Numerieke Modellering en Benadering
Practicum 1

academiejaar 2018-2019

1 Bivariate veeltermbenadering

In deze opgave benaderen we een oppervlak waarvan de functiewaarden gegeven zijn op een regelmatig rechthoekig puntenrooster. Met andere woorden, we kennen een aantal functiewaarden f_{ij} overeenkomstig met de roosterpunten $(x_i, y_j), i = 1, \ldots, M$ en $j = 1, \ldots, N$. We zullen deze data op twee manieren voorstellen: de kolomvector $\mathbf{f} \in \mathbb{R}^{MN}$ is gegeven door

$$\mathbf{f} = \begin{bmatrix} f_{11} & \cdots & f_{1N} & f_{21} & \cdots & f_{2N} & \cdots & f_{M1} & \cdots & f_{MN} \end{bmatrix}^\top$$

waarbij $(\cdot)^\top$ de transpose voorstelt en de matrix $F \in \mathbb{R}^{N \times M}$ is

$$F = \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{M1} \\ f_{12} & f_{22} & \cdots & f_{M2} \\ \vdots & \vdots & & \vdots \\ f_{1N} & f_{2N} & \cdots & f_{MN} \end{bmatrix}.$$

De relatie tussen deze twee voorstellingen wordt gegeven door de operatie vec : $\mathbb{R}^{N \times M} \to \mathbb{R}^{MN}$, gedefiniëerd door $\text{vec}(F) = \mathbf{f}$. Als benaderende functie gebruiken we een bivariate veelterm $z(x, y)$ gegeven door

$$z(x, y) = \sum_{k=0}^{m} \sum_{l=0}^{n} c_{kl} x^k y^l. \tag{1}$$

We gebruiken voor de eenvoudigheid dus de standaardbasis

$$\{\phi_{kl}(x, y) : 0 \leq k \leq m, 0 \leq l \leq n\}$$

met $\phi_{kl} = p_k(x) q_l(y)$ waarbij $p_k(x) = x^k$ en $q_l(y) = y^l$. De parameters m en n geven de maximale toegelaten graad in x en y respectievelijk. We bepalen de coefficienten $c_{kl}, k = 0, \ldots, m$ en $l = 0, \ldots, n$ zodat de benadering (1) de uitdrukking

$$\sum_{i=1}^{M} \sum_{j=1}^{N} (f_{ij} - z(x_i, y_j))^2 \tag{2}$$

minimaliseert.

Voor deze opgave zullen we gebruik maken van het Kronecker product. Dit wordt genoteerd als de binaire operatie '$\otimes$' en wordt gedefiniëerd als volgt. Het Kronecker product van twee matrices $A \in \mathbb{R}^{s \times t}$ en $B \in \mathbb{R}^{u \times v}$ is de blokmatrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1t}B \\ \vdots & \ddots & \vdots \\ a_{s1}B & \cdots & a_{st}B \end{bmatrix} \in \mathbb{R}^{su \times tv}$$

waarbij a_{ij} het element van A op de i-de rij en de j-de kolom voorstelt. De eigenschappen van het Kronecker product die belangrijk zijn voor deze opgave zijn

$$\begin{aligned}(A \otimes B)^+ &= A^+ \otimes B^+ \\ (A \otimes B)\text{vec}(F) &= \text{vec}(BFA^\top)\end{aligned}$$

waarbij $(\cdot)^+$ de Moore-Penrose pseudo-inverse voorstelt. Andere eigenschappen van het Kroneckerproduct vind je bijvoorbeeld op de wikipedia pagina.

Het probleem kan geherformuleerd worden als volgt: vind de vector $\mathbf{c} \in \mathbb{R}^{(m+1)\times(n+1)}$ zodat

$$\mathbf{c} = \arg\min \|\Gamma\mathbf{c} - \mathbf{f}\|_2^2$$

waarbij $\Gamma = A \otimes B$ met

$$A = \begin{bmatrix} p_0(x_1) & p_1(x_1) & \cdots & p_m(x_1) \\ p_0(x_2) & p_1(x_2) & \cdots & p_m(x_2) \\ \vdots & \vdots & & \vdots \\ p_0(x_M) & p_1(x_M) & \cdots & p_m(x_M) \end{bmatrix}, B = \begin{bmatrix} q_0(y_1) & q_1(y_1) & \cdots & q_n(y_1) \\ q_0(y_2) & q_1(y_2) & \cdots & q_n(y_2) \\ \vdots & \vdots & & \vdots \\ q_0(y_N) & q_1(y_N) & \cdots & q_n(y_N) \end{bmatrix}.$$

Gebruik makend van de eigenschappen van het Kronecker product vinden we dat $\mathbf{c} = \text{vec}(B^+F(A^+)^\top)$.

1.1 Opgave

1. Schrijf een functie `kkb` in Matlab die, gegeven de vectoren $\mathtt{x} = \begin{bmatrix} x_1 & \ldots & x_M \end{bmatrix}$, $\mathtt{y} = \begin{bmatrix} y_1 & \ldots & y_N \end{bmatrix}$, een matrix F met functiewaarden en de parameters m, n de matrix $C = B^+F(A^+)^\top$ berekent. Geef aan welke correctheidstests je uitvoert. Hint: de backslash operator in Matlab berekent impliciet de pseudo-inverse: $P = Q^+R$ kan uitgevoerd worden door `P = Q\R`. Voor het Kronecker product gebruik je het commando `kron`. Met behulp van deze commando's hoeft de code niet langer te zijn dan enkele lijnen.

2. We benaderen twee verschillende datasets op eenzelfde rooster. Gebruik $M = N = 31$ en het commando `linspace` om de vectoren `x` en `y` met equidistante x- en y-waarden te genereren op het interval $[-1, 1]$. Het commando `[X,Y] = meshgrid(x,y)` geeft een set equidistante roosterpunten op het vierkant $\{(x, y) \in \mathbb{R}^2 : -1 \leq x \leq 1, -1 \leq y \leq 1\}$. De twee datasets zijn:

   (a) f_{ij} gegeven door $f_{ij} = \sin((2x_i - 1)^2 + 2y_j)$. Hint: bekijk de documentatie van `meshgrid` om de matrix F te genereren met slechts één commando.

   (b) datapunten op de 'Matlabfunctie'[1]. Gebruik daarvoor het Matlab commando `membrane`: $F =$ `membrane(1,15)`.

   Gebruik de commando's `scatter3` en `surf` om de datapunten te plotten tesamen met het benaderende oppervlak op dezelfde figuur. Voeg per dataset zo'n figuur toe aan het verslag. Gebruik daarvoor $m = n = 7$. Hint: het evalueren van het benaderende oppervlak in de roosterpunten kan in een paar lijnen code gebeuren met het de gegeven functie `polyval2` op Toledo.

3. Plot het verloop van de kostfunctie (2) voor $m = n = 1, \ldots, 20$ voor de twee datasets op dezelfde figuur. Voor welk van de oppervlakken zijn veeltermoppervlakken een geschikte benadering? Hoe zou je het andere oppervlak beter kunnen benaderen?

4. De ASTER Global Digital Elevation Map is een product van METI en NASA en brengt het zeeniveau van de hele wereld in kaart onder de vorm van discrete hoogtegegevens op een regulier

[1] De functie op het logo van Matlab is een eigenfunctie van de golfvergelijking op een L-vormig domein.

rooster. Het bestand `etna.jpg` dat je op Toledo kan vinden[2] bevat een deel van deze map, namelijk een rechthoekig domein rond de vulkaan Etna op Sicilië. Elke pixel van de afbeelding bevat een hoogtewaarde. Je kan de data inlezen in matlab als een matrix F via het commando `F = double(imread('etna.jpg'));`. Gebruik de code uit de vorige opgaven om een veeltermbenadering te maken van de Etna. Normalizeren we het grid dan kunnen we opnieuw werken met

```
x = linspace(-1,1,M);
y = linspace(-1,1,N);
```

waarbij `M` en `N` de gepaste waarden hebben (overeenkomstig met de datamatrix). Na opnieuw het commando `[X,Y] = meshgrid(x,y)` uit te voeren, kan je de data op een mooie manier weergeven met

```
surf(X,Y,F,'EdgeColor','none','LineStyle','none','FaceLighting','phong')
xlim([-1,1]); ylim([-1,1]); zlim([0, 250]);
title('Etna')
```

en met `imagesc(F)` (dit geeft een bovenaanzicht). Doe hetzelfde voor het benaderende veeltermoppervlak met $m = n = 25$ en voeg de resultaten toe aan het verslag.

2 Interpolerende periodieke splinefuncties en -curven

Voor het benaderen van continue periodieke functies of gesloten krommen zijn klassieke veeltermen niet geschikt. We zullen daarvoor een andere functievoorstelling moeten aanwenden. In dit deel van het practicum maken we gebruik van periodieke interpolerende splinefuncties.

2.1 Het opstellen van periodieke interpolerende kubische splinefuncties

De theorie hierover vind je terug in paragraaf 5.2.3 van de cursus. Merk op dat een periodieke interpolerende kubische splinefunctie volledig gedefinieerd is door middel van $n+1$ abscissen $\{x_k\}_{k=0}^n$ en n functiewaarden $f_k = f(x_k)$ voor $k = 0, \ldots, n-1$. De functiewaarde f_n in het punt x_n moet gelijk zijn aan f_0 omwille van de periodiciteit.

2.2 Opgave A

1. Schrijf een Matlab-functie voor het opstellen en evalueren van periodieke interpolerende kubische splinefuncties, gebruik makende van de theorie uit paragraaf 5.2.3 van de cursus.

   `function y = periospline(x,f,t)`

   Deze functie krijgt als invoer een vector `x` van lengte $n+1$ die de abscissen bevat, een $(d \times n)$-matrix `f` van functiewaarden (zodat d verschillende functies tegelijk kunnen benaderd worden) en een vector `t` van lengte N die de punten bevat waarin we de verschillende splinefuncties willen evalueren. Als uitvoer geeft de functie de $(d \times N)$-matrix `y` terug met in elke rij de functiewaarden in de punten `t` voor de overeenkomstige rij van `f`.

2. Gebruik vervolgens je code om de periodieke functie $f(x) = \sin(x) + \sin(4x)/2$ te benaderen over één periode, vb. $[0, 2\pi]$. Teken een aantal foutencurves voor een verschillend aantal abscissen. Geef ook een grafiek die de maximale fout weergeeft in functie van het aantal abscissen.

3. Gebruik daarna de code om een gesloten kromme te benaderen. Je kan hiervoor gebruik maken van de functie `click`: deze functie geeft de coördinaten van een willekeurig stel punten die je met de muis kan aanklikken.

4. Teken de (min of meer herkenbare) omtrek van een stripfiguur uit een Vlaamse stripreeks. Bijvoorbeeld die van tante Sidonia. De routine `click` kan hierbij een hulp zijn.

[2] De data is publiek verkrijgbaar (op voorwaarde van de aanmaak van een gratis account) via de website `https://gdex.cr.usgs.gov/gdex/`.

3 Praktische schikkingen

- Het practicum wordt gemaakt in groepjes van twee.
- Het verslag kan je afgeven op mijn bureel of doorsturen via mail. De deadline is **12 april 2019, 23u59**.
- Stuur ook een zip-file door naar onderstaand e-mailadres met de code (de m-files) van de gevraagde functies (`kkb.m`, `periospline.m`).
- Het is voldoende dat slechts één groepslid het verslag en de code doorstuurt.
- Bespreek je resultaten bondig en illustreer met verduidelijkende figuren. Probeer de lengte rond de 10 pagina's te houden.

Succes!

Simon Telen (200A 01.152)
`simon.telen@cs.kuleuven.be`