Informatik IV - Tutorium XVI & XVII

Tut Nr. 11 – Haar-Wavelets, Bivariate Wavelets

David Münch

Universität Karlsruhe (TH)
Fakultät für Informatik
IBDS Prautzsch

Dank an Yusuf für die Zusammenarbeit.

17. Juli 2009

Universität Karlsruhe (TH)
Forschungsuniversität • gegründet 1825

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Inhaltsverzeichnis

Organisatorisches

Email: muenchdavid@gmail.com

https://www.stud.uni-karlsruhe.de/∼uhbro/

Tutorium 16: Freitags 8:00 Uhr - Raum -107
Tutorium 17: Freitags 9:45 Uhr - Raum -119

Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009 $\surd$ $\varnothing 35$ Punkte
- 02.07.2009 $\surd$ $\varnothing 39$ Punkte
- 23.07.2009 ÄNDERUNG! 15:45 Uhr

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009 $\surd$ $\varnothing 35$ Punkte
- 02.07.2009 $\surd$ $\varnothing 39$ Punkte
- 23.07.2009 ÄNDERUNG! 15:45 Uhr

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Ab 140 Punkten gibt es einen Notenbonus von ca. 1/3 Note.

Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.

Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009 $\surd$ $\varnothing 35$ Punkte
- 02.07.2009 $\surd$ $\varnothing 39$ Punkte
- 23.07.2009 ÄNDERUNG! 15:45 Uhr

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Ab 140 Punkten gibt es einen Notenbonus von ca. 1/3 Note.

Abgabe in Zweiergruppe erlaubt und ausdrücklich erwünscht!

Schein / Übungsblätter

Bearbeitete Hausaufgaben können abgeben werden.
Es gibt vier Scheinklausuren mit je 50 Punkten an folgenden Terminen:

- 19.05.2009 $\surd$ $\varnothing 31$ Punkte
- 09.06.2009 $\surd$ $\varnothing 35$ Punkte
- 02.07.2009 $\surd$ $\varnothing 39$ Punkte
- 23.07.2009 ÄNDERUNG! 15:45 Uhr

120 Punkte aus den Scheinklausuren sind notwendig, um einen Schein zu erhalten.

Ab 140 Punkten gibt es einen Notenbonus von ca. 1/3 Note.

Abgabe in Zweiergruppe erlaubt und ausdrücklich erwünscht!

Bitte folgendes Deckblatt verwenden:
`http://www.stud.uni-karlsruhe.de/~unbdh/deckblatt/`

Literatur

Boehm, Prautzsch: Numerical Methods. AK Peters 1993. ISBN 3-528-06350-5
http://www.ubka.uni-karlsruhe.de/hylib-bin/suche.cgi?opacdb=UBKA_OPAC&fbt=7319953&nd=3204657

Ash: Information Theory. Dover 1990. ISBN 0-486-66521-6
http://www.ubka.uni-karlsruhe.de/hylib-bin/suche.cgi?opacdb=UBKA_OPAC&nd=9866904

Goos: Vorlesungen über Informatik. Bd. 4, Springer 1998. ISBN 3-540-60650-5
http://www.ubka.uni-karlsruhe.de/hylib-bin/suche.cgi?opacdb=UBKA_OPAC&fbt=9316367&nd=6568301

Was wollen wir heute erreichen?

Was wollen wir heute erreichen?

- Hausaufgabenblatt 9 besprechen

Was wollen wir heute erreichen?

- Hausaufgabenblatt 9 besprechen
- Einführung in die Wavelet-Kompression

Was wollen wir heute erreichen?

- Hausaufgabenblatt 9 besprechen
- Einführung in die Wavelet-Kompression
- Haar-Wavelets verstehen und anwenden können

Was wollen wir heute erreichen?

- Hausaufgabenblatt 9 besprechen
- Einführung in die Wavelet-Kompression
- Haar-Wavelets verstehen und anwenden können
- Wavelets auf 2D-Bilder erweitern

Was wollen wir heute erreichen?

- Hausaufgabenblatt 9 besprechen
- Einführung in die Wavelet-Kompression
- Haar-Wavelets verstehen und anwenden können
- Wavelets auf 2D-Bilder erweitern
- Wiederholung Simplexalgorithmus

Was wollen wir heute erreichen?

- Hausaufgabenblatt 9 besprechen
- Einführung in die Wavelet-Kompression
- Haar-Wavelets verstehen und anwenden können
- Wavelets auf 2D-Bilder erweitern
- Wiederholung Simplexalgorithmus
- Wahr/Falsch Fragen

Hausaufgabe 44

Ein Bitstrom X, bei dem mit Wahrscheinlichkeit $\beta \gg \frac{1}{2}$ Nullen auftreten, werde in einen Bitstrom Y transformiert. Um zu komprimieren, sollen Blöcke der Form 0^n1, $n = 0, \ldots, 9$ und 0^{10} Huffman kodiert werden.

1. Stellen Sie eine Kodierungstabelle für die Metazeichen 0^n1 und 0^{10} auf. Berechnen Sie zunächst die Auftrittswahrscheinlichkeiten für die Metazeichen (Wörter der Eingabe) 1, 01, 001, ..., 0^91 und 0^90, wenn $\beta = 85\%$ gewählt wird.

   Gehen Sie so vor, daß (bei Verzweigungen von zwei Huffman-Kodewörtern mit gleichem Präfix) stets die unwahrscheinlichere Alternative mit "0" und die wahrscheinlichere mit "1" kodiert wird. Hierdurch wird die Kodierungstabelle eindeutig festgelegt.

2. Die Ausgabe – der von Ihnen ermittelte Huffman-Kode – sei mit Y bezeichnet. Berechnen Sie die (relative) Redundanz von Y und vergleichen Sie sie mit der Redundanz des ursprünglichen Bitstroms der Eingabe X.

Metazeichen	Auftrittsw.	Zahlenwert
1	$1-\beta$	0,1500
01	$\beta(1-\beta)$	0,1275
001	$\beta^2(1-\beta)$	0,1084
0001	$\beta^3(1-\beta)$	0,0921
00001	$\beta^4(1-\beta)$	0,0783
000001	$\beta^5(1-\beta)$	0,0666
0000001	$\beta^6(1-\beta)$	0,0566
00000001	$\beta^7(1-\beta)$	0,0481
000000001	$\beta^8(1-\beta)$	0,0409
0000000001	$\beta^9(1-\beta)$	0,0347
0000000000	β^{10}	0,1969

Eingabe X	Auftrittsw.	Kodierung Y
1	0,1500	110
01	0,1275	100
001	0,1084	011
0001	0,0921	1111
00001	0,0783	1110
000001	0,0666	1010
0000001	0,0566	0101
00000001	0,0481	0100
000000001	0,0409	10111
0000000001	0,0347	10110
0000000000	0,1969	00

$$
\begin{aligned}
L(Y) &= \sum_i p(Y = y_i) \cdot l(y_i) \\
&\approx (15\% + 12{,}75\% + 10{,}84\%) \cdot 3 \text{ bit} \\
&\quad +(9{,}21\% + 7{,}83\% + 6{,}66\% + 5{,}66\% + 4{,}81\%) \cdot 4 \text{ bit} \\
&\quad +(4{,}09\% + 3{,}47\%) \cdot 5 \text{ bit} \ + 19{,}69\% \cdot 2 \text{ bit} \\
&= 3{,}2963 \text{ bit}
\end{aligned}
$$

Für $H(Y)$ gilt:

$$
\begin{aligned}
H(Y) &= -\sum_i p(Y = y_i) \cdot \log p(Y = y_i) \\
&\approx H(\, 15\% \,,\ 12{,}75\% \,,\ 10{,}84\% \,,\ 9{,}21\% \,,\ 7{,}83\% \,,\ 6{,}66\% \,, \\
&\qquad 5{,}66\% \,,\ 4{,}81\% \,,\ 4{,}09\% \,,\ 3{,}47\% \,,\ 19{,}69\% \,) \\
&\approx 3{,}2654 \text{ bit}
\end{aligned}
$$

Die Redundanz beträgt daher

$$
R(Y) = 1 - \frac{H(Y)}{L(Y)} \approx 1 - \frac{3{,}2654 \text{ bit}}{3{,}2963 \text{ bit}} \approx 0{,}009374,
$$

Das relativ kleine Ausgabealphabet – es umfaßt 11 Kodewörter – komprimiert die Eingabe-Bitfolge X also schon beträchtlich, verglichen mit ihrer ursprünglichen Redundanz, die

$$\begin{aligned} R(X) &= 1 - \frac{H(X)}{L(X)} = 1 + \frac{\beta \cdot \log \beta + (1-\beta) \cdot \log(1-\beta)}{1 \text{ bit}} \\ &= 1 + \frac{85\% \cdot \log 85\% + 15\% \cdot \log 15\%}{\text{bit}} \approx 1 - \frac{0{,}6098 \text{ bit}}{\text{bit}} \approx 0{,}3902 \end{aligned}$$

beträgt, also ärgerliche 39%.

Hausaufgabe 45

Zeigen Sie an einem Beispiel, dass ein affiner Unterraum $\vec{y} + K$ des $\mathbb{Z}_2^n$ mehrere kleinste Elemente enthalten kann. Kleinst bedeutet minimale Anzahl von Einsen in den Koordinaten.

Hausaufgabe 45

Zeigen Sie an einem Beispiel, dass ein affiner Unterraum $\vec{y} + K$ des $\mathbb{Z}_2^n$ mehrere kleinste Elemente enthalten kann. Kleinst bedeutet minimale Anzahl von Einsen in den Koordinaten.

Lösung:

Beispiel: Der Unterraum $\begin{pmatrix} 0 \\ 1 \end{pmatrix} + \lambda \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ des $\mathbb{Z}_2^2$ hat zwei kleinste Elemente, nämlich: $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ und $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$. Beide haben eine 1 und somit gleichen Abstand zur $\vec{0}$.

Hausaufgabe 46

Betrachte den Hamming-Code, der die Gleichung $A \cdot x = 0$ löst, wenn gilt:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & a_1 \\ 1 & 1 & 0 & 0 & a_2 \\ 1 & 0 & 1 & 0 & a_3 \\ 0 & 1 & 1 & 1 & a_4 \end{pmatrix}$$

a) Gib alle Codewörter an für den Fall $a_1 = a_2 = a_3 = a_4 = 1$.

Hausaufgabe 46

Betrachte den Hamming-Code, der die Gleichung $A \cdot x = 0$ löst, wenn gilt:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & a_1 \\ 1 & 1 & 0 & 0 & a_2 \\ 1 & 0 & 1 & 0 & a_3 \\ 0 & 1 & 1 & 1 & a_4 \end{pmatrix}$$

a) Gib alle Codewörter an für den Fall $a_1 = a_2 = a_3 = a_4 = 1$.

b) Gib alle $\vec{a} = (a_1 ... a_4)^t$ an, sodass alle 1-Fehler korrigiert und alle 2-Fehler entdeckt werden können.

Hausaufgabe 46

Betrachte den Hamming-Code, der die Gleichung $A \cdot x = 0$ löst, wenn gilt:

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & a_1 \\ 1 & 1 & 0 & 0 & a_2 \\ 1 & 0 & 1 & 0 & a_3 \\ 0 & 1 & 1 & 1 & a_4 \end{pmatrix}$$

a) Gib alle Codewörter an für den Fall $a_1 = a_2 = a_3 = a_4 = 1$.

b) Gib alle $\vec{a} = (a_1 ... a_4)^t$ an, sodass alle 1-Fehler korrigiert und alle 2-Fehler entdeckt werden können.

c) Für welches $\vec{a} = (a_1 ... a_4)^t$ sind auch alle 2-Fehler korrigierbar?

Vergleich von JPEG und JPEG2000

Wavelet-Kompression

Die Wavelet-Kompression ist eine verlustbehaftete Datenkompression speziell für Bilddaten (Video geht auch) z.B. JPEG2000.

- 2D-Wavelet-Transformation durchführen, dabei erhält man genausoviele Koeffizienten wie Pixel.
  Diese Koeffizienten sind einfacher zu komprimieren, weil sich die wichtigen Informationen auf wenige K. verteilen.
- Quantisieren
- Entropie-/Lauflängencodierung

Motivation

1D-Bild der Grösse 4:

$$\gamma_4 = [2, 6, 12, 4]$$

Motivation

1D-Bild der Grösse 4:

$$\gamma_4 = [2, 6, 12, 4]$$

Einfacher Vergröberungsschritt: Berechne Durchschnitt von je zwei benachbarten Pixeln.
Mit den Detailkoeffizienten kann das ursprüngliche Bild wiederhergestellt werden.

$$\gamma_2 = [4, 8], \delta_2 = [2, -4]$$

Motivation

1D-Bild der Grösse 4:

$$\gamma_4 = [2, 6, 12, 4]$$

Einfacher Vergröberungsschritt: Berechne Durchschnitt von je zwei benachbarten Pixeln.
Mit den Detailkoeffizienten kann das ursprüngliche Bild wiederhergestellt werden.

$$\gamma_2 = [4, 8], \delta_2 = [2, -4]$$

$$\gamma_1 = [6], \delta_1 = [2], \delta_2 = [2, -4]$$

Die Wavelet-Transformierte des Bildes ist der Gesamtdurchschnitt des Bildes mit den Detailkoeffizienten in aufsteigender Reihenfolge.

$$[6, 2, 2, -4]$$

Wavelet-Transformation

Idee: Vergöbere das Bild sukzessive und speichere Detailkoeffizienten.
Betrachte ein Bild nicht als Liste von Pixeln, sondern als abschnittsweise konstante Funktion auf $[0, 1)$.
Die Bilder haben die Grösse $2^j, j \geq 0$ Beispiel: TODO

Definition: V_j

V_j ist der Vektorraum aller abschnittsweise konstanten Funktionen auf $[0, 1)$ mit 2^j gleich grossen Abschnitten.

Definition: V_j

V_j ist der Vektorraum aller abschnittsweise konstanten Funktionen auf $[0, 1)$ mit 2^j gleich grossen Abschnitten.

$\Rightarrow$ Jedes Bild der Grösse 2^j ist ein Vektor aus V_j.
Weil für alle j die Funktion (Vektoren) auf $[0, 1)$ definiert sind, ist $V_j \subseteq V_{j+1}$.

Definition: V_j

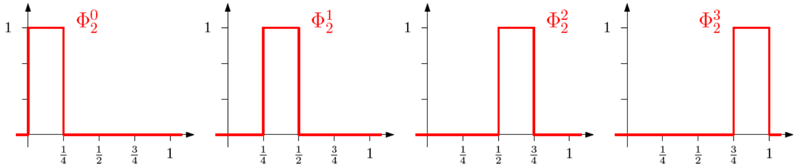
V_j ist der Vektorraum aller abschnittsweise konstanten Funktionen auf $[0,1)$ mit 2^j gleich grossen Abschnitten.

$\Rightarrow$ Jedes Bild der Grösse 2^j ist ein Vektor aus V_j.
Weil für alle j die Funktion (Vektoren) auf $[0,1)$ definiert sind, ist $V_j \subseteq V_{j+1}$.

Die Standardbasis (skalierte Translate) für V_j besteht aus

$$\Phi_j^i(x) = \Phi(2^j x - i) \text{ mit } \Phi(x) = \begin{cases} 1 & \text{falls } 0 \leq x < 1 \\ 0 & \text{sonst} \end{cases}$$

Basisvektoren der Haar-Basis für $j = 2$



Alle Bilder der Grösse $2^j = 4$ lassen sich aus Linearkombinationen dieser Basis darstellen.

Zusammen mit dem Skalarprodukt $< f, g >= \int_0^1 f(x) \cdot g(x) dx$ ist V_j ein euklidischer Vektorraum.

Zusammen mit dem Skalarprodukt $< f, g >= \int_0^1 f(x) \cdot g(x) dx$ ist V_j ein euklidischer Vektorraum.

Weil V_j Untervektorraum von V_{j+1} ist, hat V_j einen orthogonalen Komplementärraum W_j bezüglich dem SKP $< \cdot, \cdot >$ und es gilt $V_j \oplus W_j = V_{j+1}$.

Zusammen mit dem Skalarprodukt $< f, g >= \int_0^1 f(x) \cdot g(x)dx$ ist V_j ein euklidischer Vektorraum.

Weil V_j Untervektorraum von V_{j+1} ist, hat V_j einen orthogonalen Komplementärraum W_j bezüglich dem SKP $< \cdot, \cdot >$ und es gilt $V_j \oplus W_j = V_{j+1}$.

Nun sieht man, dass man ein Bild aus V_{j+1} als ein Bild aus V_j und einem Anteil aus W_j darstellen kann.

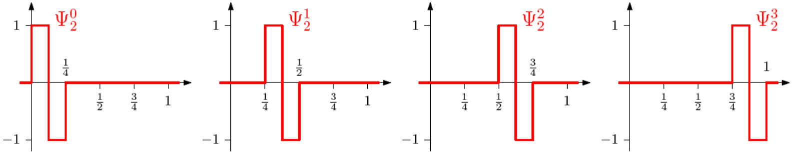
Die Basisfunktionen aus W_j heissen **Wavelets**.

Haar-Wavelets

Die Haar-Wavelets sind die zur Haar-Basis korrespondierenden Basisfunktionen von W_j.

$$\Psi_j^i = \Psi(2^j x - i) \text{ mit } \Psi(x) = \begin{cases} 1 & \text{falls } 0 \leq x < 1/2 \\ -1 & \text{falls } 1/2 \leq x < 1 \\ 0 & \text{sonst} \end{cases}$$

Haar-Wavelets von W_2 zur Haar-Basis des V_j



Weil nach Definition $V_j \perp W_j$ gilt: $V_j \oplus W_j = V_{j+1}$

Ein Bild f der Grösse $2^j = n$ kann man nun wie folgt darstellen:
$f = \sum_i c_n^i \cdot \Phi_n^i$
$f = c_o \cdot \Phi + \sum_i d_0^i \cdot \Psi_0^i + \cdots + \sum_i d_{n-1}^i \cdot \Psi_{n-1}^i$
mit den Analysegleichungen
$c_k^i = 1/2 c_{k+1}^{2i} + 1/2 c_{k+1}^{2i+1}$
$d_k^i = 1/2 c_{k+1}^{2i} - 1/2 c_{k+1}^{2i+1}$

Ein Bild f der Grösse $2^j = n$ kann man nun wie folgt darstellen:
$f = \sum_i c_n^i \cdot \Phi_n^i$
$f = c_o \cdot \Phi + \sum_i d_0^i \cdot \Psi_0^i + \cdots + \sum_i d_{n-1}^i \cdot \Psi_{n-1}^i$
mit den Analysegleichungen
$c_k^i = 1/2 c_{k+1}^{2i} + 1/2 c_{k+1}^{2i+1}$
$d_k^i = 1/2 c_{k+1}^{2i} - 1/2 c_{k+1}^{2i+1}$

oder mit den Zerlegungsgleichungen
$\Phi_k^{2i} = 1/2 \Phi_{k-1}^i + 1/2 \Psi_{k-1}^i$
$\Phi_k^{2i+1} = 1/2 \Phi_{k-1}^i - 1/2 \Psi_{k-1}^i$

Ein Bild f der Grösse $2^j = n$ kann man nun wie folgt darstellen:
$f = \sum_i c_n^i \cdot \Phi_n^i$
$f = c_o \cdot \Phi + \sum_i d_0^i \cdot \Psi_0^i + \cdots + \sum_i d_{n-1}^i \cdot \Psi_{n-1}^i$
mit den Analysegleichungen
$c_k^i = 1/2c_{k+1}^{2i} + 1/2c_{k+1}^{2i+1}$
$d_k^i = 1/2c_{k+1}^{2i} - 1/2c_{k+1}^{2i+1}$

oder mit den Zerlegungsgleichungen
$\Phi_k^{2i} = 1/2\Phi_{k-1}^i + 1/2\Psi_{k-1}^i$
$\Phi_k^{2i+1} = 1/2\Phi_{k-1}^i - 1/2\Psi_{k-1}^i$

Beispiel:
$f = 3\Phi_2^0 + 1\Phi_2^1 + 3\Phi_2^2 + 5\Phi_2^3 \in V_2$
$f = (2\Phi_1^0 + 4\Phi_1^1) + (1\Psi_1^0 - 1\Psi_1^1)$
$f = (3\Phi_0^1) - (1\Psi_0^1) + (1\Psi_1^0 - 1\Psi_1^1)$

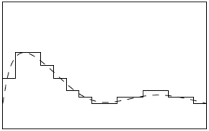
Quantisierung

- Normiere Basisfunktionen:
  $\Phi_k^i \rightsquigarrow 2^{k/2}\Phi_k^i$
  $\Psi_k^i \rightsquigarrow 2^{k/2}\Psi_k^i$

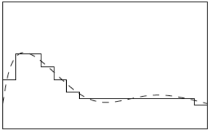
Quantisierung

- Normiere Basisfunktionen:
  $\Phi_k^i \rightsquigarrow 2^{k/2}\Phi_k^i$
  $\Psi_k^i \rightsquigarrow 2^{k/2}\Psi_k^i$
- Setze die kleinsten Koeffizienten nach der L_2-Norm auf Null.

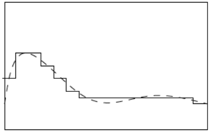
Wavelet-Kompression



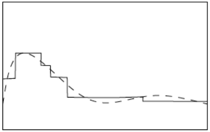
12 von 12 Koeffizienten



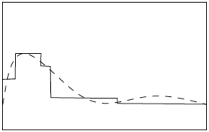
10 von 12 Koeffizienten



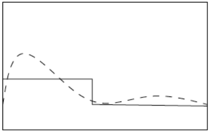
8 von 12 Koeffizienten



6 von 12 Koeffizienten



4 von 12 Koeffizienten



2 von 12 Koeffizienten

Aufgabe

Gegeben: die Standard Haar-Grundfunktion $\Phi(x)$

a) Ersetze das Haar-Wavelet durch die Funktion $\Psi(x) = \Phi(2x)$ und gib eine Zerlegungsgleichung und die Zweiskalen-Relation an.

Aufgabe

Gegeben: die Standard Haar-Grundfunktion $\Phi(x)$

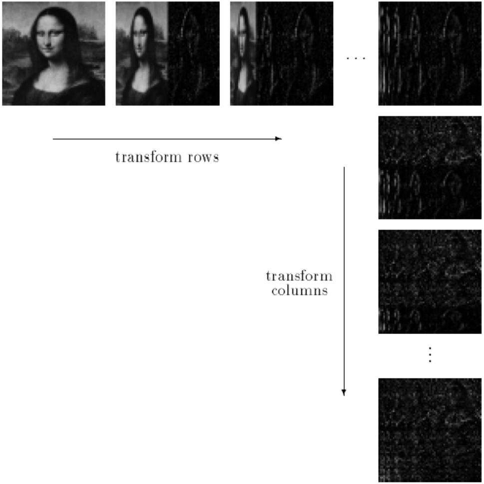
a) Ersetze das Haar-Wavelet durch die Funktion $\Psi(x) = \Phi(2x)$ und gib eine Zerlegungsgleichung und die Zweiskalen-Relation an.

b) Sei $V_1 = span\{\Phi(3x), \Phi(3x-1), \Phi(3x-2)\}$ und $< f, g >= \int_0^1 f(x)g(x)dx$ ein Skalarprodukt. Gib dazu orthogonale Wavelets an, die das orthogonale Komplement W_0 von $V_0 = span\{\Phi(x)\}$ in V_1 aufspannen. (Es gilt also $V_1 = V_0 \oplus W_0$.)

Und jetzt das ganze bivariat...

- Standard-Zerlegung
- Nicht-Standard-Zerlegung

Standard-Zerlegung



Nicht-Standard-Zerlegung

Aufgabe

Berechne die Nicht-Standard-Zerlegung für die Matrix

$$A = \begin{pmatrix} 1 & -1 & 0 & 8 \\ 5 & 7 & 4 & 0 \\ 8 & 2 & 6 & 4 \\ 6 & 4 & 2 & 8 \end{pmatrix}$$

.

Lesen!

Wavelets for computer graphics: A primer
http://www.cis.udel.edu/~amer/CISC651/wavelets_for_computer_graphics_Stollnitz.pdf

Simplexverfahren

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Lineares Programm

Ein *Lineares Programm* besteht aus einer linearen Zielfunktion

$$c_1x_1 + c_2x_2 + \ldots + c_nx_n$$

die maximiert/minimiert werden soll, sowie m Ungleichungen der Form

$$\begin{array}{ccccccccc}
a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \substack{\leq \\ \geq \\ =} & b_1 \\
\vdots & & & & \ddots & + & & & \vdots \\
a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \substack{\leq \\ \geq \\ =} & b_m
\end{array}$$

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Lineares Programm

Ein *Lineares Programm* besteht aus einer linearen Zielfunktion

$$c^\top x$$

die maximiert/minimiert werden soll, sowie m Ungleichungen der Form

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \overset{\le}{\underset{=}{\ge}} & b_1 \\ \vdots & & & & \ddots & + & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \overset{\le}{\underset{=}{\ge}} & b_m \end{array}$$

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Lineares Programm

Ein *Lineares Programm* besteht aus einer linearen Zielfunktion

$$c^\top x$$

die maximiert/minimiert werden soll, sowie m Ungleichungen der Form

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & \leq & b_1 \\ \vdots & & & & \ddots & + & & & \vdots \\ a_{m1}x_1 & + & a_{m2}x_2 & + & \cdots & + & a_{mn}x_n & \leq & b_m \end{array}$$

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Lineares Programm

Ein *Lineares Programm* besteht aus einer linearen Zielfunktion

$$c^\top x$$

die maximiert/minimiert werden soll, sowie m Ungleichungen der Form

$$Ax \leq b$$

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Lineares Programm

Ein *Lineares Programm* besteht aus einer linearen Zielfunktion

$$c^\top x$$

die maximiert/minimiert werden soll, sowie m Ungleichungen der Form

$$Ax \leq b$$

Dabei sind $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$.

Das Simplex-Verfahren ist ein Algorithmus zur Lösung linearer Programme.

Lineares Programm in Informatik-4-Normalform

Ein *Lineares Programm* besteht aus einer linearen Zielfunktion

$$c^\top x$$

die maximiert werden soll, sowie m Ungleichungen der Form

$$A'x + b \geq 0$$

Aufgabe

Ein Kaffeehändler will zwei Sorten Kaffee einkaufen, eine teure Sorte A und eine billigere Sorte B. Von der Sorte A kann er höchstens $120kg$, von der Sorte B höchstens $180kg$ bekommen. Aus diesen beiden Sorten stellt er zwei Mischungen her: Die erste Mischung soll 20% der Sorte A und 80% der Sorte B, die zweite Mischung soll 60% der Sorte A und 40% der Sorte B enthalten. Der Verkaufspreis der ersten Mischung beträgt 12 Euro, der zweiten Mischung 16 Euro je Kilogramm. Welche Menge muss der Händler von jeder Mischung herstellen, damit er einen möglichst hohen Erlös erreicht? Bestimme die Lösung mit dem Simplexverfahren.

Das Ende Ist Nahe...

Ein paar Wahr/Falsch Fragen.

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben.

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X
- Es gibt keinen Code C so dass $L(C) < L(S)$ wobei S der Shannoncode ist.

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X
- Es gibt keinen Code C so dass $L(C) < L(S)$ wobei S der Shannoncode ist. X

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X
- Es gibt keinen Code C so dass $L(C) < L(S)$ wobei S der Shannoncode ist. X
- In einem mehrschichtigen Perzeptron reicht eine verdeckte Schicht nicht aus, um jede berechenbare stetige Funktion beliebig gut anzunähern.

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X
- Es gibt keinen Code C so dass $L(C) < L(S)$ wobei S der Shannoncode ist. X
- In einem mehrschichtigen Perzeptron reicht eine verdeckte Schicht nicht aus, um jede berechenbare stetige Funktion beliebig gut anzunähern. X

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X
- Es gibt keinen Code C so dass $L(C) < L(S)$ wobei S der Shannoncode ist. X
- In einem mehrschichtigen Perzeptron reicht eine verdeckte Schicht nicht aus, um jede berechenbare stetige Funktion beliebig gut anzunähern. X
- Ein async. Hopfield-Netz erreicht immer nach endlich vielen Schritten einen stabilen Zustand.

Wahr/Falsch-Fragen

- Bei evolutionären Algorithmen können die Eltern im Gegensatz zu den genetischen Algorithmen immer überleben. X
- Es gibt keinen Code C so dass $L(C) < L(S)$ wobei S der Shannoncode ist. X
- In einem mehrschichtigen Perzeptron reicht eine verdeckte Schicht nicht aus, um jede berechenbare stetige Funktion beliebig gut anzunähern. X
- Ein async. Hopfield-Netz erreicht immer nach endlich vielen Schritten einen stabilen Zustand. ✓

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$.

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik.

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik. X

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik. X
- In einem rekurrenten Neuronalen Netz kann es Rückwärtskanten geben.

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik. X
- In einem rekurrenten Neuronalen Netz kann es Rückwärtskanten geben. ✓

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik. X
- In einem rekurrenten Neuronalen Netz kann es Rückwärtskanten geben. ✓
- Der Lernalgorithmus für das einfache Perzeptron terminiert immer.

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p||q)$ ist eine Metrik. X
- In einem rekurrenten Neuronalen Netz kann es Rückwärtskanten geben. ✓
- Der Lernalgorithmus für das einfache Perzeptron terminiert immer. X

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik. X
- In einem rekurrenten Neuronalen Netz kann es Rückwärtskanten geben. ✓
- Der Lernalgorithmus für das einfache Perzeptron terminiert immer. X
- Haben die Codeworte x_i mindestens den Abstand

$$d(x_i, x_j) \geq 2e + 1 \qquad \text{für alle } i \neq j$$

können f falsche Bits korrigiert werden, wenn $f \leq e$.

Wahr/Falsch-Fragen

- Ein Kanal ist deterministisch falls $H(Y|X) = 0$. ✓
- Der Kullback-Leibler-Abstand $D(p\|q)$ ist eine Metrik. X
- In einem rekurrenten Neuronalen Netz kann es Rückwärtskanten geben. ✓
- Der Lernalgorithmus für das einfache Perzeptron terminiert immer. X
- Haben die Codeworte x_i mindestens den Abstand

$$d(x_i, x_j) \geq 2e + 1 \qquad \text{für alle } i \neq j$$

können f falsche Bits korrigiert werden, wenn $f \leq e$. ✓

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar.

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X
- Bei der Übertragung von LZW kodierten Daten braucht das Wörterbuch oft den meisten Platz.

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X
- Bei der Übertragung von LZW kodierten Daten braucht das Wörterbuch oft den meisten Platz. X

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X
- Bei der Übertragung von LZW kodierten Daten braucht das Wörterbuch oft den meisten Platz. X
- Bei der Wavelettransformation findet kein Informationsverlust statt.

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X
- Bei der Übertragung von LZW kodierten Daten braucht das Wörterbuch oft den meisten Platz. X
- Bei der Wavelettransformation findet kein Informationsverlust statt. ✓

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X
- Bei der Übertragung von LZW kodierten Daten braucht das Wörterbuch oft den meisten Platz. X
- Bei der Wavelettransformation findet kein Informationsverlust statt. ✓
- Nach Dueck, Scheuer und Wallmeister ist das Simulierte Tempern dem Sintflut- und Schwellwert-Verfahren überlegen.

Wahr/Falsch-Fragen

- Reguläre Kodes sind stets dekodierbar. X
- Bei der Übertragung von LZW kodierten Daten braucht das Wörterbuch oft den meisten Platz. X
- Bei der Wavelettransformation findet kein Informationsverlust statt. ✓
- Nach DUECK, SCHEUER und WALLMEISTER ist das Simulierte Tempern dem Sintflut- und Schwellwert-Verfahren überlegen. X

Quellen

Pajor - Informatik 4 Tutorium SS2007

Prautzsch - Skript Informatik 4 SS2008

Wavelets for computer graphics: A primer
`http://www.cis.udel.edu/~amer/CISC651/wavelets_for_computer_graphics_Stollnitz.pdf`

Wikipedia

Reflexion

Was haben wir heute gelernt?

Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 9 besprochen

Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 9 besprochen
- Theorie der Haar-Wavelets verstanden

Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 9 besprochen
- Theorie der Haar-Wavelets verstanden
- Anwendung von Wavelets

Reflexion

Was haben wir heute gelernt?

- Hausaufgabenblatt 9 besprochen
- Theorie der Haar-Wavelets verstanden
- Anwendung von Wavelets
- Simplexalgorithmus wiederholt

Noch Fragen?

Vorschau

Vorschau

- Besprechung 4. Scheinklausur

Vorschau

- Besprechung 4. Scheinklausur
- Sonst nichts mehr.

Bis zum letzten Mal