



Grundlagen der Multimediatechnik

Wintersemester 2021/22

Übungsblatt 2

8. November 2021

Wichtig: Bitte kennzeichnen Sie Ihre Abgabe mit Ihrem Namen und dem Namen Ihres Übungspartners.

Laden Sie bitte Ihre Abgabe als PDF-/ZIP-Datei auf der ILIAS-Plattform hoch. Andere Dateiformate sowie Scans von handgeschriebenen Abgaben werden nicht gewertet. Achten Sie darauf, nur kompilierbaren, kommentierten Code abzugeben. Nicht-kompilierbarer Code wird mit **0 Punkten** bewertet!

Alle Abgaben müssen folgender Namenskonvention entsprechen: `gmt_uebungXX_nachname1_nachname2.format`

Aufgabe 1: Informationstheorie

[10 Punkte]

1. Gegeben sei eine Quelle Q mit 4 Symbolen: A, B, C und D .

[5 Punkte]

Die Auftrittswahrscheinlichkeiten der Symbole seien gegeben mit:

$$p(A) = 0.5$$

$$p(B) = 0.2$$

$$p(C) = 0.18$$

$$p(D) = 0.12$$

- (a) Wie groß ist die Information I eines einzelnen Symbols?
- (b) Wie groß ist der Entscheidungsgehalt H_0 der Quelle?
- (c) Wie groß ist die Entropie $H(X)$ der Quelle?
- (d) Wie groß wäre die Entropie bei gleicher Auftrittswahrscheinlichkeit der 4 Symbole?
- (e) Wie groß ist die Redundanz R_Q der Quelle?

2. Gegeben sei nun folgende Binärsequenz (die Leerzeichen dienen nur der Übersicht): [5 Punkte]

0001 0100 1010 0010 1000 1111 0010 0010

- (a) Berechnen Sie für die Sequenz:

i. Den Entscheidungsgehalt

ii. Die Entropie

iii. Die Redundanz pro Symbol (R) und für die gesamte Nachricht (R_C , mit $\bar{L}_C = 1$)

- (b) Es sind jeweils zwei benachbarte Bits zu einem Zeichen zusammenzufassen. Beantworten Sie für die sich daraus ergebende Symbolfolge alle 3 Fragen aus dem vorherigen Aufgabenteil (mit $\bar{L}_C = 2$). Warum unterscheiden sich R und R_C nicht?

Aufgabe 2: Huffman

[15 Punkte]

Ziel dieser Aufgabe ist es, den Algorithmus der Huffman-Kodierung zu implementieren. Die Huffman-Kodierung ist eine Entropiekodierung und ordnet jedem Quellsymbol unter Berücksichtigung deren Häufigkeiten Codewörter mit variabler Länge zu.

Verwenden Sie das mitgelieferte Notebook `Uebung02.ipynb` zur Bearbeitung dieser Aufgabe. Dort wird eine Klasse `node` mitgeliefert, die durch ihren Namen, einer zugehörigen Häufigkeit und ihre zwei Kinderknoten definiert ist.

Anmerkung: Für die Vervollständigung des Skriptes sind keine externen Bibliotheken notwendig und sind somit für die Bearbeitung dieser Aufgabe unzulässig (**keine imports!**).

1. Vervollständigen Sie die `create_Huffman()`-Funktion, die als Input einen Eingabestring erhält und den entsprechenden Huffman-Tree zur Kodierung zurückgibt. Verwenden Sie für die Implementierung des Baumes die im Skript mitgelieferte Node-Klasse. Speichern Sie den jeweiligen `char`-Value im Namen der Node. [6 Punkte]
2. Implementieren Sie die `getCode()`-Funktion, die den Huffman-Baum als Input bekommt und die Kodierung der Symbole zurückgibt. [3 Punkte]
3. Implementieren Sie die `encoding()`-Funktion, die einen String gemäß des zuvor implementierten Huffman-Baumes kodiert und zurückgibt. [2 Punkte]
4. Anschließend implementieren Sie die `decoding()`-Funktion, die einen Binärstring gemäß des zuvor implementierten Huffman-Baumes in Klartext dekodiert. [3 Punkte]
5. Weshalb ist die Kodierung immer eindeutig? [1 Punkt]

Abgabe: Dienstag, 16. November 2021, 08:00 Uhr im ILIAS-System