

Ausgabe: 28.11.2019

Übung 8

Abgabe: 05.12.2019, 12:00 Uhr

Aufgabe 17 (16 Punkte)

Mit Hilfe der Multiskalenanalyse lässt sich jede Funktion $v \in \text{span}\{\phi_{J,k} : k \in \Delta_J\}$ in Einzelskalendarstellung

$$v = \mathbf{c}_J^T \Phi_J = \sum_{k \in \Delta_J} c_{J,k} \phi_{J,k}$$

oder in Multiskalendarstellung

$$v = \mathbf{c}_{j_0}^T \Phi_{j_0} + \mathbf{d}_{j_0}^T \Psi_{j_0} + \dots + \mathbf{d}_{J-1}^T \Psi_{J-1}$$

bezüglich der Waveletbasis

$$\Psi^J := \Phi_{j_0} \cup \bigcup_{j=j_0}^{J-1} \Psi_j =: \bigcup_{j=j_0-1}^{J-1} \Psi_j$$

darstellen.

Die Wavelettransformation T_J bildet die Detailkoeffizienten $\mathbf{d}^J := (\mathbf{c}_{j_0}, \mathbf{d}_{j_0}, \dots, \mathbf{d}_{J-1})^T$ auf die Koeffizienten \mathbf{c}_J der Einzelskalendarstellung ab. Die Detailkoeffizienten enthalten Informationen zwischen den Level, und die (betragsmäßige) Größe der Koeffizienten ist relevant für die Exaktheit der Darstellung. Dadurch lassen sich Daten mit Hilfe der so genannten *besten N-Term Approximation* komprimieren. Bei der besten N -Term Approximation werden die betragsmäßig kleinsten Koeffizienten gelöscht, d.h. auf 0 gesetzt.

Die Wavelet-Transformation ergibt sich durch mehrfaches Anwenden der Matrix $M_j := (M_{j,0}, M_{j,1})$, welche einen Basiswechsel

$$\begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = \begin{pmatrix} M_{j,0}^T \\ M_{j,1}^T \end{pmatrix} \Phi_{j+1} = M_j^T \Phi_{j+1}$$

zwischen

$$\Phi_j := \{\phi_{j+1,k} : k \in \Delta_{j+1}\} \quad \text{und} \quad \{\Phi_j \cup \Psi_j\} := \{\phi_{j,k} : k \in \Delta_j\} \cup \{\psi_{j,k} : k \in \nabla_j\}$$

beschreibt. Die Wavelet-Transformation kann nun als lineare Abbildung

$$(\Phi_{j_0}, \Psi_{j_0}, \Psi_{j_0+1}, \dots, \Psi_{J-1})^T = T_J^T \Phi_J$$

geschrieben werden, wobei die Transformationsmatrix weiter dargestellt werden kann als

$$T_J := T_{J,J-1} \cdots T_{J,j_0} : \ell_2(\Delta_J) \rightarrow \ell_2(\Delta_J) \quad \text{mit} \quad T_{J,j} := \begin{pmatrix} M_j & 0 \\ 0 & I_{(\#\Delta_J) - (\#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J) \times (\#\Delta_J)}.$$

Umgekehrt gilt die Rekonstruktionsidentität

$$\Phi_{j+1} = G_j^T \begin{pmatrix} \Phi_j \\ \Psi_j \end{pmatrix} = G_{j,0}^T \Phi_j + G_{j,1}^T \Psi_j,$$

mit einer entsprechenden Matrix G_j . Analog ergibt sich die inverse Wavelet-Transformation

$$\Phi_J = (T_J^T)^{-1} (\Phi_{j_0}, \Psi_{j_0}, \Psi_{j_0+1}, \dots, \Psi_{J-1})^T,$$

wobei die Transformationsmatrix weiter dargestellt werden kann als

$$T_J^{-1} := T_{J,j_0}^{-1} \cdots T_{J,J-1}^{-1} : \ell_2(\Delta_J) \rightarrow \ell_2(\Delta_J) \quad \text{mit} \quad T_{J,j}^{-1} := \begin{pmatrix} G_j & 0 \\ 0 & I_{(\#\Delta_J) - (\#\Delta_{j+1})} \end{pmatrix} \in \mathbb{R}^{(\#\Delta_J) \times (\#\Delta_J)}.$$

In dieser Aufgabe sollen hierarchische Basen bzgl. der charakteristischen Funktion (kardinaler B-Spline mit $k = 1$, $\#\Delta_j = 2^j$) betrachtet werden. Die zugehörigen Matrizen sind gegeben durch

$$M_{j,0} = 2^{-1/2} \begin{pmatrix} 1 & 0 & & & \\ 1 & 0 & & & \\ 0 & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & 1 \end{pmatrix}, \quad M_{j,1} = 2^{-1/2} \begin{pmatrix} 1 & 0 & & & \\ -1 & 0 & & & \\ 0 & 1 & & & \\ & & \ddots & & \\ & & & 1 & \\ & & & & -1 \end{pmatrix},$$

$$G_{j,0} = 2^{-1/2} \begin{pmatrix} 1 & 1 & & & \\ 0 & 0 & 1 & 1 & \\ & & & \ddots & \\ & & & & 1 & 1 \end{pmatrix}, \quad G_{j,1} = 2^{-1/2} \begin{pmatrix} 1 & -1 & & & \\ 0 & 0 & 1 & -1 & \\ & & & \ddots & \\ & & & & 1 & -1 \end{pmatrix}.$$

- a) Schreiben Sie eine Funktion $\mathbf{w} = \text{FWT}(\mathbf{x}, j_{\text{Max}})$ und eine Funktion $\mathbf{x} = \text{invFWT}(\mathbf{w}, j_{\text{Max}})$, welche die Fast-Wavelet-Transformation und ihr Inverses durchführen. Der Vektor \mathbf{x} mit Koordinaten bezüglich der Einzelskalenbasis auf Level $j = j_{\text{Max}}$ wird dabei in einen Vektor \mathbf{w} mit Wavelet-Koordinaten transformiert und umgekehrt.

Wählen Sie dazu jeweils $j_0 = 2$ und führen Sie die Transformation mittels des Pyramid-Schemas durch (stellen Sie die Matrizen T_j und T_j^{-1} nicht explizit auf).

- b) Schreiben Sie eine Funktion $\mathbf{dc} = \text{largestNTerm}(\mathbf{d}, \mathbf{N}, j_{\text{Max}})$, die zu gegebenem Vektor von Detailkoeffizienten $\mathbf{d}^{j_{\text{Max}}} = \mathbf{d}$ die \mathbf{N} betragsmäßig größten Koeffizienten von $\mathbf{d}^{j_{\text{Max}}}$ erhält und die restlichen Koeffizienten auf 0 setzt. Dies entspricht einer Variante der besten N -Term Approximation, bei der die Anzahl der Koeffizienten gesteuert wird.
- c) Schreiben Sie eine Funktion $\mathbf{dc} = \text{reLErrorNTerm}(\mathbf{d}, \mathbf{r}, j_{\text{Max}})$, die zu gegebenem Vektor von Detailkoeffizienten $\mathbf{d}^{j_{\text{Max}}} = \mathbf{d}$ einen Vektor \mathbf{dc} zurückliefert, der die Bedingung

$$\|\mathbf{d}^{j_{\text{Max}}} - \mathbf{dc}\|_{\ell_2(\Delta_{j_{\text{Max}}})} \leq \mathbf{r} \|\mathbf{d}^{j_{\text{Max}}}\|_{\ell_2(\Delta_{j_{\text{Max}}})}$$

erfüllt und minimale Anzahl an Nicht-Null-Einträgen hat. Es sollen also so viele Einträge wie möglich gelöscht werden, sodass der relative Fehler in der $\ell_2(\Delta_{j_{\text{Max}}})$ -Norm kleiner als eine vorgegebene Schranke $\mathbf{r} \in (0, 1)$ ist. Dies entspricht einer Variante der besten N -Term Approximation, bei der der relative Fehler gesteuert wird.

- d) Ziel ist es nun, mit Hilfe der beiden Varianten der besten N -Term Approximation eine Datenkompression anhand einer vorgegebenen Datenmenge durchzuführen. Als Datenmenge verwenden Sie die bereinigten Schlusskurse des Dax der letzten $2^{11} = 2048$ Handelstage zwischen dem 2.5.2005 und dem 1.5.2013. Diese Daten finden Sie in dem Matlab-File `Dax.mat` auf der Homepage der Vorlesung. Gehen Sie dazu wie folgt vor:

- i) Bestimmen Sie die Entwicklungskoeffizienten \mathbf{c}_J der Einzelskalenbasis, sodass jeder Datenpunkt exakt interpoliert wird. Beachten Sie dabei die Skalierung der Basisfunktionen $\phi_{j,k}(x) := 2^{j/2} N_1(2^j x - k)$.
- ii) Transformieren Sie die Koeffizienten \mathbf{c}_J in die Detailkoeffizienten \mathbf{d}^J .
- iii) Wenden Sie die beste N -Term Approximation auf \mathbf{d}^J an.
- iv) Transformieren Sie zurück in Einzelskalendarstellung, um die Approximation plotten zu können.

Führen Sie diese Datenkompression bezüglich der besten N -Term Approximation aus Teil b) mit den Parametern $\mathbf{N} = 150, 500, 1000$ aus sowie bezüglich der Variante aus Teil c) mit den Parametern $\mathbf{r} = 0.005, 0.01, 0.05$. Plotten Sie für beide Varianten je ihre originale Approximation ohne Kompression sowie die drei komprimierten Approximationen in vier „subplots“. Bestimmen Sie zudem die Kompressionsrate

$$1 - \frac{\#\text{supp}(\mathbf{dc})}{\#\text{supp}(\mathbf{d}^J)}$$

in Prozent und geben Sie diese bei jedem Plot mit an. Hierbei bezeichnet $\#\text{supp}(\mathbf{d})$ die Anzahl der Nicht-Null-Einträge des Vektors \mathbf{d} .

Hinweis: Falls Ihre Implementierung eines vorherigen Aufgabenteils nicht funktioniert, können Sie die entsprechende (geschützte) Implementierung von der Homepage der Vorlesung verwenden.

Aufgabe 18 (4 Punkte)

Zeigen Sie das *Lemma von Stechkin*:

Sei $0 < p \leq q < \infty$ und $\mathbf{a} = (a_n)_{n \in \mathbb{N}} \in \ell_p(\mathbb{N})$. Definiere für $N \in \mathbb{N}$ die Indexmenge \mathcal{I}_N mit $\#\mathcal{I}_N = N$ derart, dass $|a_n| \geq |a_m|$ für alle $n \in \mathcal{I}_N$ und $m \in \mathbb{N} \setminus \mathcal{I}_N$. Das heißt, \mathcal{I}_N enthält die N Indizes der betragsmäßig größten Folgenglieder von \mathbf{a} . Definiere die Folge $\mathbf{a}_N = ((\mathbf{a}_N)_n)_{n \in \mathbb{N}}$ durch

$$(\mathbf{a}_N)_n = \begin{cases} a_n & \text{für } n \in \mathcal{I}_N, \\ 0 & \text{sonst.} \end{cases}$$

Die Folge \mathbf{a}_N besteht also aus den N betragsmäßig größten Folgengliedern von \mathbf{a} und sonst Nullen. Dann gilt

$$\|\mathbf{a} - \mathbf{a}_N\|_{\ell_q(\mathbb{N})} \leq N^{-r} \|\mathbf{a}\|_{\ell_p(\mathbb{N})} \quad \text{mit} \quad r := \frac{1}{p} - \frac{1}{q} \geq 0.$$

Hinweis: Ordnen Sie die Folge \mathbf{a} betragsmäßig absteigend an.