

15.06.2005

Testitapausten ideointia...

1. Datan oikein lukeminen: Ohjelman tulostettava rivinumero, sarake, ja luettu arvo, myös tyhjä-arvo-tapauksessa ”luettiin tyhjä arvo, käytetään arvoa x”

Suunnitteluteknistä pohdintaa:

Päärakenne?

1. while (kierrokset (ehto per muuttuja tms.))
 1. while (blokitustaulu)
 1. Päivitysstrategia
 2. while (blokki)
 1. ehdotusstrategia
 2. päivitetään muuttujaa taulukkonumeron ja indeksin perusteella

Generaattori voi kovakoodata päivityskohtaan switchin muuttujan perusteella, jonka caseissa lasketaan P_acc:n arvo.

1. Generaattori?
 1. Inputtiedostojen luku
 2. Kirjoita koodi
2. Datan/Alkuarvojen lukurutiini
3. Päärakenne? (kts. yllä)
4. Jakaumanvalitsijafunktio
5. Tulostusfunktio

Alkuarvot voitaisiin antaa vaikka yhdessä tiedostossa näin:

```
? alpha_i i=1:100
```

```
3.2
```

```
.3
```

```
...
```

```
? beta_j j=1:50
```

```
5.
```

```
32
```

```
1.0
```

```
...
```

```
? x_i_j i=1:100 j=1:50
```

```
1 3.2 5.3 3.4 2.1
```

```
1.23 5.2 3.1 ...
```

```
... ..
```

```
# Jos vaikka puuttuu kamaa obs-datasta paikasta 8,4 ja paikasta 13:34:
```

```
? obs_i_j i=8:8 j=4:4
```

4.32

? obs_i_j i=13:13 j=34:34

5

Ja ehdotusjakaumat samalla formaatilla:

alpha_i i=1:100 Poisson(3)

#ja sitten yksittäisille

alpha_i i=45:45 Poisson(1)

....

x_i_j i=1:100 j=1:50 Norm(3, 0.1) RW

*RW= random walk.

Tämähän on helppoa ja kivaa.

Blokitus: monen muuttujan samanaikainen päivitys: ovatko nämä riippuvia vai riippumattomia?

Tarvitaan mallia. Kovakoodattava blokitus, siis generaattorissa. Höh.

Blokitustiedosto:

(a_1, a_2, a_65 b_42)

(t_1, a_3, f_32) Gibbs

...

Ei tarvita jakaumia, koska jakauma on muuttujan ominaisuus, ei blokin.

Jos ei sanota algoritmia, se on defaulttina M-H.

Monen muuttujan päivitys samassa blokissa näin?

$$P(a' \wedge x') = P(a') \cdot P(x' | a')$$

$$P_{acc} = \frac{P(a') \cdot P(x' | a') \cdot q(x', x) \cdot q(a', a)}{P(a) \cdot P(x|a) \cdot q(x, x') \cdot q(a, a')}$$

Kysyttävä tästä.

