

Laborübung 1

Die Bearbeitung der Aufgaben erfolgt in Gruppen zu je max zwei Teilnehmern. Alle Aufgaben sind schriftlich zu lösen und vor dem nächsten Labortermine per eMail einzureichen. Eine vollständige Abgabe besteht aus den von Ihnen erstellten Octave Dateien (.m) und dem Protokoll (PDF). Alle praktische Realisierungen sind dem Dozenten im Rahmen der Gruppenrücksprache zu präsentieren. Erfolgreiche Vorführungen werden auf dem Aufgabenblatt schriftlich quittiert. Eine erfolgreiche Laborteilnahme ist Voraussetzung für die Teilnahme an der Klausur am Ende des Semesters. Bitte bedenken Sie, dass eine gute Vorbereitung zu den Laborterminen unabdingbar ist.*

Im Labor zur Vorlesung „Kanal- und Quellencodierung“ wird GNU Octave 4.4 benutzt. Octave ist weitgehend codekompatibel zu Matlab, beide Tools verwenden eine Skriptsprache, die interpretiert werden muss. Wenn Sie noch keine Erfahrungen mit GNU Octave haben, nutzen Sie diese erste Übung, um sich damit vertraut zu machen bzw. Ihr Wissen aufzufrischen.

GNU Octave ist eine Open Source Software und kann kostenfrei heruntergeladen und installiert werden. Sie finden diese Software für alle gängigen Betriebssysteme, u. a. Windows und Linux, kostenlos unter der folgenden Adresse: <http://www.gnu.org/software/octave>. Im Labor für Digitaltechnik ist Octave bereits installiert und kann sofort benutzt werden.

Aufgabe 1

Starten Sie GNU Octave 4.4 und machen Sie sich mit der graphischen Oberfläche vertraut. Suchen Sie sich im Internet ein Tutorial, um einen ersten Einblick zu bekommen. Die offizielle Octave Online-Doku finden Sie unter <https://www.gnu.org/software/octave/doc/interpreter>. Nutzen Sie auch die interne Hilfe (z.B. Eingabe `help <befehl>` im Command Window) von Octave, um Sprachelemente zu finden und zu verstehen.

Beim Öffnen zeigt der File Browser auf das aktuelle Arbeitsverzeichnis bzw. beim ersten Start auf das Verzeichnis des Programmes selbst. Ändern Sie zu allererst das Arbeitsverzeichnis auf ihren persönlichen (gruppenspezifischen) Bereich auf dem Server.

GNU Octave verwendet die gleichen primären GUI-Elemente (File Manager, Workspace, Command History, Command Window) wie Matlab. Im Command Window können Befehle eingetragen und direkt ausgeführt werden. Für die Umsetzung umfangreicherer Algorithmen empfiehlt sich das Anlegen von Skripten als m-File (entspricht Programmen bzw. Funktionen in anderen Programmiersprachen). m-Files bestehen

im Wesentlichen aus einer Folge von Befehlen. Es ist hilfreich zu verstehen, dass Octave grundsätzlich alles als Matrizen behandelt, z.B. sind Vektoren Matrizen mit nur einer Zeile oder Spalte.

Erstellen Sie zur Übung mit dem integrierten Editor einige Skripte (Folge von Befehlen) zur Übung und arbeiten Sie diese ab. Haben Sie keine Angst davor einfach ein bisschen herumzuspielen. Machen Sie sich dann insbesondere damit vertraut wie man

1. grundlegende Operationen anwendet, z.B. Berechnung einer Summe, eines Skalarprodukts, Betrags, Potenzen, usw.
2. Zufallszahlen generiert
3. Schleifen und if-Abfragen umsetzt
4. selbst Funktionen definiert und aufruft
5. Funktionsgraphen plotten kann, z.B. eine Parabel, Sinusfunktion von $0 \dots 2\pi$, ...
6. Variablen oder Daten in Dateien speichert bzw. einliest und auf dem Bildschirm ausgibt

Aufgabe 2

Nach Shannon ist die untere Grenze für die mittlere Codewortlänge (in Bits) für die Codierung einer Zufallsvariablen durch ihre Entropie gegeben. Dies soll hier näher erörtert und am Beispiel untersucht werden. Zur Analyse wird die Häufigkeitsverteilung grafisch mittels eines Histogramms dargestellt. Ein Histogramm ist ein Säulendiagramm, in dem gesammelte Daten zu Klassen zusammengefasst werden. Die Größe einer Säule entspricht dabei der Anzahl der Daten in einer Klasse.

1. Erzeugen Sie einen Vektor gleichverteilter Zufallszahlen mit 5000 Samples. Stellen Sie diese in einem Histogramm in 256 Klassen gleicher Breite dar und lassen Sie sich die Anzahl der Elemente zurückgeben, die in jede Klasse fällt. Schreiben Sie dann eine einfache Funktion, um die Entropie der Verteilung

$$\sum_{\mu} p_{\mu} \cdot \log_2 \frac{1}{p_{\mu}}$$

zu schätzen. μ stellt die Zufallsvariable von Interesse dar. Welche Schätzwerte erhalten Sie?

2. Wenn Sie die Anzahl der Klassen ändern, hat dieses auch einen Einfluss auf die Entropieschätzung. Erläutern Sie die Zusammenhänge. Gehen Sie auch darauf ein was passiert, wenn die Anzahl der Klassen beliebig groß gemacht werden.

3. Erzeugen Sie einen neuen Vektor mit Zufallszahlen (gleiche Anzahl von Samples) und zeigen Sie wie sich die Entropie ändert. Erläutern Sie die Hintergründe. Wie könnte eine allgemeingültige untere Grenze für die geschätzte Codewortlänge ermittelt werden?
4. In der ZIP-Datei zu diesem Laborterminal sind zwei Bilddateien (im PPM-Format) enthalten. Öffnen Sie nacheinander beide Bilddateien in Octave und analysieren Sie für beide Bilddateien die Verteilung der Intensitätswerte mittels Histogramm der Pixelwerte (wieder mit 256 Klassen). Berechnen Sie dann jeweils die Entropie des Bildes mit der von Ihnen bereits geschriebenen Funktion. Vergleichen und deuten Sie die Ergebnisse für beide Bilder.
5. Die Octave-Funktion *ecdf* berechnet die empirische Verteilungsfunktion, im wesentlichen entspricht diese dem Integral des Histogramms. Verwenden Sie diese Funktion, um die kumulative Verteilung von beiden Bildern darzustellen. Dazu müssen Sie die Bildvariable in einen Spaltenvektor umformen. Plotten Sie die Verteilungsfunktion für jedes Bild und vergleichen Sie das Ergebnis.
6. Angenommen Sie möchten die Bilder zum Zweck einer effizienteren Speicherung oder Übertragung komprimieren. Beschreiben Sie ob und wie mit Hilfe der empirischen Verteilungsfunktion eine Kompression der Bilddaten möglich wäre.

Ab hier bitte keine Eintragungen vornehmen!

Aufgabe 1	Aufgabe 2