Aalto-yliopisto, Matematiikan ja Systeemianalyysin laitos **Apiola/Kuortti**

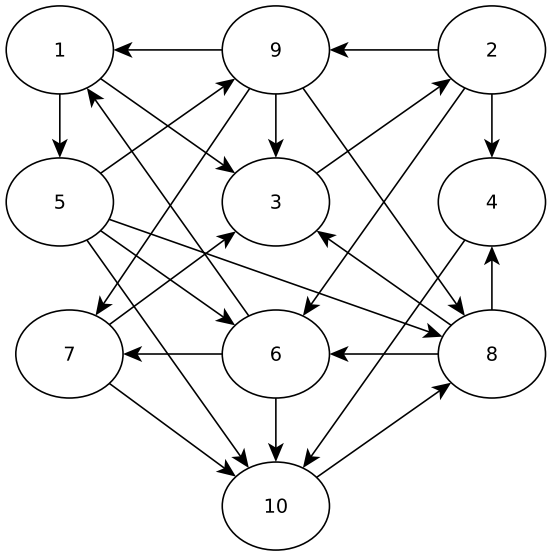
Mat-1.C Matemaattiset ohjelmistot

Projektitehtävät MATLABille 11-14.3.2013

---

1. mlLi090.tex

Seuraava kuva esittää kymmenen sivun ”internettiä”.



Laske tämän verkon tärkein sivu käyttämällä PageRank-algoritmia:

- Luo verkon vierusmatriisi $A = [a_{ij}]$, missä

$$a_{ij} = \begin{cases} 1 & \text{jos sivulta j on linkki sivulle i} \\ 0 & \text{muuten} \end{cases}$$

- Laske vierusmatriisin suurin ominaisarvo, ja vastaava ominaisvektori.
- Normalisoi laskemasi suurinta ominaisarvoa vastaava ominaisvektori (jaa kaikki vektorin alkiot vektorin summalla). Mikä on tämän verkon tärkein sivu?
- Piirrä verkon kuva käyttäen laatimaasi vierusmatriisia ja `gplot`-komentoa. Tutustu `gplot`in help-sivuun.

2. Oletetaan, että meille on annettu dataa muodossa $(x_k, y_k).k = 1 \ldots m$, johon muodustuu kaksi murtopisteen erottamaa lineaarista suuntausta. Esimerkiksi

```
x=-2:0.1:4; y=0.2*sin(3*x);
y(x<1)=y(x<1)+0.5*(x(x<1)-1);
y(x>=1)=y(x>=1)+2*(x(x>=1)-1);
```

muodostaa selvän murtopisteen kohtaan $x = 1$. Intuitiivisesti tuntuu selvältä, että tällaiseen dataan kannattaa sovittaa PNS-suoran sijaan paloittain lineaarinen funktio, ts. ”suora murtopisteellä.”

Kirjoita ohjelma joka tekee tämän: ohjelman tulee valita murtopiste (s, t) tasosta hiiren klikkauksen perusteella (kts. vihje) ja sovittaa paloittain lineaarisen funktion dataan tätä murtopistettä käyttäen, ts. sovittaa suoran

$$y = k_1 x + b_1, x < s$$

pisteisiin $(x_k, y_k), x_k < s$ ja suoran

$$y = k_2 x + b_2, x > s$$

pisteisiin $(x_k, y_k), x_k > s$.

Vihje: Tehtävän keskeinen osa on murtopisteen valinta ja datapisteiden suodatus.

Murtopisteen valintaan kannattaa käyttää `ginput` funktiota, joka valitsee klikatun pisteen kuvasta tyyliin

```
[x y] = ginput(1);
```

Datan suodatukseen kannattaa käyttää MATLABin loogista indeksöintiä: esimerkiksi valitaan kaikki vektorin pisteet, jotka ovat pienempiä kuin 5.

```
a = b(b<5);
```

3.

a) Luo dataa seuraavalla skriptillä:

```
r = 0.5+0.5*rand(10,1);
theta =2*pi*rand(10,1)
x = 3*r.*cos(theta);
y = 3*r.*sin(theta);
```

ja piirrä data pisteittäin.

b) Sovitamme dataan ympyrän muotoa $(x-c_1)^2+(y-c_2)^2 = r^2$. Ympyrän sovituksessa etsitään kahta arvoa: ympyrän keskipistettä (c_1, c_2), ja sen sädettä r. Helpoimmin sovitus onnistuu huomaamalla, että $(x - c_1)^2 + (y - c_2)^2 = r^2 \Leftrightarrow 2xc_1 + 2yc_2 + (r^2 - c_1^2 - c_2^2) = x^2 + y^2$. Asettamalla $c_3 = r^2 - c_1^2 - c_2^2$, saadaan yhtälö muotoa

$$2xc_1 + 2yc_2 + c_3 = x^2 + y^2.$$

Tälle yhtälölle voidaan tehdä vaadittu datan sovitus, ja ratkaista arvot (c_1, c_2, c_3), jonka jälkeen $c_3$sta ratkaistaan r.

Vihje: Pisteittäinen piirtäminen onnistuu komennolla `plot(x,y,'.')`. Ympyrän, jonka keskipiste on (x, y) ja säde r, voi piirtää komennolla `plot(x+r*cos(0:0.02:2*pi,y+r*sin(0:0.02:pi)))`.

4. Ratkaise differentiaaliyhtälöryhmä

$$\begin{cases} \frac{dx}{dt} = -\sigma x + \rho y \\ \frac{dy}{dt} = \sigma x - y - xz \\ \frac{dz}{dt} = -\beta z + xy \end{cases}$$

numeerisesti välillä $[0, 20]$, kun $\sigma = 10, \rho = 28$ ja $\beta = 8/3$. Piirrä ratkaisukäyrät samaan kuvaan, ja piirrä käyrät $x(t)$ ja $z(t)$ parametrisesti. Tämän jälkeen piirrä 3-ulotteinen parametrisoitu käyrä kaikista koordinaateista.

Onko ratkaisu rajoitettu? Suppeneeko se kohti jotain arvoa?

Kokeile muuttaa alkuarvoja, sekä parametrien arvoja. Vallitsevan teorian mukaan systeemi on *kaoottinen dynaaminen systeemi*, jonka käyttäytyminen voi muuttua merkitsevästi jo pienistä muutoksista lähtötilanteessa; itse asiassa termi perhosvaikutus keksittiin kuvaamaan juuri tämän systeemin käytöstä.

Vihje: Kolmiulotteinen parametrisoitu käyrä (tai pistejoukko) piirretään MATLABissa funktiolla `plot3`.

5. Olkoot c ja z_0 kompleksilukuja. Tällöin rekursion

$$z_n = z_{n-1}^2 + c$$

määräämä dynaaminen systeemi tunnetaan kvadraattisena kuvauksena. Valituille luvuille c ja z_0 ylläoleva rekursio johtaa kompleksiseen lukujonoon $z_1, z_2, z_3 \ldots$. Tätä jonoa kutsutaan z_0:n kiertoradaksi. Riippuen lukujen c ja z_0 valinnasta ratojen muotoja on useita.

Annetulle kiinteälle luvulle c useimmilla z_0 rata lähestyy ääretöntä (eli $|z_n|$ kasvaa rajatta kun $n \to \infty$.) Joillakin c ja z_0 rata kuitenkin suppenee kohti jotain periodista silmukkaa (eli arvot kiertävät z_0 jollain tietyllä etäisyydellä $|z_n|$); joillakin alkuarvoilla rata on kaoottinen. Nämä alkuarvot z_0 ovat kuvauksen Julia-joukko.

Tässä harjoituksessa kirjoitetaan MATLAB-ohjelma, joka laskee ns. täytetyn Julia-joukon, joka koostuu niistä alkioista z_0 joiden radat jollain annetulla arvolla c eivät kasva rajatta – tavallinen Julia-joukko on tämän joukon reuna.

On näytetty, että jos $|z_n|$ kasvaa isommaksi kuin 2 jollain arvolla n, rekursio kasvaa rajatta. Arvoa n jolla tämä tapahtuu, kutsutaan tässä tehtävässä pisteen z_0 "pakonopeudeksi."

Aloita kirjoittamalla funktio `n = escapeVelocity(z0,c,N)`, jossa N on jokin yläraja pakonopeuksille (erityisesti: jos $|z_n| < 2 \ \forall \ n < N$, funktion tulee palauttaa N. Näin vältetään ikuiset silmukat).

Luodaksesi Julia-joukon kirjoita funktio `M=julia(zMax,c,N)`. Argumentti `zMax` määrää kompleksitasosta nelikulmion $|Im(z)| < z_{max}, |Re(z)| < z_{maz}$. c ja N ovat samat argumentit kuin edellä, palautettava matriisi $\mathbf{M}$ koostuu määritetyn hilan pakonopeuksista.

Aloita funktion `julia` kirjoittaminen määrittelemällä 500×500 hila realitasossa, luo sen avulla vastaava hila $\mathbf{Z}$ kompleksitasolle, ja aja funktio `escapeVelocity` jokaiselle matriisin $\mathbf{Z}$ alkiolle.

Vihje: Realiakselin väli $[a, b]$ määritellään MATLABissa komennolla `I = linspace(a,b,n)`, missä n on haluttujen pisteiden määrä, kuten esim. 500. Hila reaalitasolle määritellään komennolla `[x y] = meshgrid(t1,t2)` , missä t1 ja t2 ovat välejä reaaliakselilta. Tästä luodaan kompleksitasoa peittävä hila komennolla `z = x+i*y`.

Kompleksiluvun modulin saa selville itseisarvofunktiolla `abs`.

6. mlT005.tex
(Osa kaavoista epäselviä html:ssä, katso pdf-tehtäviä!)
Monte Carlo-approksimaatio π:lle.

Piirrä kuva

```
t=linspace(0,2*pi);
x=cos(t);y=sin(t);
plot(x,y,[1 1 -1 -1 1],[-1 1 1 -1 -1]);
axis([-1.5 1.5 -1.5 1.5])
axis square
```

Heitetään tikkaa kuvan mukaiseen tauluun (tikat eivät eksy taulua ympäröivään neliön ulkopuolelle, ehkä tähän oikeasti tarvitaan "satunnaisrobotti"). Jos tikkojen osumatarkuus on satunnaismuuttuja, joka on tasajakautunut neliöllä
$-1 < x <, 1 < y < 1$, niin ympyrään ja neliöön osuneiden tikkojen lukumäärän suhde lähenee lukua $\pi/4$, kun satunnaisheittojen lukumäärä kasvaa. Miksi? Generoi tasajakautuneita pistepareja ja laske ko. osuus.

Alla on vihjettä pikku esitystä varten. Toisaalta tehtävä ei kaipaa mitään lisäopiskelua, tai vihjeitä. Kenties ehto `X.^2+Y.^2 <= 1` ja bittivektorin ykkösten lukumäärän laskeminen vähemmän Matlabia osaaville. (Vrt. edellinen kolmiotehtävä mlT004.tex.)

Vihje: Kirjassa C.vanL on hyvä tiivis selvitys aiheesta "Random processes"1.3.2 ss. 34 - 37. Tästä aiheesta voisi tehdä pienen harjoitustyön.

Esiintyviä Matlab-funktioita:

`http://www.mathworks.com/access/helpdesk/help/techdoc/ref/hist.shtml`
`http://www.mathworks.com/access/helpdesk/help/techdoc/ref/rand.shtml`
`http://www.mathworks.com/access/helpdesk/help/techdoc/ref/randn.shtml.`

Satunnaisprosesseihin ja tähän tehtävään (Monte Carlo simulaatio π:n laskemiseksi) on CV-sivulla selkeät skriptit:

`http://www.cs.cornell.edu/cv/Books/SCMV/Mfiles/chap1.htm#Dice`

Dice ja Darts ala C. van Loan. Opiskele, kokeile ja esittele.

Avainsanat: mlTodari, mlPerusteet, matlabperusteet, Monte Carlo, looginen ineksinti, satunnaisluvut

7. Tässä tehtävässä tutkitaan kuvien, matriisien ja singulaariarvojen yhteyksiä.

Matriisin $\mathbf{A} \in \mathbb{R}^{m \times n}$ singulaariarvohajotelma on

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T,$$

missä matriisi **S** on diagonaalimatriisi, ja matriisit **U** ja **V** ovat ortogonaalisia neliömatriiseja. Matriisin sisältämää informaatiota voidaan tietyssä mielessä kompressoida tiputtamalla osia singulaariarvohajotelmasta pois; on todistettavissa että (MATLABilla ilmaistuna) `U(:,1:k)*S(1:k,1:k)*V(:,1:k)'` on paras mahdollinen rank(k)-approksimaatio matriisille **A**.

Kuva voidaan ajatella $m \times n$ matriisina, missä i, j alkio ilmaisee vastaavassa paikassa olevan pikselin väriarvon. Tutkitaan sitten kuinka singulaariarvoja voidaan käyttää hyväksi kuvien pakkaamisessa ja hahmontunnistuksessa.

Lue haluamasi kuva sisään MATLABin `imread` komennolla. Komento luo (yleensä, mutta hieman kuvasta riippuen), $m \times n \times 3$ matriisin. Tämä vastaa RGB-esitystä: ensimmäisessä kerroksessa on punaisen värin intensiteetit, toisessa vihreän ja kolmannessa sinisen. Muuta tämä matriisi harmaaskaalaan komennolla `rgb2gray`. Tämän jälkeen tee matriisille singulaariarvohajotelma komennolla `[u s v] = svd(P)`, missä `P` on kuvasi matriisiesitys. Tutki sitten millä k:n arvolla komentojono

```
>> M = u(:,1:k)*s(1:k,1:k)*v(:,1:k)';
>> image(M)
```

tuottaa havaittavia tuloksia. Pitäisi myös päteä, että kuvan isommat hahmot alkavat erottua ensin, mikä tekee singulaariarvoista huomattavan tehokkaan työkalun hahmontunnistuksessa.

Vihje: Kuvan ulottuvuuksien ei kannata olla kovin isoja: singulaariarvohajotelma on raskas laskettava. Jos haluat lisähaastetta, erottele kuvan värikerrokset, tee hajotelma niille erikseen, ja kokoa tulokset. Näin saat aikaan värikuvia.