Dr. Jürg M. Stettbacher
Neugutstrasse 54
CH-8600 Dübendorf

Telefon: +41 43 299 57 23
E-Mail: dsp@stettbacher.ch

Information und Entropie

Praktikum

Version 4.02
2018-03-10

Zusammenfassung: In diesem Praktikum geht es darum, ein Programm zu schreiben, das von gegebenen Daten in einer Datei die Information und die Entropie bestimmt.

Inhaltsverzeichnis

1 Einleitung

In der Informationstheorie betrachten wir den Ausgang einer Datenquelle als Strom von Zufallsvariablen X_k mit $k = 0, 1, 2, \ldots$ und jedes Symbol x_n mit $n = 0 \ldots N-1$, das eine Zufallsvariable annehmen kann, als Zufallsereignis. Sind die Auftretenswahrscheinlichkeiten $P(x_n)$ bekannt, so kann für jedes Symbol x_n die Information $I(x_n)$ berechnet werden und für die Quelle der mittlere Informationsgehalt, also die Entropie H.

In diesem Praktikum betrachten wir verschiedene Datenquellen. Von jeder Quelle liegt uns ein endlicher Ausschnitt ihres Ausgangs in Form einer Datei vor. Es handelt sich um ASCII-Dateien (*.txt) und jedes ASCII-Zeichen daraus stellt ein Symbol x_n dar. Gesucht sind jeweils der Informationsgehalt $I(x_n)$ für jedes Symbol x_n einer Quelle, sowie die Entropie H der Quelle.

2 Dateien

Die folgenden Dateien stehen Ihnen für das Bearbeiten des Praktikums zur Verfügung:

- Template für Programmcode: *Entropy_template.java*
- Testdaten: *source_1.txt* bis *source_6.txt*

Das Java-Template enthält den grössten Teil des notwendigen Codes für das Lösen der Aufgaben. Die Dateien *source_1.txt* bis *source_4.txt* enthalten Testdaten aus vier verschiedenen binären Quellen, wobei jedes Symbol mit einem Byte dargestellt wird. Die Dateien *source_5.txt* und *source_6.txt* enthalten deutschen Text im ASCII-Format ohne Umlaute und ohne Ziffern.

3 Aufgabe

> Öffnen und betrachten Sie die Testdateien erst, wenn es die Aufgabenstellung verlangt.

Lösen sie die folgenden Teilaufgaben.

1. Verwenden Sie das Java-Template, um ein Programm *Entropy.java* zu schreiben. Sie brauchen nur an den bezeichneten Stellen in der Vorlage den Code zu ergänzen[1].

   Ziel: Das Programm soll eine bestimmte ASCII-Datei mit Testdaten öffnen und zeichenweise einlesen. Der Name der Datei ist auf der Kommandozeile als Argument anzugeben. Für die gelesenen Zeichen, resp. Symbole wird zuerst ein Histogramm erzeugt, das angibt, wie oft jedes der N verschiedenen Symbole vorkommt. Dabei werden nur druckbare Zeichen berücksichtigt, das heisst, dass die ASCII-Steuerzeichen zu ignoriert sind. Aus dem Histogramm berechnet das Programm zuerst die Wahrscheinlichkeit $P(x_n)$, dann den Informationsgehalt $I(x_n)$ von jedem Zeichen x_n und schliesslich die Entropie H der Quelle.

   Das Programm lässt sich mit dem folgenden Befehl auf einer Kommandozeile[2] übersetzen:

   ```
   > javac Entropy.java
   ```

   Führen Sie das Programm aus und betrachten Sie die Ausgabe:

   ```
   > java Entropy data/source_1.txt
   ```

[1] Sie dürfen auch eine andere Programmiersprache verwenden. Das Template steht jedoch nur für Java zur Verfügung.

[2] Verwenden Sie unter Linux eine normale Shell, zum Beispiel *bash* und unter Windows den Command Prompt *cmd.exe*. Voraussetzung ist ferner, dass der Systempfad so gesetzt ist, dass die Programme *javac* und *java* gefunden werden. Selbstverständlich kann auch eine integrierte Entwicklungsumgebung verwendet werden. Für das Praktikum ist dies jedoch nicht notwendig.

Entropy: 1 Bit pro Symbol
1 Millionsymbol -> Darstellung mit 1Mio. Bit
1 Mio Bit -> 125'000 Kilobyte

Repition: Quellentheorem

2. Betrachten Sie die Ausgaben des Programms für die Testdatei *source_1.txt:*

   (a) Wie gross ist das N der Quelle[3]? 2 Symbole

   (b) Welche Symbole x_n produziert die Quelle? {-,0}

   (c) Wie gross sind die Wahrscheinlichkeiten $P(x_n)$ der Quelle? nahezu 50/50

   (d) Überlegen Sie, auf welche Dateigrösse sich die Testdatei *source_1.txt* komprimieren lassen sollte. 75-80%

   (e) Verwenden Sie beispielsweise das Programm *zip* (Linux[4]) oder *pkzip* (Windows), um die Testdatei *source_1.txt* zu komprimieren. Überprüfen Sie die Dateigrösse. von 977KB auf 155 KB

   (f) Was ist das Fazit aus den Resultaten?

   Die Datei lässt sich um fast 80% komprimieren

3. Wiederholen Sie die vorhergehende Aufgabe mit der Testdatei *source_2.txt*.

   (a) Gibt es Unterschiede? Minimale Unterschied - es handelt sich um Tausendstel

   (b) Falls ja, wie sind die Unterschiede erklärbar? Das Vorkommen der Zeichen hat sich geändert

   (c) Wenn Sie eine Erklärung gefunden haben: Öffnen und betrachten Sie die beiden Testdateien in einem Texteditor[5]. Was stellen Sie fest?

   Die beiden Dateien sind anderst angeordnet

4. Nun drehen wir den Spiess um und verwenden die Testdaten *source_3.txt* und *source_4.txt*:

   (a) Berechnen (resp. schätzen) Sie die Entropie H der betreffenden Quellen, indem Sie untersuchen, wie stark sich die Testdateien komprimieren lassen. sehr schwer zu schätzen, vermutlich sehr ähnlich

   (b) Vergleichen Sie die Resultate mit der Ausgabe des Java-Programms *Entropy* von oben.

   (c) Was stellen Sie fest? Erklären Sie allfällige Unterschiede. kann keine unterschiede feststellen

   (d) Wenn Sie eine Erklärung gefunden haben: Öffnen und betrachten Sie die beiden Testdateien in einem Texteditor. Was stellen Sie fest?

   c) Die Zeichenlänge ist unterschiedlich

   4b) Ergebnisse sind nahezu identisch

5. Führen Sie nun das Programm *Entropie.java* mit den Testdaten *source_5.txt* aus.

   (a) Wie gross ist die Entropie H dieser Quelle? 3.720492766070573

---

[3] Bitte unterscheiden Sie zwischen der Quelle und der Datei. Die Quelle mit ihren inneren Eigenschaften erzeugt einen unendlich langen Strom von Symbolen. Die Dateien enthalten nur einen endlichen Ausschnitt davon.

[4] Unter Linux können Sie den folgenden Befehl für die Kompression verwenden:
```
> zip source_1.zip data/source_1.txt
```

[5] Verwenden Sie zum Beispiel *less* (Linux), resp. *notepad2* oder *notepad++* (Windows).

(b) Wie vertrauenswürdig sind die Resultate, die das Programm anzeigt?
Aufgrund der statistischen Abhängigkeit sind die Resultate nur bedingt vertrauenswürdig

6. Führen Sie nun das Programm *Entropie.java* mit den Testdaten *source_6.txt* aus.

   (a) Wie gross ist die Entropie H dieser Quelle? 4.35040848177349

   (b) Wenn wir davon ausgehen, dass die berechnete Entropie nicht exakt stimmt: Ist die wahre Entropie dann grösser oder kleiner als der berechnete Wert? Begründen Sie Ihre Antwort nachdem Sie sich den Inhalt der Datei angesehen haben oder lösen Sie die folgende Zusatzaufgabe. Kleiner, wieder aufgrund der Statistik von Buchstaben einer Sprache

7. Fakultative Zusatzaufgabe:

   Schreiben Sie jetzt ein zweites Programm *Entropy2.java*, indem Sie das erste abändern. Das Programm soll für ein gegebenes Zeichen ζ die bedingte Entropie $H(X|\zeta)$ des Nachfolgezeichens ermitteln.

$$H(X|\zeta) = \sum_{x_m} P(x_m|\zeta) \cdot log_2 \frac{1}{P(x_m|\zeta)} \tag{1}$$

   Die Formel bedeutet dies: Es wird eine Entropie, resp. ein mittlerer Informationsgehalt aller Symbole x_m berechnet. Die Symbole x_m sind dabei jene, die in der Datei jeweils direkt auf das Symbol ζ folgen. X ist die Zufallsvariable, welche die Symbole x_m produziert[6].

   Beachten Sie, dass Sie die Wahrscheinlichkeiten $P(x_m|\zeta)$ direkt aus der Testdatei ermitteln können.

   Der Aufruf des neuen Programms soll beispielsweise so aussehen:

```
> java Entropy2 data/source_6.txt c
```

   Dabei ist das letzte Argument (der Buchstabe c) das Zeichen ζ.

   (a) Wählen Sie nacheinander verschiedenen Zeichen ζ aus dem Vorrat der Quelle *source_6.txt* und beobachten Sie die Resultate. Was stellen Sie fest?

   (b) Welches sind typische Extremal-Beispiele?

   (c) Welche Schlussfolgerung ziehen Sie für die Entropie H der Quelle *source_6.txt*?

[6] Aus diesem Grund schreiben wir X in $H(X|\zeta)$ gross. Wir nennen $H(X|\zeta)$ in diesem Fall die *bedingte Entropie der Zufallsvariable X, gegeben das vorausgehende Symbol* ζ.