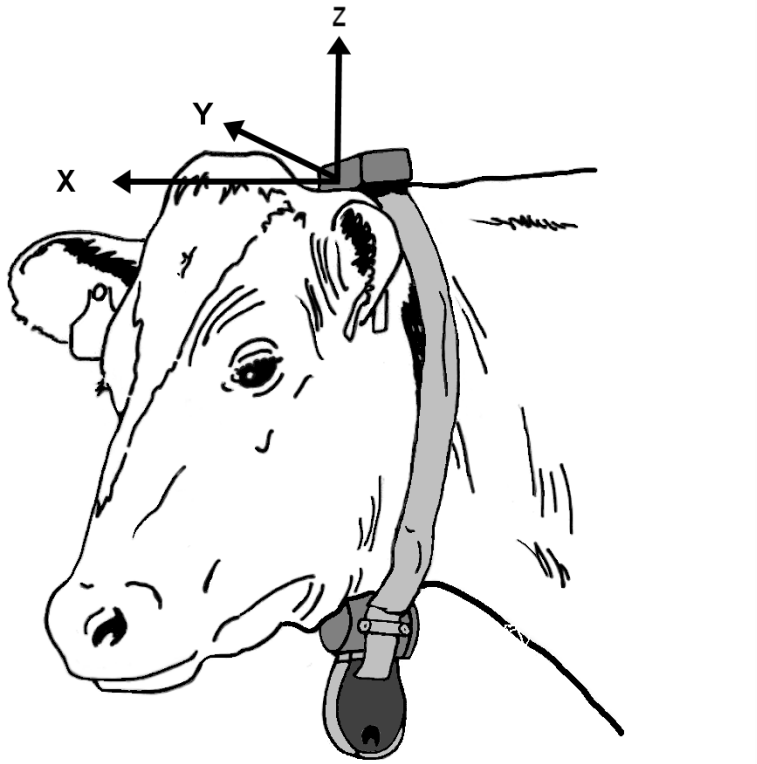
[Extra] Esim. 2: Miten FT?





Piirteiden eristys: esimerkkinä kiihtyvyys

Ongelma: Datasta on kerättyä näytepaloja, esim. 10s pätkiä, joilta on kiihtyvyydet ja joiden aikainen toiminto (luokka) tiedetään. Eristä piirteitä joilla eri toimintoihin liittyvät sarjat voisi erottaa toisistaan!



Tavallisia piirteitä

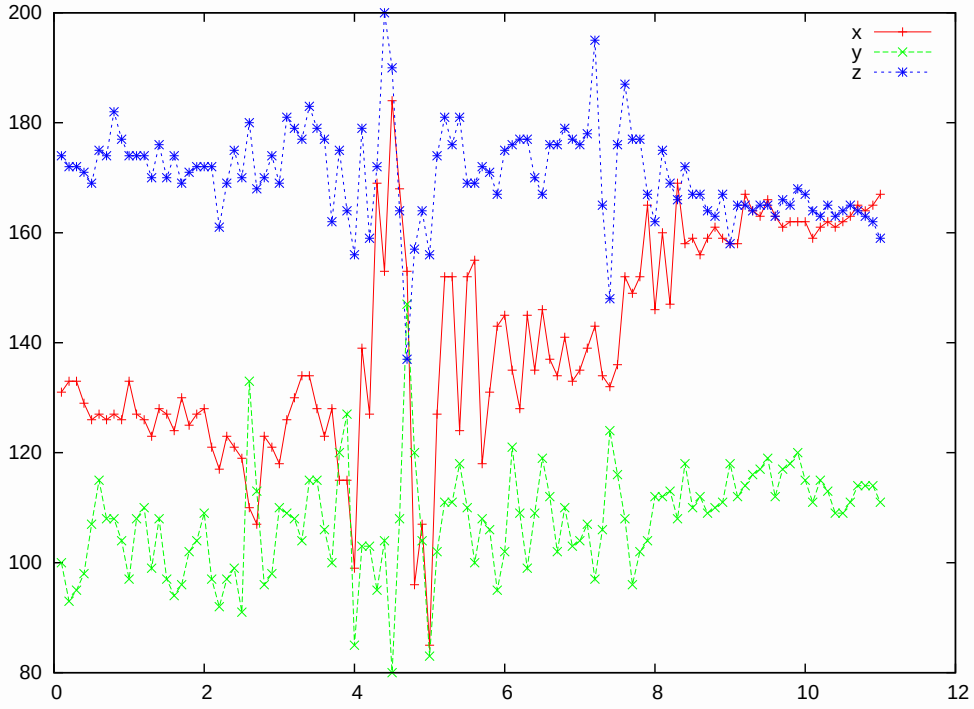
Kukin “data-alkio” on siis osasarja $(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)$.
Ajanhetkellä i kokonaiskiihtyvyys (sen magnitudi)

$$kok = \sqrt{x_i^2 + y_i^2 + z_i^2}.$$

- x:n, y:n ja z:n sekä kokonaiskiihtyvyyden keskiarvo, **keskihajonta**, minimi, maksimi, “energia” (neliösumma)
- joskus myös vinouma (skew) ja kurtoosi (“huipukkuus”, kurtosis)
- jos pätkässä tarpeeksi pisteitä, myös xy-, xz- ja yz-korrelaatiot voivat olla hyviä (mutta herkkiä ulkopuolisille)
- Fourier-analyysistä ensimmäiset komponentit (jos tasaisen periodinen liike)

Heuristisia piirteitä: piikit

- Piikkien lukumäärä, keskimääräinen tai maksimaalinen korkeus, korkeus/leveys-suhde, jne.
- Hyviä kun raju epäsäännöllinen liike!
- Voi tutkia myös muunnetusta signaalista

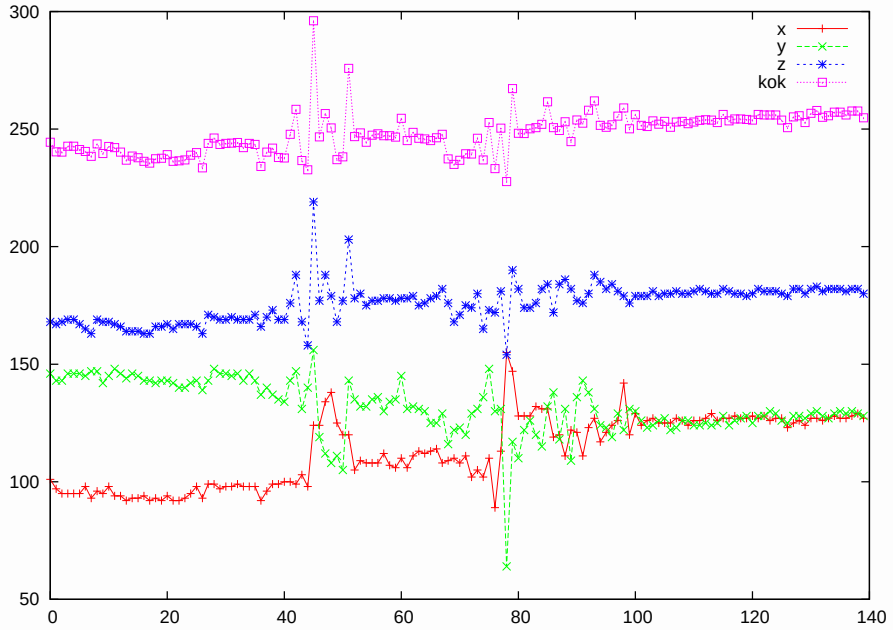


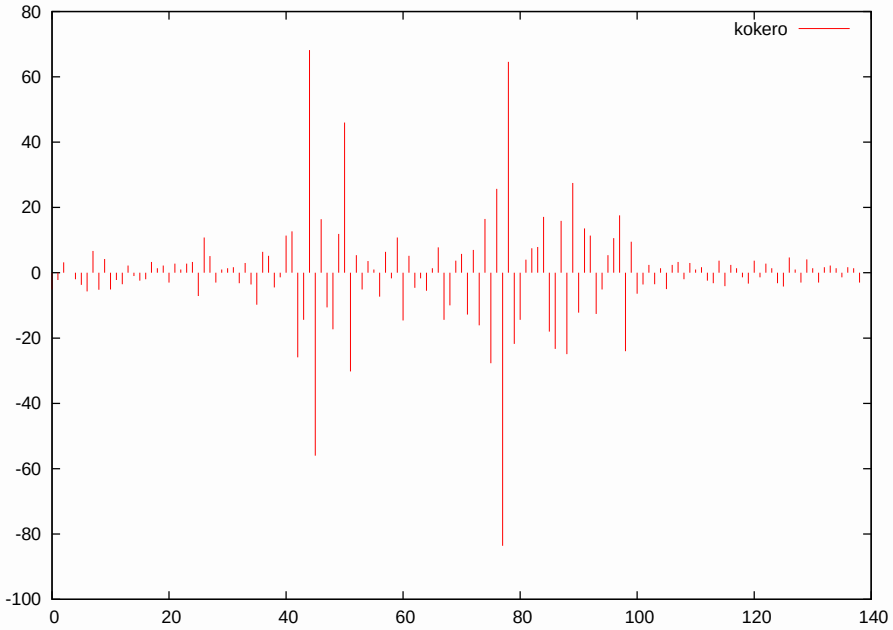
Lehmä nousee ma-kuulta

Muunnettu signaali: kiihtyvyyden muutokset

- Usein informatiivisempaa tutkia kiihtyvyyden magnitudin ja suunnan muutoksia!
- Kokonaiskiihtyvyyden magnitudierot mittarin orientaatiosta riippumattomia!
- kulmaerot vaativat origon tietämistä (arvio riittää)
- kannattaa tutkia myös xy-, xz- ja yz-tasoissa
- muunnetun signaalin “aaltojen” amplitudi ja frekvenssi sekä piikit hyviä piirteitä

Esimerkki: laskeutumisen kokonaiskiihtyvyyden magnitudierot





Tapahtumadatan mallinnusmenetelmiä

Aikasarja muotoa $(t_1, A_1), \ldots, (t_n, A_n)$, missä A_i:t tapahtumia (kategorisia muuttujia)

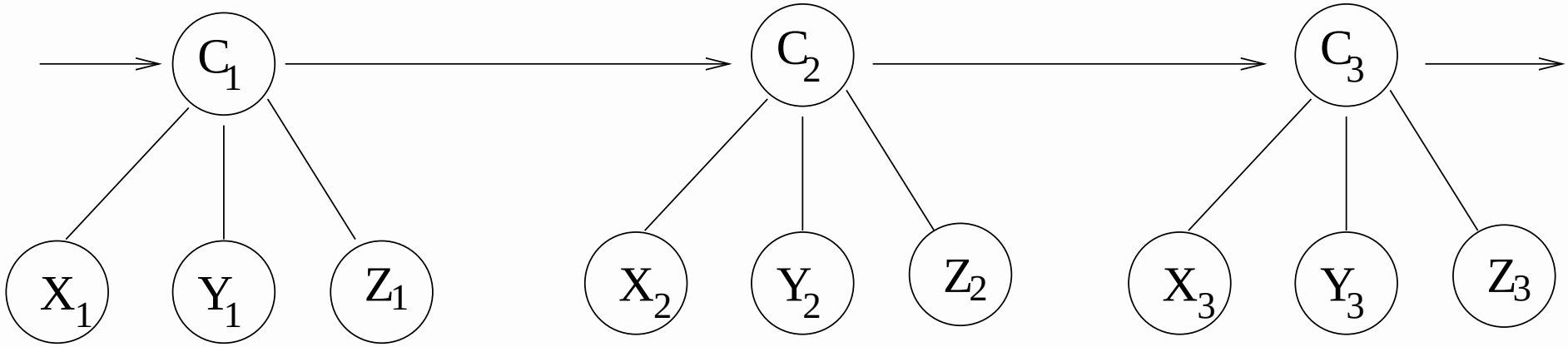
- Markov-ketjut
  - $\approx$ probabilistisia tilakoneita
  - esim. Jos Mansikki juo nyt, millä todennäköisyydellä Mansikki alkaa nuolla seuraavaksi?
  - Ongelma: riippuu siitä, monestiko on jo toistanut juo–nuolee-sykliään

Tapahtumadatan mallinnusmenetelmiä

- Episodit
  - Usein toistuvia tapahtumahahmoja, esim. tapahtumat A ja B esiintyvät peräkkäin tai “rinnakkain”
  - Ongelma: perustuu vain frekvensseihin – onko mitään tilastollista relevanssia?
  - Haku edellyttää että hahmon templaatti on valmiiksi määritelty

HMM:t ja dynaamiset Bayes-verkot (ennustukseen)

Kuvassa dynaaminen naiivi-Bayes-verkko



esim. C= toiminto ja X, Y ja Z kiihtyvyysdatasta eristettyjä piirteitä. Voidaan hyödyntää tietoa edellisestä toiminnosta!

Vinkkejä

- Piirrä (plottaa) paljon kuvia! (usein “puolet” mallinnusta)
- Periodisten hahmojen löytäminen vaatii paljon dataa!
  - esim. eläimillä periodi yleensä vuorokausi eli oltava paljon vuorokausia joka yksilöltä
- Tarjolla monia tutkimushaasteita, esim.
  - tilastotieteellisesti soundi (ja yleisempi) versio episodeista
  - temporaalisen datan louhinta riippuvuussäännöillä (mukaan ajallista kontekstia kuvaavat muuttujat)

Yleistä asiaa 1. Työkalujen demosessio ma 6.5. klo 16.15→

Nyyttikesti-periaatteella! Ohjelmassa ainakin:

- SPSS (Salla ja Sari)
- R (Jonne)
- Cytoscape (verkkojen visualisointi, Antti)
- SOM:it VisualDatalla (Nina)
- Fourier-muunnos ja sen tulkinta matlabilla (Markus)
- joku yleinen visualisointityökalu, ehkä gnuplot

tarpeen mukaan 5–10(–15) min demous, ekstrapisteitä oman valinnan mukaan (esim. 1–3p)

Yleistä 2. Harjoitustyöt

- tästä eteenpäin Wilhelmiina ohjaa kaikkia töitä (Käyttäkää hyväksi! tapaamisia, sposteja, välitulosten tarkastelua, jne.)
  päivystys ti 15-16 (huone F007 tai työpuh. 050-4422472)
- sovitaan lopputapaamiset viikoille 19–20 (kun jo mallinnustuloksia)
- varaudu intensiiviseen työskentelyyn!
- loppuesitykset ja -raportit viikolla 21 (raporttien deadline ke 29.5.)
- arviointi pääosin itsearviointina

Luento 8: Mallien arviointi ja vertailu sekä menetelmien valinta

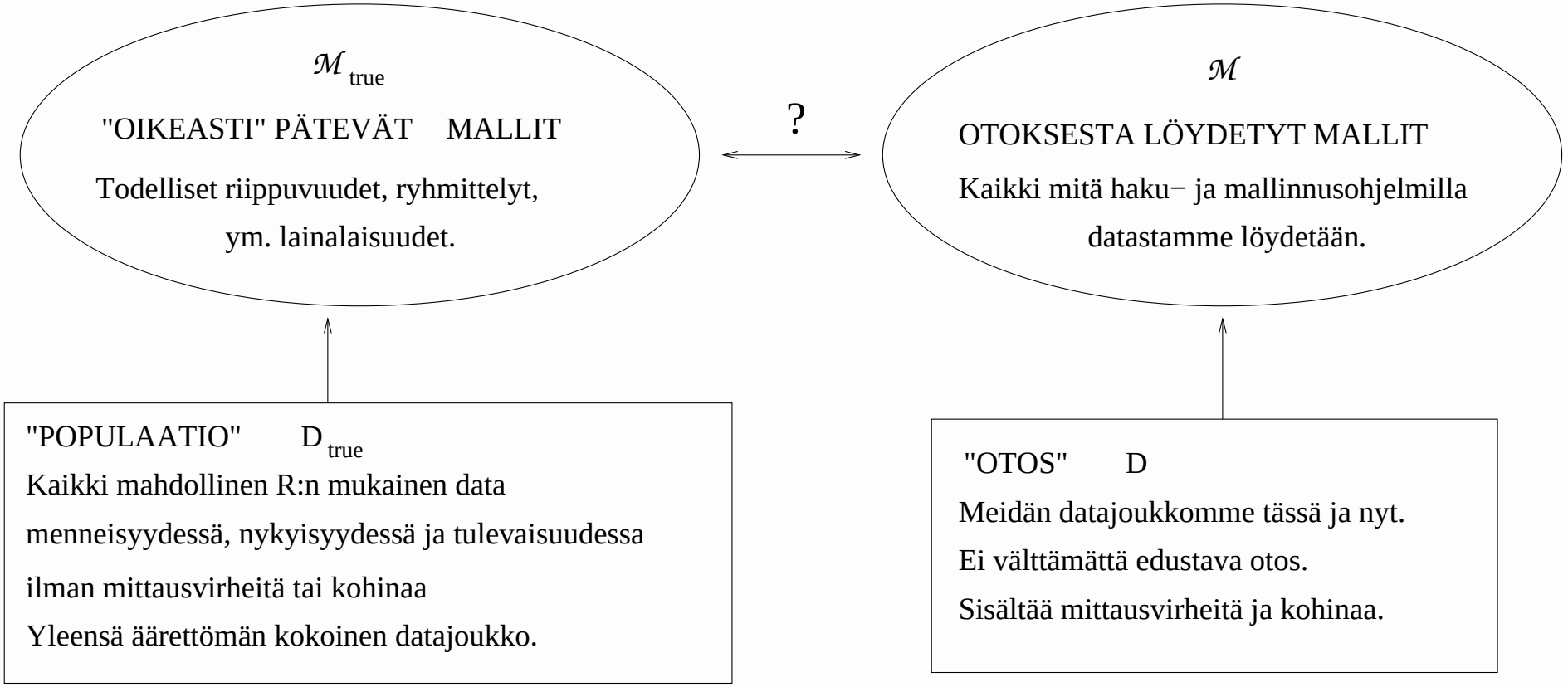
1. Kun löydetty jokin hahmo tai malli
   - Onko se “todellinen” vai vain sattuman tuotetta?
   - Miten hyvä se on?
   - Miten testata sen pätemistä?
2. Annettuna data ja ongelma
   - Mitä luokittelu- tai klusterointimenetelmää kannattaa kokeilla?
   - Miten parametrit kannattaa valita?
   - “Voisiko Sami tehdä ohjelman, joka valitsisi parhaan luokittelijan parametreineen automaattisesti?”

Luento 8: Sisältö

1. Mallien arvioinnin perusongelma
2. Riippuvuuksien systemaattinen haku
   - merkitsevyys ja moninkertaisen testauksen ongelma
3. Luokittelijat
   - Miten arvioida luokittelijan hyvyyttä?
   - validointi ja testaus
   - luokittelumenetelmän valinta
4. Klusterointi
   - Onko löydetty klusterointi “aito”?
   - klusterointimenetelmän valinta

1. Mallien arvioinnin perusongelma

Meillä on vain D, joten voimme muodostaa vain $\mathcal{M}$:n.
Sisältyykö se tai edes osa siitä $\mathcal{M}_{true}$:een?



Kommervenkkejä

Huom! Satunnaisotos=otos jonka alkiot valittu satunnaisesti populaatiosta → todennäköisesti edustava otos

1. **D ei ole satunnaisotos (edustava)**, vaan jotenkin valikoitunut
   - Yleistä DM:ssä! Data usein sivutuotetta, ei kyseiseen mallinnukseen kerätty.
   - löydetään vain kyseistä osapopulaatiota kuvaavia malleja (tai havaitaan, ettei sille löydy malleja)
   - esim. Suurin osa turkistarhan rotista pyydetty talviaikaan. Onko talven takia enemmän vatsahaavaa vai onko turkistarha stressaavampi asuinpaikka?

Kommervenkkejä (jatk.)

2. **D on satunnaisotos, mutta sisältää hirveästi kohinaa** (epätarkat–virheelliset muuttuja-arvot)
   - voi olla oikeita malleja (eli $\mathcal{M}_{true} \neq \emptyset$), mutta ei pystytä löytämään
   - tai löydetään kohinan tuottamia malleja vaikka $\mathcal{M}_{true} = \emptyset$
   - esim. yhden päivän kiihtyvyysdatasta opitaan luokittelija, joka tunnistaa kaikki lehmien toiminnot lähes 100% tarkkuudella, mutta ei kykene tunnistamaan mitään toisen päivän datasta.
   - kohina + ylisovittunut malli paha yhdistelmä!

Kommervenkkejä (jatk.)

3. **D on hyvin pieni**
   - ei voi olla edustava (ellei hyvin triviaali data)
   - ei takeita että mallit (tai niiden puute) pätisivät muille D_{true}:n otoksille
   - esim. Poikivien lehmien toimintodataa vain muutamalta lehmältä ja kultakin vain 1–2 vajaata vuorokautta. $\rightarrow$ Voi tehdä hypoteeseja, mutta ei takeita, että havainnot pätisivät yleisemmin. (=kvalitatiivinen tutkimus, tavoitteena ymmärrys)
   - Huom! Voidaan kyllä **falsifioida** aiemman tutkimuksen väitteitä (jos löytyy vastaesimerkkejä). (Esim. lehmien pesänrakennus)

Ratkaisuja

Deskriptiiviset mallit/hahmot

- Tehdään johtopäätöksiä vain siitä populaatiosta, jolle datamme olisi edustava otos
- Tutkitaan löytöjen tilastollista merkitsevyyttä: millä todennäköisyydellä johtuisivat sattumasta?

Prediktiiviset mallit

- Testataan opetusjoukon ulkopuolisella (siitä riippumattomalla) datalla
- Arvioidaan yleistysvirhettä (=odotettua virhettä ulkopuolisella datalla)

2. Riippuvuksien systemaattinen haku

Ongelma: Etsitään datasta K parasta riippuvuussääntöä $X \rightarrow A$ tai lineaariregressiota $A = \beta_1 A_1 + \ldots \beta_k A_k + \alpha$. Millä todennäköisyydellä hahmot johtuvat sattumasta?

- Arvioidaan yksittäisten hahmojen tilastolliset merkitsevyydet (p-arvot)
- Fisheriläiset vain raportoivat hahmot ja niiden p-arvot
- Neyman-Pearsonilaiset vertaavat jokaista p:tä johonkin valitsemaansa raja-arvoon p_{max} ja kertovat mitkä hahmot olivat merkitseviä merkitsevyystasolla p_{max} (tyypillisesti $p_{max} = 0.05$ tai $p_{max} = 0.01$)
  - Mielivaltaisia raja-arvoja, mutta perinne jatkuu vahvana!

Moninkertaisen testauksen ongelma (multiple testing problem)

- Olk. jokin testisuure T s.e. hahmo johtuu sattumasta todennäköisyydellä $p \leq 0.05$ jos $T \geq T_{0.05}$.
- Hahmo “läpäisee testin” tasolla 0.05, jos sen T-arvo oli $T \geq T_{0.05}$. Muuten se hylätään sattuman tuotteena.
- Jos testataan 1000 hahmoa, jotka kaikki läpäisevät testin, on odotettavissa, että 50 näistä onkin sattuman tuotetta.
- ei ole olemassa hyvää ratkaisua!
- klassinen ratkaisu kiristää tasovaatimuksia, kun testien lukumäärä kasvaa
  esim. Bonferroni-korjaus: Jos testataan m hahmoa, vaaditaankin että $p_{max} = \frac{0.05}{m}$ (voi olla kova vaatimus ja hukata oikeasti merkitseviä hahmoja)

Ongelma: Mittasuureiden esioletukset

Mittasuureet T (esim. χ^2, F, z) vastaavat p-arvoja vain tiettyjen esioletusten pätiessä. Usein epärealistisia!

Esim. χ^2-testi riippuvuussäännöille

- vaatii, että kaikki kombinaatiot $fr(XA)$, $fr(X\neg A), fr(\neg XA)$ ja $fr(\neg X\neg A)$ riittävän suuria
- lisäksi allaoleva hypergeometrinen jakauma ei saa olla liian vino (skewed, käytännössä $fr(A) << 0.5$ tai $fr(A) >> 0.5$)
- Ratkaisu: laske suoraan p_F hypergeometrisesta jakaumasta ("Fisherin eksakti testi") tai käytä Mutual Information-mittaa (robusti, antaa suurinpiirtein samat tulokset)

Esim.2 lineaariregression hyvyys

Monia mittalukuja ja testejä!

Tarkastellaan esimerkkiä: M=rotan paino, V=vartalonpituus ja H= hännän pituus. Halutaan tutkia miten V ja/tai H selittävät M:ää. $n = 50$.

Malli	r^2	p_F	MSE
$\overline{M} = 32.37H - 268.40$	0.67	1.9e-13	1324
$\overline{M} = 24.24V - 271.52$	0.75	1.4e-16	989
$\overline{M} = 14.56H + 16.30V - 332.89$	0.81	4.9e-18	782

Onko "Häntä, Vartalo → Paino" paras malli?

Em. mittalukujen perusteella olisi. Mutta onko ylisovittunut?
→ lasketaan Mallow:in C-mitta:

$$C = \frac{SSE}{MSE_k} - n + 2p,$$

missä p on mallin muuttujien lkm, SSE on mallin virheiden neliösumma ja MSE_k on keskimääräinen virheiden neliösumma kun kaikki k muuttujaa mukana.

Pieni C hyvä. Suositus: Jos mahdollista, valitse malli, jossa $C \approx p$.

Onko "Häntä, Vartalo → ylisovittunut?

Nyt C(vartalo→ paino)=14.8>2.0= C(häntä, vartalo→ paino).

⇒ valitaan "Häntä, Vartalo → Paino"

(Kannattaisi vielä **ristiinvalidoida**!)

Ks. esim. Milton ja Arnold: Introduction to probability and statistics: Principles and applications for engineering and computing sciences.

3. Luokittelijat

Ongelma: Datasta opittu hieno luokittelija, joka luokittelee yli 81% rotista oikein vatsahaavallisiin tai -haavattomiin, pelkästään vuodenajan, paikan ja ruumiin mittojen perusteella. Lisäksi oikeellisuus on määritetty **ristiinvalidoinnilla**, joten sen pitäisi kuvata mallin hyvyyttä opetusjoukon ulkopuolella.

Eikö Liisan pitäisi julkaista heti merkittävä tutkimustulos?? “Rottien vatsahaavan salat selvitetty”

Katsotaan sekaannusmatriisia

		Luokittelijan mukaan		
		Vatsa-haava	Ei vatsa-haavaa	Σ
Oikeasti	Vatsa-haava	TP=0	FP=50	TP+FP=50
	Ei vatsa-haavaa	FN=0	TN=217	FN+TN=217
	Σ	TP+FN=0	FP+TN=267	n=267

Tähän ei tarvitsisi mitään luokittelijaa! (Oikeasti oli 81.3% vatsahaavattomia)

$\kappa = 0$ olisi paljastanut että luokittelijan hyvyys sattumaa.

Luokittelun hyvyyden mittoja: $accuracy$ *ja* κ

Olk. $M(x_i)$ luokittelijan määräämä luokka pisteelle x_i ja c_i sen oikea luokka. (Kaavoissa ol. binääriluokittelu)

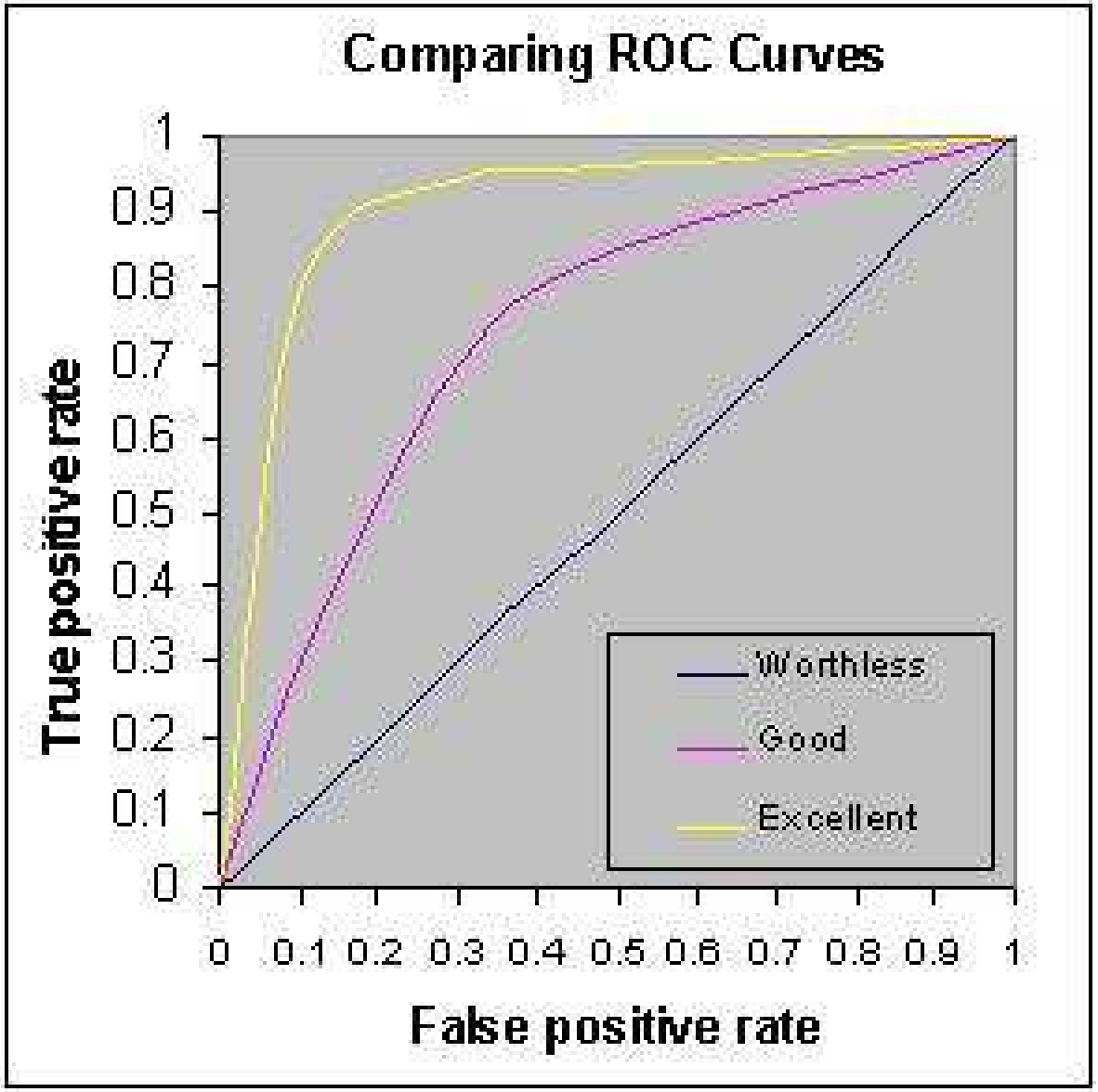
$$accuracy = \frac{\sum_{i=1}^{n} I(M(x_i) = c_i)}{n} = \frac{TP + TN}{n}$$

(I=indentiteettifunktio eli $I(M(x_i) = c_i) = 1$ kun $M(x_i) = c_i$ ja $I(M(x_i) = c_i) = 0$ muuten). $err = 1 - accuracy$

$$\kappa = \frac{n(TP + TN) - fr(C)(TP + FP) - fr(\neg C)(TN + FN)}{n^2 - fr(C)(TP + FP) - fr(\neg C)(TN + FN)}$$

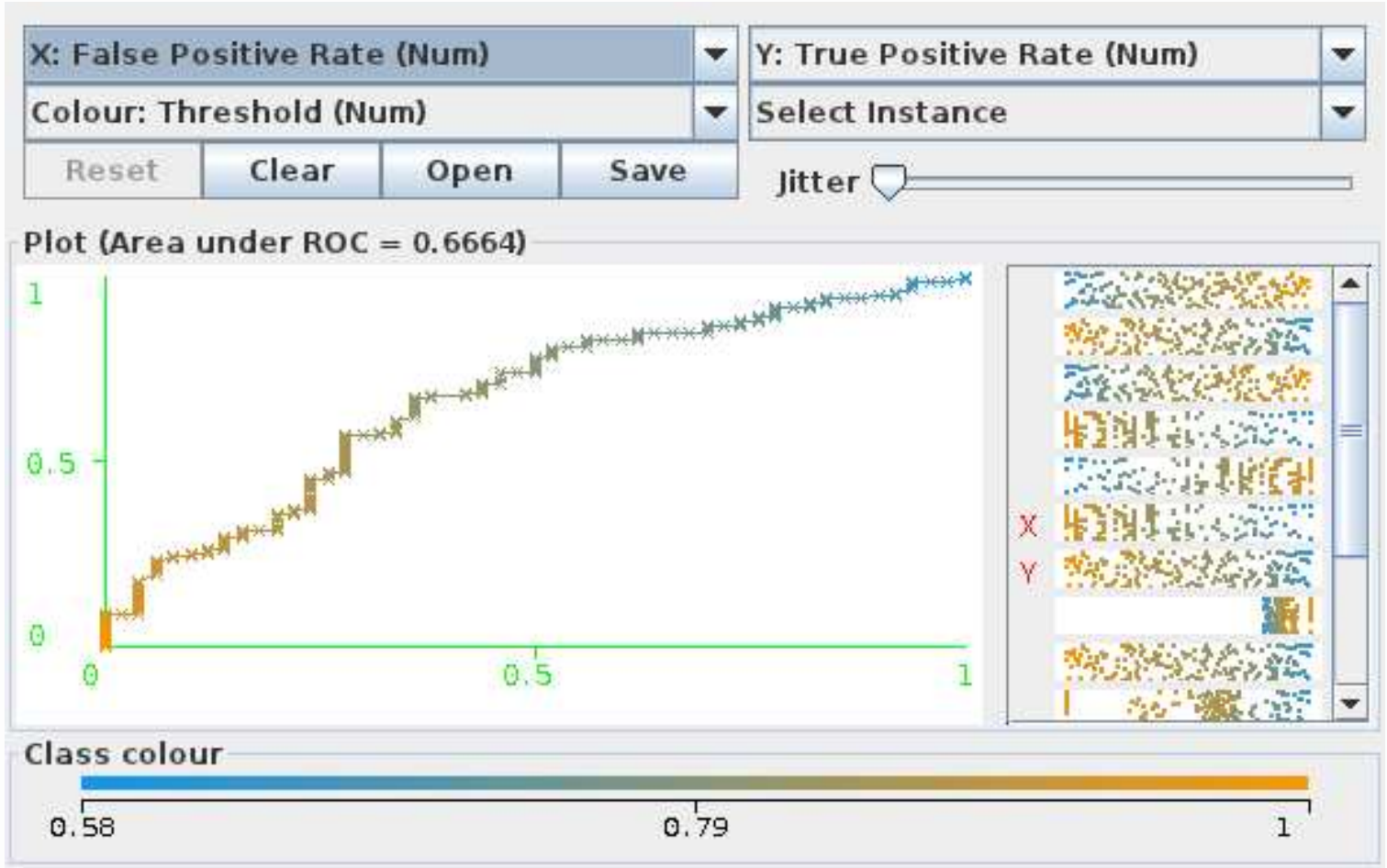
(tutkii monastiko luokittelija ja todellisuus samaa mieltä, ilman että se olisi sattumaa) Suositus: oltava $\kappa > 0.4$.

Luokittelun hyvyyden mittoja: ROC-area (AUC)



AUC Wekassa

Klikkaa tulosta ja valitse Visualize threshold curve. Katso että x-akselilla false positive rate ja y-akselilla true positive rate.



Validointi ja testaus

Termejä käytetään sekaisin!

Perusmenetelmät:

1. Valitaan satunnaisesti osa (esim. 1/3) datasta testaukseen ja loput mallin oppimiseen.
   - Ongelma: Oppimiseen jää vähemmän dataa ja voi tulla huonompi malli.
   - Testidatassa ei saisi esiintyä samoja alkioita kuin opetusdatassa! (duplikaatit ja melkein duplikaatit ongelma) **Miksi?**

Validointi ja testaus (jatk.)

2. m-kertainen ristiinvalidointi (cross validation):
   - Jaa data satunnaisesti m:ään suurinpiirtein yhtä suureen palaan. Olkoon palat $d_1, \ldots, d_m$.
   - tyypillisesti $m = 10$, mutta pienessä datassa voi olla $m = n$ (Leave-one-out cross validation, LOOCV)
   - Toista mallin muodostus ja testaus m kertaa: Kerralla i valitaan testijoukoksi d_i ja lopuista opitaan malli.
   - Laske virheiden (tms. mittalukujen) keskiarvot.

Validointi ja testaus (jatk.)

- Joskus luokittelijan oppimisalgoritmi käyttää itse erillistä validointijoukkoa esim. muuttujien valintaan
- Tätä joukkoa ei saa käyttää testaukseen! (on jo ollut mukana mallin muodostuksessa)
- Testijoukon pitäisi arvioida miten hyvin luokittelija toimii muodostusdatan ulkopuolisella datalla (off-sample data)
- Suositus: Luokittelijan valintaan (monesta vaihtoehdosta) ja testaukseen eri datat!

Vinkkejä

- Jos yleistysvirhe (testidatalla tai ristiinvalidoinnilla) hirmuhyvä, kannattaa epäillä! (bugi tai valikoitunut data)
- Jos lasket datan epäkonsistenttien rivien osuuden, saat alarajan parhaalle mahdolliselle virheelle (olettaen että sama epäkonsistenssi oikeassa populaatiossa)
- Tutki aina mitä olisi luokitteluvirhe puhtaalla arvauksella! (pitää saada sitä parempi)
- luokkien epätasainen jakauma voi aiheuttaa ongelmia $\rightarrow$ tasoita

Luokittelumenetelmän valinta

“Voisiko Sami tehdä ohjelman, joka valitsisi parhaan luokittelijan parametreineen automaattisesti?”

- Universaalisti parasta luokittelumenetelmää ei ole! (Ei edes satunnaisarvausta parempaa – sanoo **“No free lunch” -teoreema**!)
- Sopivimman menetelmän valinta heuristista ja riippuu esitietämyksestä (data, ongelma-alue, tavoitteet)
- jopa parhaat luokittelijoiden oppimisalgoritmit tuottavat huonoja tuloksia joissain ongelmissa!

$\Rightarrow$ **Tiedonlouhijan hallittava valikoima erilaisia menetelmiä**

[Extra] No free lunch -teoreema

Olk. F opittava kohdefunktio joka liittää johonkin muuttujajoukkoon X luokan C.

Olk. $E(err|F, d, A_i)$= virheen odotusarvo algoritmilla A_i ja opetusdatalla d, kun odotusarvo yli kaikkien mahd. funktioiden F.

NFL: Jos kaikki F:t yhtä todennäköisiä, niin kaikille algoritmeilla A_1 ja A_2 pätee $E(err|F, d, A_1) = E(err|F, d, A_2)$

- toinen algoritmeista voisi olla vaikka satunnaisarvaaja!

[Extra] No free lunch -teoreema

- Jos F:ien todennäköisyydet vaihtelevat, lasketaan odotusarvo yli kaikkien $P(F)$:ien, jolloin pätee $E(err|d, A_1) = E(err|d, A_2)$
- Samat tulokset pätevät, kun ehdollistetaan datan koolla n itse datan d sijasta.
- Yllätys: Vaikka tuntisimme d:n (mutta emme F:ää), ei mikään algoritmi ole parempi toista.

“Paha ei ole kenkään algoritminen, vaan toinen on heikompi toista.”

Opas luokittelumenetelmän valintaan

Ominaisuus	Päätöspuu	Naiivi-Bayes	Yl. Bayes-luokittelija	Neuro-verkko	SVM	*K* lähintä naapuria
Datan tyyppi numeerinen						
kategorinen						
sekadata						
Datan laatu sis. ulkopuolisia						
kohinaa						
puuttuvia arvoja						

Opas luokittelumenetelmän valintaan

Ominaisuus	Päätöspuu	Naiivi-Bayes	Yl. Bayes-luokittelija	Neuro-verkko	SVM	K lähintä naapuria
Datan määrä vähän						
Tehokkuus luokittelijan oppiminen						
luokittelu						
luokittelijan päivitys						

Opas luokittelumenetelmän valintaan

Ominaisuus	Päätöspuu	Naiivi-Bayes	Yl. Bayes-luokittelija	Neuro-verkko	SVM	*K* lähintä naapuria
Ilmaisuvoimainen (epälin. lk-rajat)						
Havainnollinen						

4. Klusteroinnin arviointi

Ongelma: Onko datassa klustereita? Montako? Johtuvatko sattumasta? Miten klusteroinnin tilastollista merkitsevyyttä pitäisi arvioida?

- Ei kunnollista teoriaa!
- Eräs tapa iteratiivinen testaus (eri K:n arvoille)
  - Olk. $M(K)$ klusteroinnin hyvyysmitan paras arvo, kun K klusteria
  - Ol. että koko data yhtä normaalijakautunutta klusteria
  - Lasketaan $M(2)$:n **jakauma** tällä oletuksella ja verrataan saavutettua M-arvoa siihen $\rightarrow$ saadaan todennäköisyys, että näin hyvä $M(2)$-arvo johtuisi sattumasta

4. Klusteroinnin arviointi (jatk.)

- Mikäli 2 klusteria oli uskottavaa, tehdään siitä uusi nollahypoteesi (oletus)
- Lasketaan $M(3)$:n jakauma tällä oletuksella jne.
- Ongelma: $M(K)$:n jakaumaa ei voi määrittää tarkasti, vain karkeita arvioita! (Ks. Duda et al. s. 558)
- Malliperustainen klusterointi määrittää datan uskottavuuden annettuna klusterointi!
- lisäksi monia heuristisia klusteroinnin hyvyysmittoja $\rightarrow$ voi käyttää optimaalisen K:n määritykseen (mahd. käännekohdasta K)

Vinkkejä

- Jos muuttujat eri skaaloissa (tai eri suureita), skaalaa samalle välille
- Kokeile monia eri menetelmiä ja kutakin useita kertoja (välillä data sotketaan, jos järjestyksestä riippuvainen menetelmä)
- Jos vaihtelu vähäistä, voi ainakin klusterien ydinalueita pitää uskottavina
- Klusterointia voi käyttää piirteiden valintaan luokittelua varten (valitaan piirteet jotka tuottavat homogeenisia klustereita) (toimii erityisesti jos K – NN-luokittelija)

Opas klusterointimenetelmän valintaan

Ominaisuus	K-means	K-medians	Single-link	Complete-	Average-link	Centroid-link	Minimum variance	Model-based (+Norm. jak.)
Klusterit muoto vapaa								
voi olla erikokoisia								
päällekäisiä								

Opas klusterointimenetelmän valintaan

Ominaisuus	K-means	K-medians	Single-link	Complete-link	Average-link	Centroid-link	Minimum variance	Model-based (+Norm. jak.)
Datan laatu sietää kohinaa								
ulkopuolisia								
epärelevantteja muuttujia								

Opas klusterointimenetelmän valintaan

Ominaisuus	K-means	K-medians	Single-link	Complete-link	Average-link	Centroid-link	Minimum variance	Model-based (+Norm. jak.)
Ei riipu datan järj.								
Tehokas (kun n iso)								