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

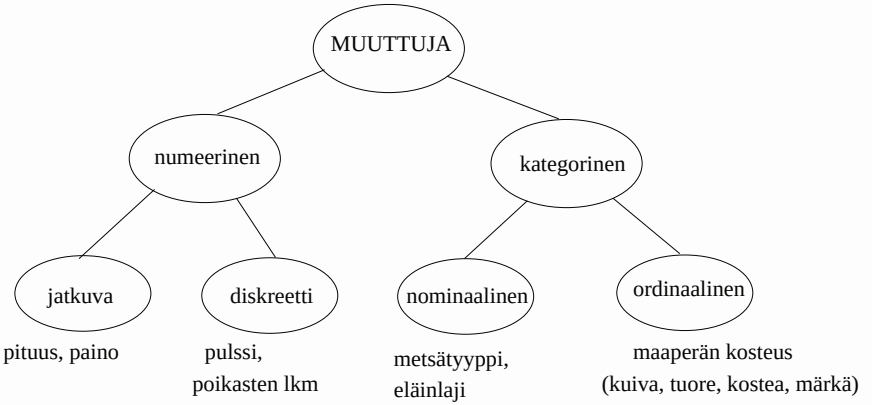
Staattinen ja temporaalinen data

- 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 $\rightarrow$ 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 $\rightarrow$ 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)

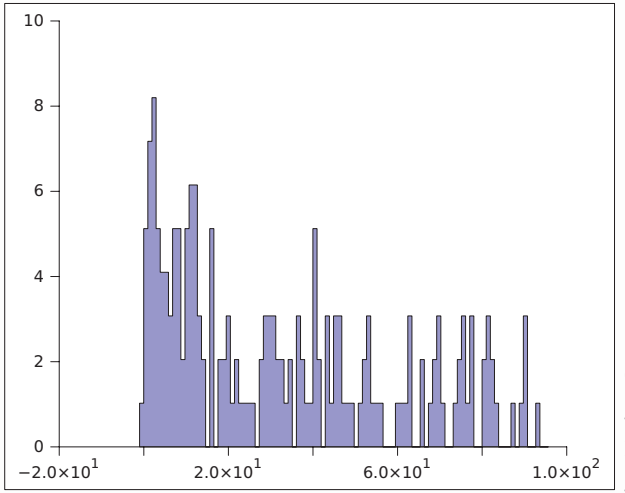
- kertoo että ongelma on laskennallisesti *** vaikea!
- vähintään yhtä vaikea kuin aikavaativuusluokan NP (non-deterministic polynomial time) ongelmat
- todennäköisesti eksponentiaalinen aikavaativuus eli aika-askelten lukumäärä kasvaa eksponentiaalisesti syötteen koon funktiona
- esim. jos ohjelman 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"

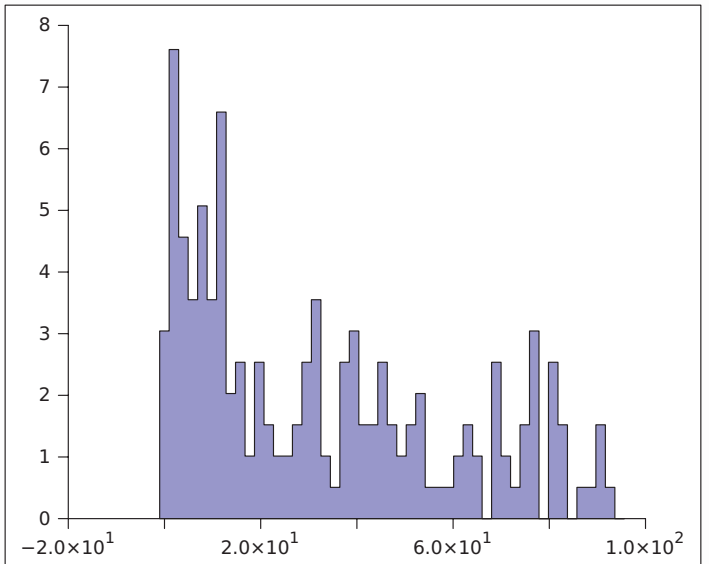
1. Valitse aluksi suuri osavälien (bins) lukumäärä, esim. 100
2. Muodosta histogrammi ja esitä graafisesti
3. Jos histogrammissa selviä kukkuloita ja välillä selviä aukkoja, voi aukkojen kohdalle asettaa diskretointirajan
4. Jos matala tasainen jakauma, vähennä osavälien lkm:ää
5. Jos korkea, mutta ei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

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

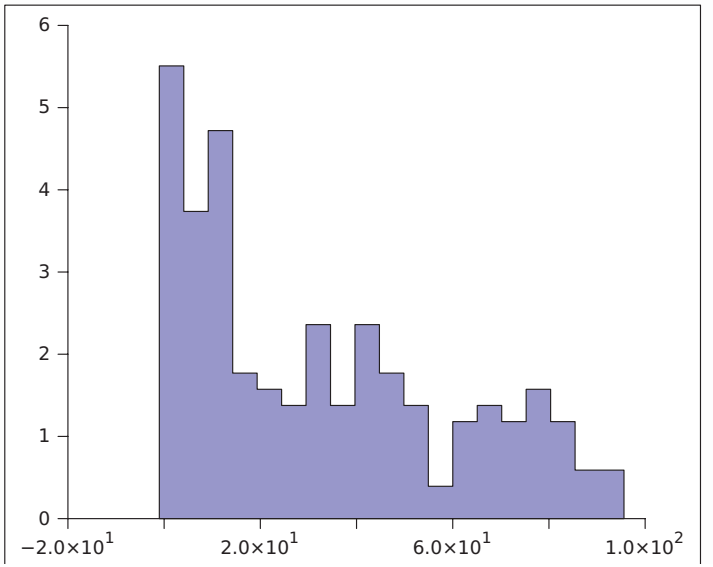


esim. 3-jako:
$x < 25$,
$25 \leq x < 58$,
$x \geq 58$

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)