Business from technology

Tehokas on-line monitorointi

Goodnet päätösseminaari 19.10.2012
Jorma Kilpi
VTT Technical Research Centre of Finland

Taustaa

- GoodNet projektin puitteissa on ajateltu ensisijaisesti verkon monitorointia Y.1731:n avulla
  - **Resilientti** Ethernet-verkko, jossa vähintään kaksi MEPpiä
  - MEPpien välillä on koko ajan mittausta
    - Kehyksen (edestakainen) viive (**frame delay**)
    - Kehysten hävikki (**frame loss**)
- Esitetään metodi viivedatan muodostaman aikasarjan (**time series**) käsittelyyn
- Esiteltävän metodin sovellusalue on kuitenkin laajempi:
  - Tarvittava perusoletus on **stationaarisista** jaksoista koostuva havaintolähde

Mitä tarkoittaa ”tehokas on-line”?

- **On-line:** Mittaushavaintoa ei talleteta, mutta se prosessoidaan ja riittävä määrä informaatiota havainnosta tallennetaan kiinteän kokoiseen tietorakenteeseen ennen kuin uusi havainto tulee.
  - Kyseessä on siis *häviöllinen tiedonpakkaus* (**lossy compression**).
  - Toinen tähän asiayhteyteen liittyvä termi on *sarjallinen estimointi* (**sequential estimation**)
- Tehokkuudella tarkoitetaan
  - että relevantin **informaation** hävikki on mahdollisimman pieni ja
  - että yhden havainnon prosessointi on vakioaikaista (*O(1)*)

Mitä on stationaarisuus?

- Aikasarjan, eli ajan mukaan järjestyksessä havaitun datan analyysissä joudutaan tekemään *stationaarisuusoletus:*
  - **Se tilastollinen jakauma joka kuvaa havaintojen satunnaisuutta ei muutu kun aika kuluu**
- Havaintojärjestys voi silti sisältää informaatiota lähteen satunnaisuuden luonteesta (tilastollinen riippuvuus)
- Ilman tätä oletusta tilastollisilla tunnusluvuilla (**statistic**) ei ole oikein mitään perusteltua *tulkintaa*.
  - Jos stationaarisuus ei päde, niin aikasarjasta pitäisi pystyä eliminoimaan trendit ja periodiset komponentit (on-line) **ennen** kuin siitä lasketaan mitään tunnuslukuja

Miksi stationaarisuusoletus voidaan tehdä?

- Kun mitataan viivettä kahden MEPin välillä, niin viipeen jakauma riippuu käytetystä polusta
- Vikasietoisessa Ethernet-verkossa reitityksen oletetaan tapahtuvan topologian virittävän puun avulla (**Spanning Tree Protocol, STP**)
  - Polku ei yleensä vaihdu (stationaarinen tila) ja kun se vaihtuu niin havaitaan jokin anomalia
  - Myös kehysten hävikki riippuu polusta, erityisesti STP:n aiheuttamista polun muutoksista
- Verkon ja polun säännöllisellä kuorman vaihtelulla on suhteellisen pieni vaikutus viipeen jakaumaan, koska kuormitusta resilientissä verkossa harvoin päästetään liian suureksi.

Miksi monitorointidataa yleensä tallennetaan?

- **Palvelun taso (Service Level Agreement, SLA)**
  - Eri asiakkaille eri SLA:t, eri laskutus
  - Verifiointi jälkikäteen
- **Tietoturva**
  - Tietoturvaan liittyvät anomaliat
- **Luotettavuus**
  - Luotettavuuteen liittyvät anomaliat
    - GoodNet projektin näkökulma verkkoon: halutaan parempaa dataa luotettavuuteen liittyvistä ilmiöistä

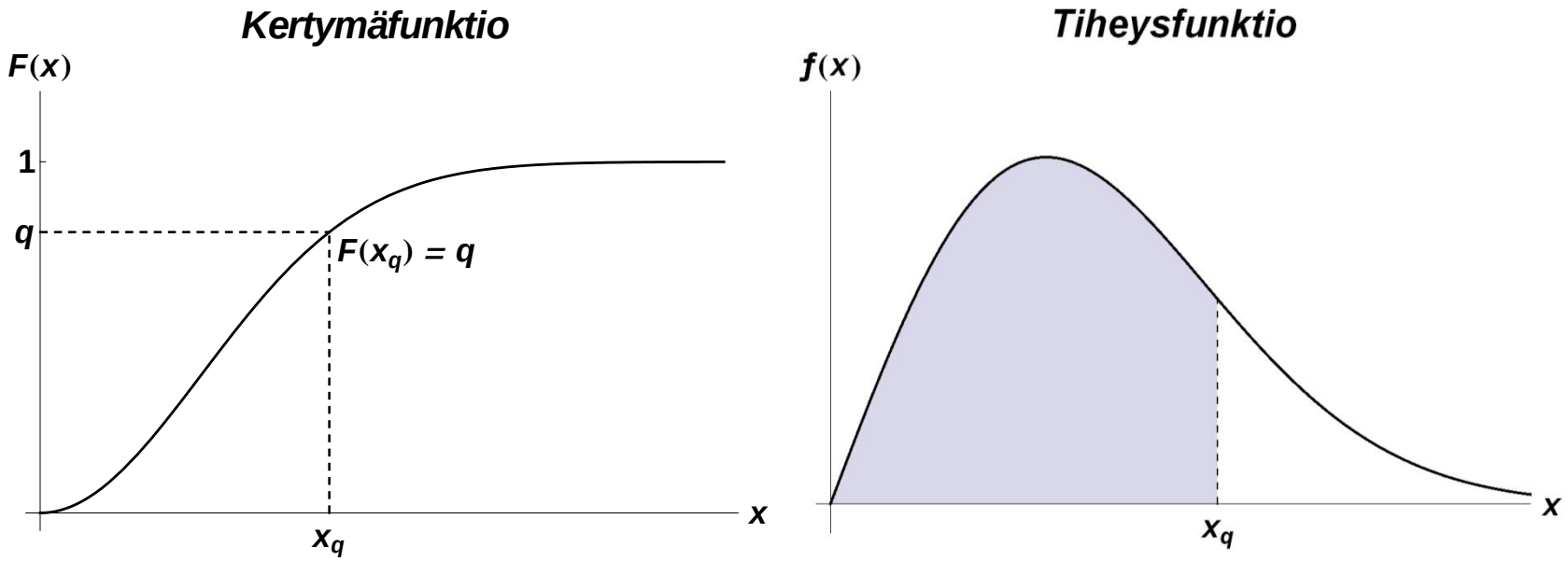
Kannattaako tallentaa kaikki data?

- Monitorointidataa kertyy tasaiseen tahtiin
  - Tiukka SLA vaatii tiheää näytteistystä → dataa kertyy nopeaan tahtiin
  - MEPpien lkm kasvaa → dataa kertyy enemmän
  - Jälkikäteinen (**off-line**) prosessointi on sitä työläämpää mitä enemmän dataa on kertynyt
- Jos ”oleellinen” informaatio on tallessa, niin jälkikäteinen analyysi on edelleen mahdollista ja myös nopeampaa
  - Oleellinen informaatio on jotain jonka tulkittavuus säilyy
    - Mittausprosessista pitäisi tallettaa riittävästi **metadataa**

Mitkä tunnusluvut sitten kannattaa laskea on-line?

- Nopeasti laskettavia (on-line) taikalukuja ei tiettävästi ole
- *Otoskeskiarvo* ja *otoskeskihajonta* (on-line) eivät ole riittävän informatiivisia lukuja osoittamaan onko ne laskettu stationaarisesta aikasarjasta jos aikasarjaa itseään ei tallenneta
  - Tarvittava lisävaatimus olisi *otosautokorrelaatiofunktio* jolle en tiedä olevan riittävän nopeaa (on-line) algoritmiä
- Jakauman kvantiilit (**quantiles**) tarjoavat paremman mahdollisuuden.
  - Taustalla tiukempi vaatimus stationaarisuudelle (**strict sense stationary**) joka oleellisesti vaatii, että jakauman kvantiilit eivät muutu

Kvantiilin (percentiili, fraktiili) käsite



- Todennäköisyyttä *q* vastaava arvo x_q on jakauman *F q*:s kvantiili (fraktiili, percentiili), $q = F(x_q) = P\{ X \leq x_q \}$,
- Esim. Jakauman mediaani $x_{0.5}$ on luku jolle $0.5 = P\{ X \leq x_{0.5} \}$

Kvantiilin (percentiili, fraktiili) käsite

- Otoksesta lasketut kvantiilit, esimerkiksi kvartiilit $x_{0.25}$, $x_{0.5}$ ja $x_{0.75}$ jakavat datan lokeroihin (**bin**) $[-\infty,x_{0.25}]$, $]x_{0.25},x_{0.5}]$, $]x_{0.5},x_{0.75}]$ ja $]x_{0.75},\infty[$ siten, että jokaisessa lokerossa on (noin) 25% havainnoista
  - Yllä: 3 kvantiilia, 3+1 lokeroa
  - Esimerkiksi lokeroon $]x_{0.75},\infty[$ osuvat havainnot ovat siis kaikki suurempia kuin $x_{0.75}$
  - Mitä useampi kvantiili estimoidaan, sitä tarkempi kuva jakaumasta saadaan

Usean kvantiilin yhtäaikainen sarjallinen estimointi

Historiaa:

- Raj Jain, Imrich Chlamtac: *The P^2-Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations*, Communications of the ACM, October 1985, Volume 28, Number 10.
  - ”P^2 ” = ”Piecewise Parabolic”
- Kimmo Raatikainen: *Simultaneous Estimation of Several Persentiles*, SIMULATION 159, October 1987.
- Kimmo Raatikainen: *Sequential Procedure for Simultaneous Estimation of Several Percentiles*, Transactions of the Society for Computer Simulation 7, 1 (March 1990): 21-44.

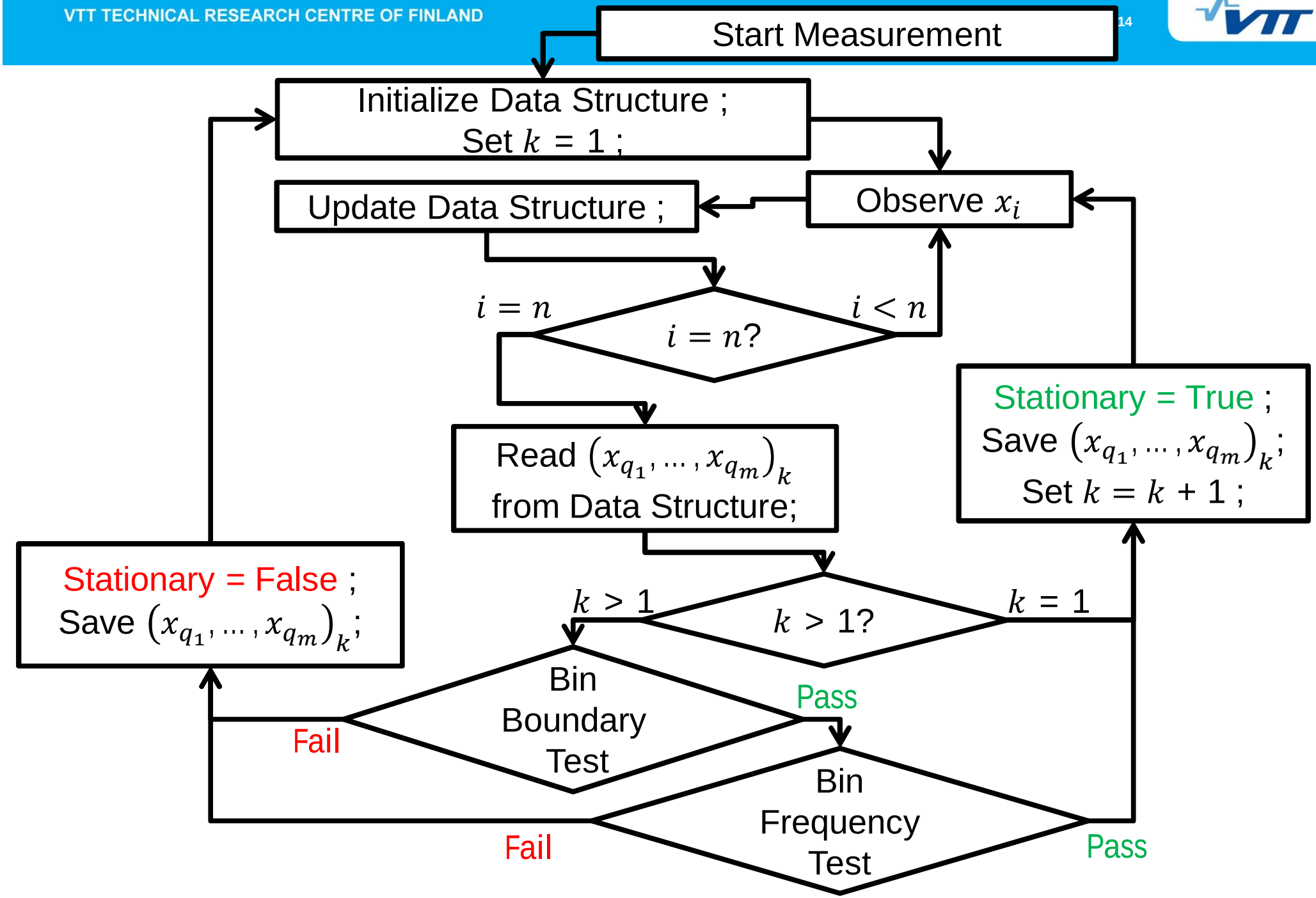
Usean kvantiilin yhtäaikainen sarjallinen estimointi

	$x_{(1)}$		x_{q_1}		x_{q_2}		x_{q_3}		$x_{(n)}$	
q_i	75476.	84609.	163163.	346173.	456085.	502812.	533273.	776620.	864839.	**heights**
n_i	1	2	3	4	5	6	7	8	9	**actual positions**
f_i	0.	0.125	0.25	0.375	0.5	0.625	0.75	0.875	1.	**increments of desired positions**
d_i	1.	2.	3.	4.	5.	6.	7.	8.	9.	**desired positions**

- Laajennettu P^2-algoritmi ylläpitää kiinteän kokoista tietorakennetta
  - Tietorakenteen koko on $(2m+3)\times 4$ lukua, m on estimoitavien kvantiilien lukumäärä.
- Tietorakenteen alustaminen vaatii $2m+3$ erisuurta havaintoa (kuvassa $m = 3$)
- Muiden sarakkeiden tarkoituksena on stabiloida estimointia
- Vektori $(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})$ voidaan lukea ja tallettaa tietorakenteesta milloin vain

P^2-algorimin soveltamisesta

- Algoritmi toimii hyvin kun lähde on stationaarinen
- Algoritmi **ei** toimi hyvin kun lähde ei ole stationaarinen
  - Tarvitaan sellainen monitorointiprosessi, joka jollain lailla kykenee tarkistamaan onko lähde stationaarinen vai ei
  - Pitää siis tuottaa metadataa, joka sisältää informaatiota siitä ollaanko stationaarisessa tilassa vai ei
  - Algoritmin tietorakenne pitää alustaa uudelleen, jos havaitaan ei stationaarinen jakso
- Seuraavalla kalvolla on yksinkertaistettu vuokaavio tällaisesta prosessista
  - Vuokaaviossa esiintyvä luku *k* on metadataa


Start Measurement
Initialize Data Structure ;
Set k = 1 ;
Update Data Structure ;
Observe x_i
$i = n$
$i = n$?
$i < n$
Read $(x_{q_1}, \dots, x_{q_m})_k$
from Data Structure;
Stationary = True ;
Save $(x_{q_1}, \dots, x_{q_m})_k$;
Set $k = k$ + 1 ;
Stationary = False ;
Save $(x_{q_1}, \dots, x_{q_m})_k$;
k > 1
k > 1?
k = 1
Bin
Boundary
Test
Fail
Pass
Bin
Frequency
Test
Fail
Pass

Miltä tallennettu lokidata näyttää?

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_1$

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_2$ // k > 1

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_3$

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_1$ // k = 1 edellinen vektori sisältää ehkä anomalian

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_2$

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_3$

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_4$

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_1$ // k = 1 edellinen vektori sisältää ehkä anomalian

$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_2$

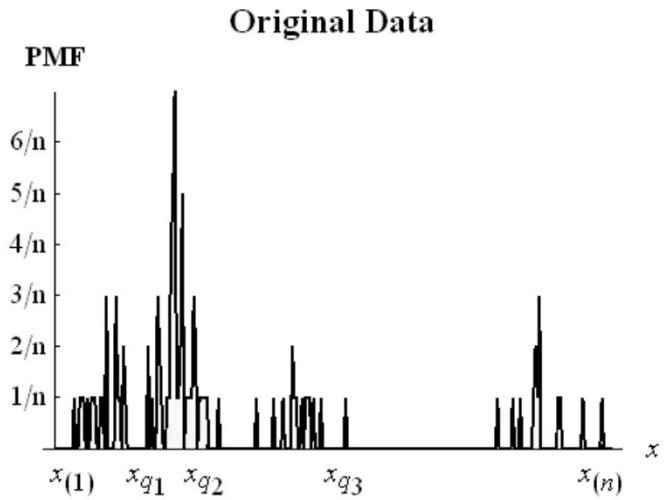
$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_3$

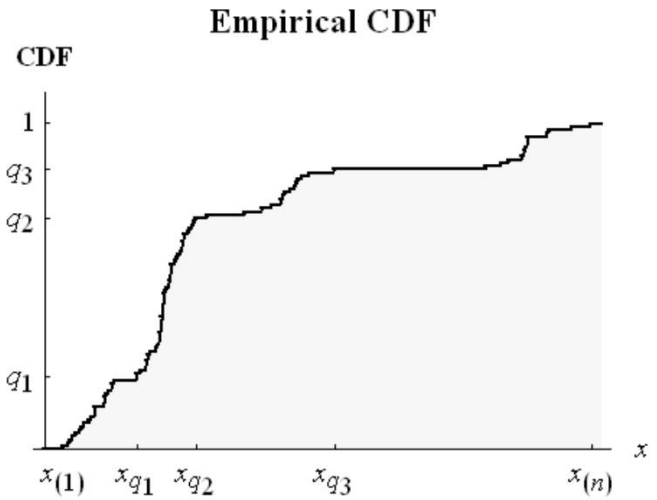
$(x_{(1)},x_{q1},\ldots,x_{qm},x_{(n)})_4$

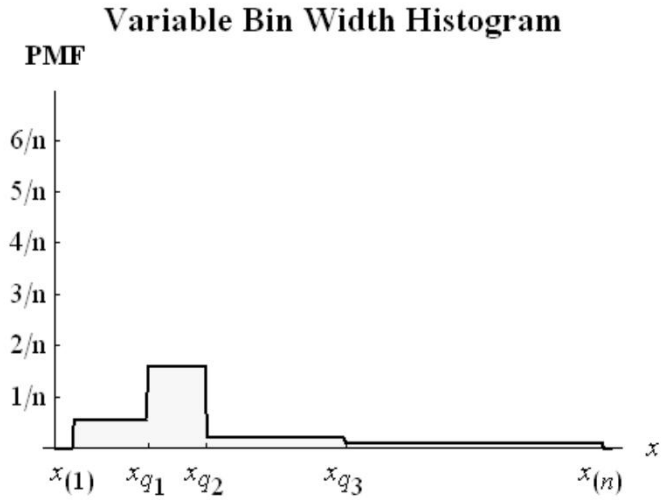
...

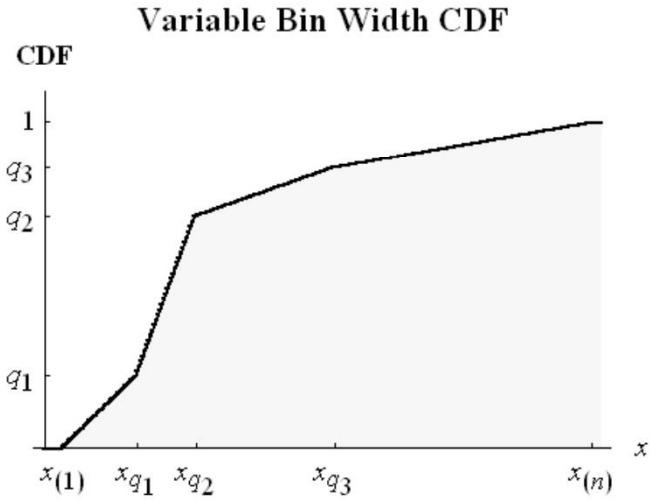
...

Jokainen stationaarinen vektori on histogrammi









Taikatemppu: MDL

- Taikatemppu, jolla edellisen kalvon kuvassa osattiin valita juuri sellaiset todennäköisyydet q_1, q_2 ja q_3, että jakauman muoto tuli näkyviin perustuu ns. **Minimum Description Length (MDL)** –metodiin
  - Informaatioteoriaan perustuva **algoritminen metodi**

Mikä on kompressiosuhde?

- Alkuperäistä dataa havaitaan *n* peräkkäistä havaintoa
- Jokaisesta *n*-blokista tallentuu m + 2 + 1 lukua: $(x_{(1)}, x_{q1}, \ldots, x_{qm}, x_{(n)})_k$
  - Kvantiilien lukumäärä *m* on vapaasti valittavissa
  - Lukumäärän *m* valinta voidaan perustaa myös MDL-periaatteeseen
- Tallennussuhde (%) on teoriassa $100 \times \frac{m+3}{n}\%$
- Käytännössä luvun *n* täytyy voida varioida jonkin verran
  - Tietorakenne vaatii, että $n > 2m + 3$
  - Luku *n* ei voi olla mielivaltaisen suuri
  - Luku *n* voi vaihdella adaptiivisesti
  - MDL-periaate tuo ehtoja lukujen *m* ja *n* välille

Aiheesta enemmän kiinnostuneille

- Kilpi, J., Varjonen, S., *Minimizing Information Loss in Sequential Estimation of Several Quantiles*,
  - Ensimmäinen luonnos (draft) saatavilla GoodNet projektin verkkosivuilta, sisältää perusidean kuvauksen mukaan lukien MDL-periaatteen hyödyntämisen
    - MDL edellyttää oppimisaskeleen ja –datan sekä hieman etukäteislaskentaa
  - Toinen versio on työn alla, tulee sisältämään ainakin vuokaaviossa esitetyn prosessin simulointeja
- Testimahdollisuuksia oikeissa verkoissa toivotaan
  - **Ota yhteyttä!**