

OHJ-2650
ALGORITMIIKAN
VAIHTUVA-AIHEINEN
KURSSI

2005/2006:

Todennäköisyys ja laskenta

2007/2008:

Koneoppiminen

2008/2009:

Approksimointialgoritmit (?)

vuosittainen kierto ...

1

Hallintoa

- ★ 5 op, syventävät opinnot, ohjelmistotieteen valinnainen kurssi, soveltuu erityisesti jatko-opintoihin
- ★ Luennot ma 14–16, ti 12–14 TC210
5. 12. – 14. 3. (12 viikkoa)
prof. Tapio Elomaa
- ★ Opintopiiri ma 12–14 TC210
 - ◇ Opiskelijoiden keskinäinen harjoitustilaisuus
 - ◇ Luennoija toimittaa mallivastauksen tarvittaessa
- ★ Koe?
- ★ Arvostelu?
- ★ elomaa@cs.tut.fi, TF214

2

Kirjallisuutta

KURSSIKIRJA

- ★ M. Mitzenmacher, E. Upfal: *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*, Cambridge University Press 2005

MUITA LÄHTEITÄ

- ★ R. Motwani, P. Raghavan: *Randomized Algorithms*, Cambridge University Press 1995
- ★ J. Hromkovič: *Algorithmics for Hard Problems: Introduction to Combinatorial Optimization, Randomization, Approximation, and Heuristics*, luku 5: *Randomized Algorithms*, Springer 2001

3

Sisältösuunnitelma

1. Tapahtumat ja todennäköisyys
2. Diskreetit satunnaismuuttujat ja odotusarvo
3. Momentit ja poikkeamat
4. Chernoffin rajat
5. Pallot, urnat ja satunnaisverkot
6. Probabilistinen menetelmä
7. Markovin ketjut ja satunnaiskulut
8. Jatkuvat satunnaismuuttujat ja Poisson-prosessit
9. Entropia, satunnaisuus ja informaatio
10. Monte Carlo -menetelmä
11. Martingaalit

4

Satunnaisuuden hyödyllisyydestä

- ★ **Vastustajan helppo tehtävä:** Yksittäistä determinististä algoritmia vastaan vastustajan on melko helppo konstruoida syöte, jolla algoritmi toimii huonosti.

Satunnaisalgoritmi voidaan nähdä determinististen algoritmien todennäköisyysjakaumana. Vastustajan voi olla helppo tuottaa yhtä (tai pientä osajoukkoa kaikista) determinististä algoritmia hämäävä syöte, mutta satunnaisesti valittua determinististä algoritmia hämäävän yksittäisen syöteen tuottaminen on huomattavasti hankalampaa.

5

- ★ **Todisteiden ylenmääräisyys.** Usein algoritmin on päätettävä onko syöteellä x vaadittu ominaisuus (esim. "onko x alkuluku?"). Ominaisuuden toteamiseksi etsitään todiste.
Deterministisen haun vaikeus voi perustua hakuavaruuden kokoon. Tehokas läpikäyvä haku (exhaustive search) suuressa avaruudessa voi olla laskennallisesti mahdotonta.
Todisteita voi kuitenkin olla paljon, jolloin todennäköisyys, että satunnaisesti valittu alkio on halutun ominaisuuden todiste on suuri. Lisäksi toisistaan riippumattomien suoritusten lisääminen pienentää todennäköisyyttä, ettei yhdelläkään toistoista löydetä todistetta halutulle tapahtumalle.

6

- ★ **Sormenjäljet ja hajautus.** Satunnaiskuvauksella pitkä merkkijono voidaan esittää lyhyenä sormenjälkenä. Joissain hahmontunnistusongelmissa voidaan osoittaa, että kaksi merkkijonoa ovat todennäköisesti identtiset, jos niiden sormenjäljet ovat samat. Luonnollisesti lyhyiden sormenjälkien vertailu on huomattavasti nopeampaa kuin itse merkkijonojen vertailu. Hajautuksen perusajatus on tietysti sama kuin sormenjälkiteknikan.
- ★ **Satunnainen syöteen uudelleenjärjestäminen** johtaa siihen, että annettu syöte on vain hyvin epätodennäköisesti patologinen yksinkertaisen käsittelyalgoritmin syöte.

7

- ★ **Kuorman tasoittaminen** hajauttamalla käytössä olevien resurssien (esim. kommunikaatiokanavien prosessoriverkossa) kesken voidaan aikaansaadat hajauttamalla. Tämä on keskeistä rinnakkais- ja hajautetuissa järjestelmissä.
- ★ **Nopeasti sekoittuvat Markov-ketjut.** Tietyn ominaisuuden omaavien kombinatoristen objektien lukumäärää voi arvioida otannalla huolellisesti määritellystä populaatiosta. Otanta voi vaatia tapausavaruuden koon laskeamista! Joskus otanta voidaan toteuttaa määrittelemällä populaation alkioille Markov-ketju ja osoittamalla, että lyhyt satunnaiskulku todennäköisesti johtaa tasaiseen otantaan.

8

- ★ **Satunnaisotanta** on tapa saada edustava otos koko aineistosta.
- ★ **Probabilistiset menetelmät ja olemassaolotodistukset.** Jotkin ominaisuudet omaavan objektin olemassaolo voidaan varmentaa tarkastelemalla satunnaisesti valittua objektia ja toteamalla, että sillä on kaivatut ominaisuudet positiivisella todennäköisyydellä. Luonnollisesti tämä ei johda menetelmään, jolla haluttu objekti löydettäisiin. Samaa strategiaa voidaan käyttää ongelman ratkaisevan algoritmin olemassaolon todistamiseen.

9

- ★ **Eristyneisyys ja symmetrian hajoitaminen.** Kun rinnakkaislaskennassa ratkaistaan ongelmaa, jolla on monia mahdollisia ratkaisuja, pitää voida varmistaa, että kaikki prosessorit pyrkivät kohti samaa ratkaisua. Täten ratkaisuavaruudesta pitää kyetä eristämään ratkaisu tuntematta yhtäkään avaruuden alkioista. Ratkaisujen implisiittisellä satunnaisjärjestämisellä voidaan eristää järjestyksen mielessä ensimmäinen ratkaisu ja pakottaa kaikki prosessorit pyrkimään kohti tuota ratkaisua. Hajautetussa laskennassa satunnaisuutta voidaan käyttää lukkiutumisen estämiseen.

10