Teoria dell'Approssimazione e Applicazioni, A.A. 2012/13

Esercitazione di laboratorio del 26 marzo 2013

Prof. Stefano De Marchi

1 Scattered data fitting

Il teorema di Haar (o Maierhuber-Curtis), ci ha detto che non è possibile interpolare dati (sparsi) in dimensione spaziale $s \geq 2$ (ovvero di valori posti in posizioni arbitrarie di $\mathbb{R}^s$) con polinomi multivariati di grado $N \geq 2$. Questo impone di scegliere funzioni di base che devono dipendere dai *data-sites* .

Gli strumenti di cui avremo bisogno nel corso della presente esercitazione sono i seguenti.

- I *punti di Halton* che abbiamo visto essere sequenze a bassa discrepanza di punti nell'ipercubo $[0,1]^s, \;\; s \geq 1$.
- La `fill-distance` (o *mesh size*) $h_{X,\Omega}$ di un insieme di punti $X \subset \Omega$ con $\Omega \subseteq \mathbb{R}^s$ è definita come
$$h_{X,\Omega} = \sup_{x \in \Omega} \min_{\mathbf{x}_j \in X} \|x - x_j\|_2 \,, \tag{1}$$
In Matlab, la possiamo determinare con il comando `hX=max(min(DME'))`, dove `DME` è la matrice delle distanze, costruita mediante la funzione `DistanceMatrix.m`, valutata in un insieme di punti di valutazione scelti (ad es. una griglia equispiaziata molto più fine di quella dei data-sites `X`).
- Le funzioni da approssimare che considereremo sono
$$f_s(\mathbf{x}) = 4^s \prod_{k=1}^{s} x_k(1-x_k), \quad \mathbf{x} = (x_1, \ldots, x_s) \in [0,1]^s \tag{2}$$
$$\mathtt{sinc}(\mathbf{x}) = \prod_{k=1}^{s} \frac{\sin(\pi x_k)}{\pi x_k}\,. \tag{3}$$

2 Esercitazione proposta

1. Usare la funzione `haltonseq.m` per costruire sequenze di punti di Halton per dimensioni spaziali $s = 1, 2, 3$. Calcolare, per ognuno di tali insiemi, la corrispondente `fill-distance`, $h_{X,\Omega}$.
2. Verificare graficamente la seguente proprietà di *annidamento* dei punti di Halton,
$$H_{s,M} \subset H_{s,N}, \;\; M < N \,. \tag{4}$$
Si chiede semplicemente di plottare, con colori diversi, i punti di Halton per valori diversi di M e N.

3. Usare la funzione `DistanceMatrix.m` su vari insiemi di punti di Halton con dimensione $s = 2$, verificando che la corrispondente *distance-matrix* risulta malcondizionata, calcolandone il numero di condizionamento usando il comando Matlab `cond`.

4. Sempre per $s = 2$, usando la funzione `DistanceMatrixFit.m`, si costruisca l'interpolante RBF con funzioni di base $B_k(x) = \|x - x_k\|_2$ (ovvero le traslate nel punto x_k della funzione $B(x) = |x|$), delle funzioni (2) e (3) calcolandone anche l'errore RMSE. Verificare che al crescere del numero di punti l' RMSE diminuisce. Anche in questo caso il RMSE dovrà essere calcolato su una griglia più fine di punti di valutazione.

5. Ripetere l'esercizio precedente usando la funzione radiale gaussiana $\Phi(\mathbf{x}) = \mathrm{e}^{-\epsilon^2\|x\|^2}$, $\epsilon > 0$ sempre per $s = 2$. Per questo esercizio si usi la funzione `RBFInterpolation2D.m` che generalizza la `DistanceMatrixFit.m`.

Ricordo che il link dove trovare i files delle funzioni Matlab citate è:
`http://www.math.unipd.it/~demarchi/TAA2010`