Kodierung mit variabler Länge: Datenkompression und Verteilungsanpassung

Georg Böcherer

Lehrstuhl für Nachrichtentechnik
Technische Universität München
Email: georg.boecherer@ieee.org

March 2016

Inhaltsverzeichnis

Wahrscheinlichkeitsrechnung und Informationsmaße

Logarithmus

Der *Logarithmus zur Basis 2* (oder *Zweierlogarithmus*) und der natürliche Logarithmus sind definiert als

$$\log_2 x = a \;\Leftrightarrow\; 2^a = x, \quad \ln x = a \;\Leftrightarrow\; e^a = x. \tag{1}$$

Aufgabe 1.

1. Für welche reellen Zahlen x ist der Logarithmus definiert?
2. Drücken Sie $\log_2 x$ mittels des natürlichen Logarithmus aus.
3. Verwenden Sie die Definition des Zweierlogarithmus, um die folgenden Logarithmusregeln zu zeigen.

$$\log_2(xy) = \log_2 x + \log_2 y \tag{2}$$

$$x \log_2 y = \log_2(y^x) \tag{3}$$

$$-\log_2 x = \log_2 \frac{1}{x}. \tag{4}$$

Logarithmus

Aufgabe 2.

1. Die Ableitung des natürlichen Logarithmus ist $\frac{\partial \ln x}{\partial x} = \frac{1}{x}$. Verwenden Sie dies, um $\frac{\partial \log_2 x}{\partial x}$ zu berechnen. Drücken Sie ihr Ergebnis mittels des Zweierlogarithmus aus.
2. Zeigen Sie

$$\log_2 x \leq (x-1)\log_2 e. \tag{5}$$

Wann gilt Gleichheit?

Zufallsvariablen

- Zufallsvariable X
- Ergebnismenge $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$
- Verteilung $P_X \colon \mathcal{X} \to \mathbf{R}$, $a \mapsto P_X(a) = \Pr\{X = a\}$ und

$$\forall a \in \mathcal{X} \colon P_X(a) \geq 0 \tag{6}$$

$$\sum_{a \in \mathcal{X}} P_X(a) = 1. \tag{7}$$

- Träger: $\operatorname{supp} P_X = \{a \in \mathcal{X} \colon P_X(a) > 0\}$.

Gemeinsame Verteilung

- Seien X, Y zwei Zufallsvariablen mit gemeinsamer Verteilung $P_{XY}\colon \mathcal{X} \times \mathcal{Y} \to \mathbf{R}$.
- Die Randverteilungen von X und Y sind

$$P_X(a) = \sum_{b \in \mathcal{Y}} P_{XY}(a, b), \quad P_Y(b) = \sum_{a \in \mathcal{X}} P_{XY}(a, b). \tag{8}$$

- Falls $P_Y(b) > 0$, dann ist die Verteilung von X bedingt auf Y geben durch

$$P_{X|Y}(a|b) = \frac{P_{XY}(a, b)}{P_Y(b)}. \tag{9}$$

Gemeinsame und bedingte Verteilung: Aufgaben

Aufgabe 3.

1. Zeigen Sie: für jedes $b \in \operatorname{supp} P_Y$ ist $P_{X|Y}(\cdot|b)$ eine Verteilung.
2. Zeigen Sie: Falls $P_X(a) = 0$ dann $P_{XY}(a, b) = 0$ für alle $b \in \mathcal{Y}$.
3. Sei $P_Y(b) = 0$. Zeigen Sie, dass die Gleichung

$$P_{X|Y}(a|b)P_Y(b) = P_{XY}(a, b) \tag{10}$$

erfüllt ist für jedes $a \in \mathcal{X}$ und jede Wahl von $P_{X|Y}(\cdot|b)$. Wir können bei $P_Y(b) = 0$ die bedingte Verteilung $P_{X|Y}(\cdot|b)$ also frei wählen.

Erwartungswert

- Sei X eine Zufallsvariable und gegeben sei eine Funktion $f\colon \mathcal{X} \to \mathbf{R}$. Der *Erwartungswert* von $f(X)$ ist definiert als

$$\mathrm{E}[f(X)] := \sum_{a \in \operatorname{supp} P_X} P_X(a) f(a). \tag{11}$$

- Falls $\mathcal{X} \subset \mathbf{R}$, dann ist auch $\mathrm{E}(X)$ definiert und heißt *der Erwartungswert von* X.

Aufgabe 4.

1. Kreieren Sie einen Ordner `+vlk`. Implementieren Sie `function Efx = vlk.expect(Px,funval)` in dem File `+vlk/expect.m`. Die Eingaben der Funktion sind ein Spaltenvektor mit Wahrscheinlichkeiten und ein Spaltenvektor mit Funktionswerten.

Informationsdivergenz

Die *Informationsdivergenz* zweier Verteilungen P_X und P_Y mit $\mathcal{X} = \mathcal{Y}$ ist definiert als

$$\mathrm{D}(P_X \| P_Y) = \sum_{a \in \operatorname{supp} P_X} P_X(a) \log_2 \frac{P_X(a)}{P_Y(a)} = E\left[\log_2 \frac{P_X(X)}{P_Y(X)}\right] \tag{12}$$

Informationsdivergenz: Aufgaben

Aufgabe 5.

1. Zeigen Sie:

$$0 \overset{(a)}{\leq} \mathrm{D}(P_X \| P_Y). \tag{13}$$

   Hinweis: Benutzen Sie $\log_2 x \leq (x-1)\log_2 e$.

2. Wann gilt Gleichheit in (a)?
3. Geben Sie ein Beispiel für $\mathrm{D}(P_X \| P_Y) \neq \mathrm{D}(P_Y \| P_X)$.
4. Implementieren Sie `function id = vlk.idiv(Px,Py)`.
   Testen Sie ihre Implementierung mit
   - `Px = [1/2 1/2]'` und `Py = [1/2 1/2]'`.
   - `Px = [1/2 1/2]'` und `Py = [1 0]'`.
   - `Px = [1 0]'` und `Py = [1/2 1/2]'`.

Entropie

Die *Entropie* einer Zufallsvariablen X ist definiert als

$$\mathrm{H}(P_X) := \sum_{a \in \operatorname{supp} P_X} P_X(a)[-\log_2 P_X(a)] = \mathrm{E}[-\log_2 P_X(X)]. \tag{14}$$

Entropie: Aufgaben

Aufgabe 6.

1. Sei P_X eine beliebige Verteilung auf $\mathcal{X}$ und P_U die Gleichverteilung auf $\mathcal{X}$. Zeigen Sie

$$\mathrm{H}(P_X) = \log_2 |\mathcal{X}| - \mathrm{D}(P_X \| P_U). \tag{15}$$

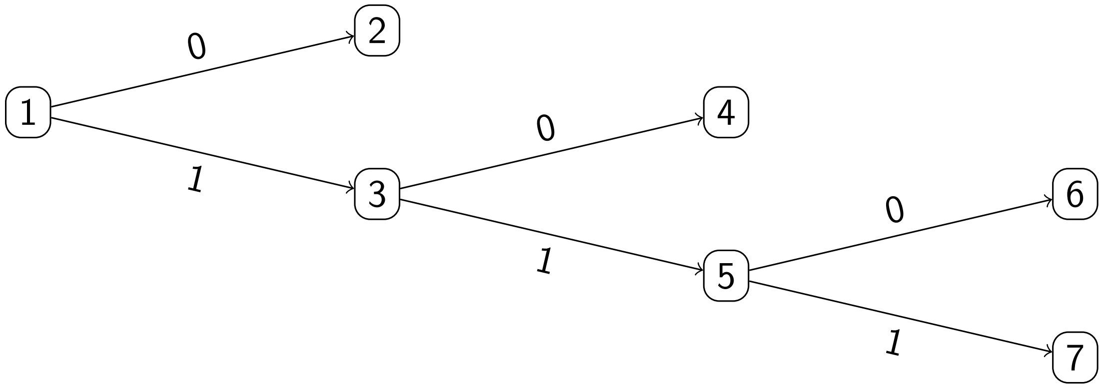
2. Zeigen Sie

$$0 \overset{(a)}{\leq} \mathrm{H}(P_X) \overset{(b)}{\leq} \log_2 |\mathcal{X}|. \tag{16}$$

3. Wann gilt Gleichheit in (a) und wann gilt Gleichheit in (b)?
4. Implementieren Sie `function en = vlk.entr(Px)`. Testen Sie ihre Implementierung mit
   - `Px = [1/2 1/2]'.`
   - `Px = [0 1]'.`
   - `Px = [0.11 0.89]'.`

Wurzelbäume mit Wahrscheinlichkeiten

Wurzelbäume: Beispiel



Lehrstuhl für Nachrichtentechnik Technische Universität München

Wurzelbäume: Knoten

- Ein Knoten ist durch eine *gerichtete Kante* mit seinem Kindknoten verbunden.
- Ein Knoten ohne Kindknoten heißt *Blatt*.
- Ein Knoten mit Kindknoten heißt *innerer Knoten*.
- Alle Knoten bis auf einen haben genau einen *Elternknoten*.
- Der Knoten ohne Elternknoten heißt die *Wurzel*.
- Die *Tiefe* eines Knotens ist die Anzahl Kanten auf dem Weg von der Wurzel zum Knoten.

Knotennummerierung

Wir verwenden die folgenden Konventionen:

- Die Wurzel hat die Nummer 1.
- Die Knoten sind aufsteigend nummeriert nach Tiefe.
- Bei Knoten gleicher Tiefe kommen Blätter vor inneren Knoten.

Aufgabe 7. Wenn ein binärer Wurzelbaum n Blätter hat, wieviele innere Knoten hat er dann? Wieviele Knoten hat er insgesamt?

Pfade und Pfadlängen

- Der Weg von der Wurzel zu einem Blatt heißt *Pfad*.
- Die Länge eines Pfades heißt *Pfadlänge*.
- Die Pfadlänge entspricht der Tiefe des zugehörigen Blattes.

Knotenmengen

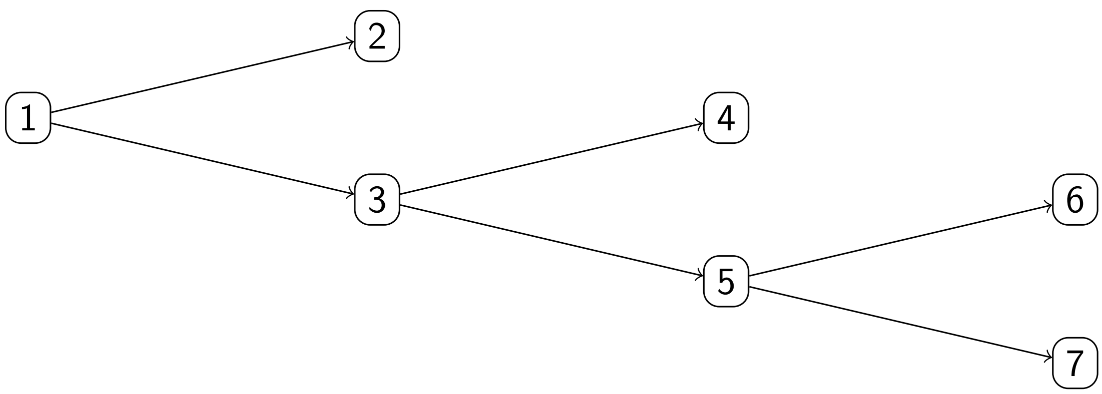
Für einen Baum definieren wir folgende Mengen:

- $\mathcal{N}$: die Menge aller Knoten.
- $\mathcal{L}$: die Menge der Blätter.
- $\mathcal{L}_i$: die Menge aller Blätter, die von Knoten i abstammen.
- $\mathcal{B} = \mathcal{N} \setminus \mathcal{L}$: die Menge aller innerer Knoten.

Lehrstuhl für Nachrichtentechnik Technische Universität München

Knotenmengen: Beispiel

- $\mathcal{N} = \{1, 2, 3, 5, 6, 7\}$
- $\mathcal{L} = \{2, 4, 6, 7\}$
- $\mathcal{L}_3 = \{4, 6, 7\}$
- $\mathcal{B} = \mathcal{N} \setminus \mathcal{L} = \{1, 3, 5\}$



Aufgaben

Aufgabe 8.

1. Kreieren Sie einen Ordner `@RootedTree` und darin ein File `RootedTree.m` mit untenstehender Klassendefinition.
2. Vervollständigen Sie den Konstruktor. Der input `ls` ist ein Spaltenvektor mit Blättertiefen in aufsteigender Reihenfolge, also `ls(1)`$\leq$`ls(2)`$\cdots$. Die Nummerierungskonventionen sollen eingehalten werden. Testen Sie mit `ls = [1 2 3 3]'`.

```
classdef RootedTree < handle
    properties
        numNodes;
        successors;
    end
    methods
        function tree = RootedTree(ls)
            tree = tree@handle;
            tree.numNodes = ...
            tree.successors = zeros(tree.numNodes,2);
    ...
    end
end
```


Blattverteilung

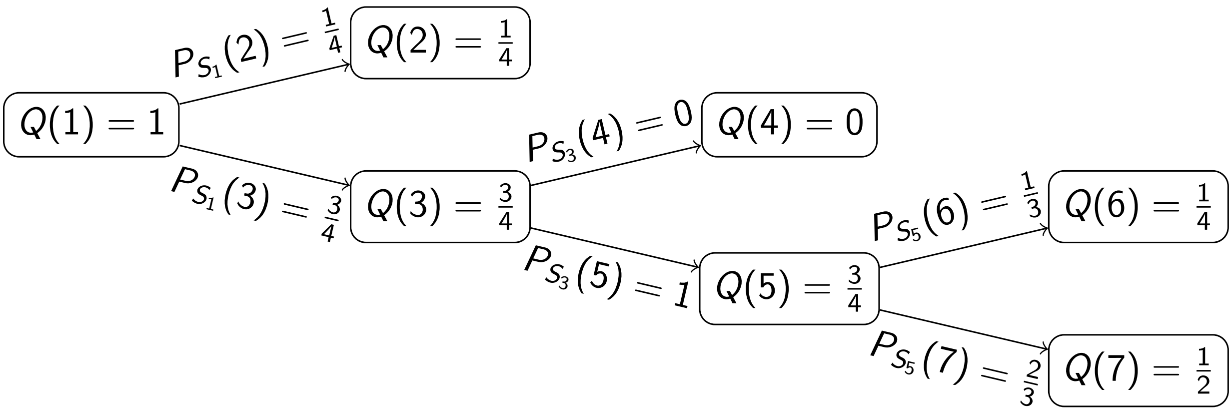
Aufgabe 9. Sei L eine Zufallsvariable mit Ergebnismenge $\mathcal{L}$ und Verteilung Q.

1. Was ist die Wahrscheinlichkeit, dass der Pfad, der beim Blatt L endet, durch den Knoten i geht? Wir bezeichnen diese Wahrscheinlichkeit mit $Q(i)$.
2. Sei t die minimale Blatttiefe und $s \leq t$. Zeigen Sie: Q definiert eine Verteilung über Knoten der Tiefe s.
3. Sei $\mathcal{S}_i$ die Menge der Kindknoten von Knoten i. Gegeben, dass der Pfad zu L durch i geht, mit welcher Wahrscheinlichkeit geht er durch Knoten $a \in \mathcal{S}_i$? Wir bezeichnen diese Verzweigungsverteilung mit P_{S_i} und die zugehörige Zufallsvariable mit S_i.

Beispiel

Blattverteilung: $Q(2)=\frac{1}{4},\ Q(4)=0,\ Q(6)=\frac{1}{4},\ Q(7)=\frac{1}{2}$.

- $Q(3)=\sum_{i\in\mathcal{L}_3} Q(i) = 0+\frac{1}{4}+\frac{1}{2}=\frac{3}{4}$
- $\mathcal{S}_1=\{2,3\}$
- $P_1(3)=\frac{Q(2)}{Q(1)}$



Lehrstuhl für Nachrichtentechnik Technische Universität München

Blattverteilung

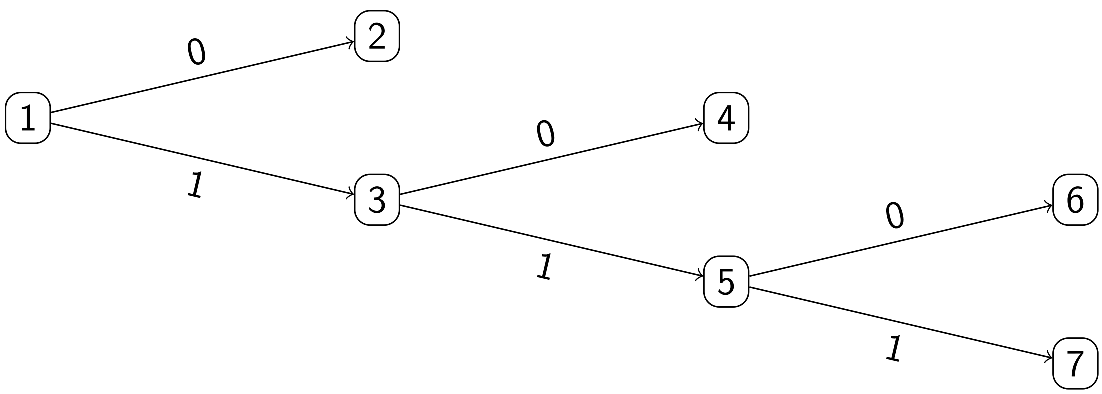
Aufgabe 10. Ergänzen Sie die `properties` von `RootedTree` um `Q` und implementieren Sie `function RootedTree.setPL(tree,Pl)`, welche `Q` berechnet. `Pl` ist ein Spaltenvektor mit Blattwahrscheinlichkeiten.

Kantenlabel

- Jeder innere Knoten hat 2 Kinder.
- Wir geben den Kanten Label $x \in \mathcal{X} = \{0, 1\}$.
- Wir definieren $x(i)$ als das Label der Kante, die zu Knoten i führt.
- Die Label der Pfade durch den Baum bilden eine Menge $\mathcal{W}$ von Wörtern mit Buchstaben in $\mathcal{X}$.
- Konvention: Falls das 0-Kind eines Knotens die Nummer n hat, dann hat das 1-Kind desselben Knotens die Nummer $n + 1$.

Lehrstuhl für Nachrichtentechnik Technische Universität München

Kantenlabel: Beispiel



Kantenlabel: Aufgabe

Aufgabe 11. Ergänzen Sie die `properties` von `RootedTree` um `W`. Implementieren Sie `function RootedTree.setW(tree)`. `W` ist ein cell array mit n Zeilen. Die Zeilen enthalten die Label der Pfade als Vektoren.

Verzweigungsverteilungen

- Wir können durch eine Labelverteilung P_X nun Verzweigungsverteilungen definieren:

$$j \in \mathcal{B}, i \in \mathcal{S}_j \colon P_{S_j}(i) = P_X[x(i)]. \tag{17}$$

- P_X definiert auch eine Verteilung über die Wörter im Wörterbuch, nämlich

$$P_X^{\mathcal{W}}(a) = P_X(a_1) \cdots P_X(a_{\ell(a)}). \tag{18}$$

Aufgabe 12. Implementieren Sie `function RootedTree.setPS(tree,Px)`. Diese Funktion berechnet `Q` aus der Labelverteilung `Px`.

LANSIT[1]

- Sei f eine Funktion, die jedem Knoten $i \in \mathcal{N}$ einen reellen Wert $f(i)$ zuordnet.
- Für jeden Knoten $i \in \mathcal{N} \setminus 1$ definieren wir $\Delta f(i) := f(i) - f(\text{Elternknoten von } i)$.
- Sei S_i eine Zufallsvariable mit Ergebnismenge $\mathcal{S}_i$ und Verteilung P_{S_i}.

Proposition 1 (LANSIT)

$$\mathsf{E}[f(L)] - f(1) = \sum_{j \in \mathcal{B}} Q(j)\, \mathsf{E}[\Delta f(S_j)] \tag{19}$$

[1] Leaf-Average Node-Sum Interchange Theorem, siehe [1].

LANSIT: Beweis

- Betrachte einen Baum mit Knotenmenge $\mathcal{N}$.
- Sei $\mathcal{S}_j \subseteq \mathcal{L}$ eine Menge von Blättern mit einem gemeinsamen Elternknoten j.

$$\sum_{i \in \mathcal{S}_j} Q(i) f(i) = \sum_{i \in \mathcal{S}_j} Q(j) P_{S_j}(i) \Big[f(i) - f(j) + f(j) \Big] \tag{20}$$

$$= Q(j) f(j) \underbrace{\Big[\sum_{i \in \mathcal{S}_j} P_{S_j}(i) \Big]}_{=1} + Q(j) \sum_{i \in \mathcal{S}_j} P_{S_j}(i) \Delta f(i) \tag{21}$$

$$= Q(j) f(j) + Q(j)\, \mathsf{E}[\Delta f(S_j)] \tag{22}$$

- $\mathcal{N} \leftarrow \mathcal{N} \setminus \mathcal{S}_j$ ist ein neuer Baum mit weniger Blättern. Die Knotenwahrscheinlichkeiten sind weiterhin definiert durch Q.
- Wiederhole die Prozedur bis j der Wurzelknoten 1 ist. Dann $Q(j = 1) = 1$ und $Q(j = 1) f(j = 1) = f(1)$.

$\square$

LANSIT: Aufgaben[2]

Aufgabe 13. Benutzen Sie das LANSIT um die folgenden Identitäten zu zeigen.

1. **Path Length Lemma.** Funktion $\ell(i) :=$ Knotentiefe von i.

$$\mathrm{E}[\ell(L)] = \sum_{i\in\mathcal{B}} Q(i). \tag{23}$$

2. **Leaf Entropy Lemma.** Funktion $f(i) = -\log_2 Q(i)$.

$$\mathrm{H}(P_L) = \sum_{i\in\mathcal{B}} Q(i)\,\mathrm{H}(P_{S_i}). \tag{24}$$

3. **Leaf Divergence Lemma.** Gegeben seien weitere Knotenwahrscheinlichkeiten Q' mit der zugehörigen Blattverteilung $P_{L'}$. Funktion $f(i) = \log_2 \frac{Q(i)}{Q'(i)}$. Dann

$$\mathrm{D}(P_L\|P_{L'}) = \sum_{i\in\mathcal{B}} Q(i)\,\mathrm{D}(P_{S_i}\|P_{S'_i}). \tag{25}$$

[2] Siehe [2].

LANSIT: Aufgaben

Aufgabe 14.

1. Implementieren Sie `function y = RootedTree.NS(tree,dfun,param)` wobei `dfun` ein handle für Funktionen der Form `function d = dfun(i,j,psji,param)` ist. `i` ist ein Elternknoten, `j` ist ein Kindknoten von `i` und `psji` ist $P_{S_j}(i)$. Das Eingabeargument `param` kann der Benutzer frei wählen.
2. Verifizieren Sie die Lemmata für Path Length, Leaf Entropy und Leaf Divergence, indem Sie für Beispielbäume jeweils die linke Seite der Gleichung mittels der Blattverteilung berechnen und die rechte Seite mittels der `RootedTree.NS` Funktion.

Nachtrag: Vollständige binäre Bäume

Ein binärer Baum heißt *vollständig*, falls jeder Knoten entweder zwei oder keinen Kindknoten hat.

Nachtrag: Zulässige Pfadlängen

Seien $\ell_1, \ell_2, \dots, \ell_n$ Pfadlängen. Wir wollen einen Test entwickeln, mit dem wir prüfen können, ob für diese Pfadlängen ein vollständiger beziehungsweise ein unvollständiger binärer Baum existiert. Sei $\ell_{\max} = \max_i \ell_i$. Betrachte einen vollständigen Baum, bei dem alle Blätter die Tiefe $\ell_{\max}$ haben.

- Der Baum hat $2^{\ell_{\max}}$ Knoten der Tiefe $\ell_{\max}$.
- Ein Knoten der Tiefe $\ell \leq \ell_{\max}$ hat $2^{\ell_{\max}-\ell}$ nachfolgende Knoten der Tiefe $\ell_{\max}$.
- Wir berechnen

$$\sum_{i=1}^{n} 2^{-\ell_i} = 2^{-\ell_{\max}} \underbrace{\sum_{i=1}^{n} 2^{\ell_{\max}-\ell_i}}_{(\star)} \tag{26}$$

  Die Summe $(\star)$ ist gleich $2^{\ell_{\max}}$, falls die ℓ_i Pfadlängen eines vollständigen Baumes sind.

Nachtrag: Kraft-Ungleichung

Wir haben nun folgenden Test. Seien $\ell_1, \ell_2, \ldots, \ell_n$ positive ganze Zahlen.

- $\sum_{i=1}^{n} 2^{-\ell_i} = 1 \Rightarrow$ es existiert ein vollständiger binärer Baum mit den Pfadlängen ℓ_i.
- $\sum_{i=1}^{n} 2^{-\ell_i} < 1 \Rightarrow$ es existiert ein unvollständiger binärer Baum mit den Pfadlängen ℓ_i.
- $\sum_{i=1}^{n} 2^{-\ell_i} > 1 \Rightarrow$ es existiert weder ein vollständiger noch ein unvollständiger binärer Baum mit den Pfadlängen ℓ_i.

Für eine nicht-binäre Labelmenge $\mathcal{X}$ mit $|\mathcal{X}| = D > 2$ testen wir $\sum_{i=1}^{n} D^{-\ell_i}$.

Verteilungsanpassung

Gedächtnislose Quelle

- Eine *gedächtnislose Quelle* $\boxed{P_X}$ erzeugt Symbole $X_1X_2X_3\cdots$, die statistisch unabhängig sind und alle die Verteilung P_X haben.
- Sei $n > 0$. Wir schreiben $X^n := X_1X_2\cdots X_n$. Es gilt

$$\Pr(X^n = a^n) = P_X(a_1)P_X(a_2)\cdots P_X(a_n) \tag{27}$$

für alle $a^n \in \mathcal{X}^n = \mathcal{X} \times \cdots \times \mathcal{X}$.

LANSIT: Aufgaben

Aufgabe 15. (freiwillig) Seien X^n und Y^n Zufallsvektoren. Zeigen Sie:

1. Entropy-Kettenregel:

$$H(P_{X^n}) = \sum_{i=1}^{n} H(P_{X_i|X^{i-1}}|P_{X^{i-1}}) \tag{28}$$

2. Informationsdivergenz-Kettenregel:

$$D(P_{X^n}\|P_{Y^n}) = \sum_{i=1}^{n} D(P_{X_i|X^{i-1}}\|P_{Y_i|X^{i-1}}|P_{X^{i-1}}) \tag{29}$$

Verteilungsanpasser

Ein Verteilungsanpasser transformiert eine Eingangssequenz in eine Ausgangssequenz:

$$\boxed{P_X} \to X_1 X_2 \cdots \to \boxed{\text{Verteilungsanpasser}} \to Y_1 Y_2 \cdots$$

Die Ausgabe ist eine Sequenz von Symbolen aus $\mathcal{Z}$ deren Verteilung einer Zielverteilung P_Z ähnlich sein soll.

Wörterbuch und Kode

- Vollständiges Wörterbuch:
  - Die Buchstaben der Eingabe sind in $\mathcal{X}$
  - Die Label der Pfade eines vollständigen Baumes mit Labeln in $\mathcal{X}$ nennen wir *Wörterbuch* $\mathcal{W}$.
  - Beispiel: $\mathcal{X} = \{a, b, c\}$, $\mathcal{W} = \{a, b, ca, cb, cc\}$.
- Vollständiger Kode:
  - Die Buchstaben der Ausgabe sind in $\mathcal{Z}$
  - Die Label der Pfade eines vollständigen Baumes mit Labeln in $\mathcal{Z}$ nennen wir *Kode*.
  - Beispiel: $\mathcal{Z} = \{0, 1\}$, $\mathcal{C} = \{0, 100, 101, 110, 111\}$. $\mathcal{C}$ ist ein binärer Kode.

Parsen der Eingabe

- Wir parsen die Eingabe mit einem vollständigen Wörterbuch $\mathcal{W}$ mit Buchstaben in $\mathcal{X}$.
- Dies erzeugt Wörter W mit der Verteilung $P_X^{\mathcal{W}}$, die gegeben ist durch

$$P_X^{\mathcal{W}}(w) = P_X(w_1)P_X(w_2)\cdots P_X(w_{\ell(w)}), \quad \text{für jedes } w \in \mathcal{W}. \tag{30}$$

Aufgabe 16. Zeigen Sie mit Hilfe des LANSIT:

$$\mathrm{H}(P_X^{\mathcal{W}}) = \mathrm{E}[\ell(W)]\,\mathrm{H}(P_X) \tag{31}$$

Ausgabe von Kodewörtern

Wir wählen als Ausgabe Kodewörter aus einem vollständigen Kode $\mathcal{C}$ mit Buchstaben aus $\mathcal{Z}$. Der Verteilungsanpasser bildet die geparsten Wörter auf Kodewörter ab mit einer injektiven Funktion $f\colon \mathcal{W} \to \mathcal{C}$. Sei $Y = f(W)$ das ausgegebene Kodewort.

- Die erwartete Kodewortlänge ist $\mathrm{E}[\ell(Y)]$.
- Kodewort-Zielverteilung:

$$P_Z^{\mathcal{C}}(y) = P_Z(y_1)P_Z(y_2)\ldots P_Z(y_{\ell(y)}), \quad \text{für jedes } y \in \mathcal{C}. \tag{32}$$

- Die tatsächlich erzeugte Verteilung von Y ist

$$P_Y(y) = \begin{cases} P_X^{\mathcal{W}}[f^{-1}(y)] & \text{falls } \exists w\colon f(w) = y,\\ 0 & \text{sonst.} \end{cases} \tag{33}$$

Kode variabler Länge: Aufgabe

Aufgabe 17. Gegeben ist die Zielverteilung $P_Z(0) = P_Z(1) = \frac{1}{2}$ und der Kode $\mathcal{C} = \{0, 10, 11\}$. Die erzeugte Verteilung sei $P_Y(0) = P_Y(10) = P_Y(11) = \frac{1}{3}$.

1. Berechnen Sie die Kodewort-Zielverteilung.
2. Berechnen Sie die erwartete Kodewortlänge.

Rate

- Die *Rate* R des Verteilungsanpassers ist (erwartete Information)/(ewartete Ausgabelänge), das heißt

$$R := \frac{\mathrm{H}(P_X^{\mathcal{W}})]}{\mathrm{E}[\ell(Y)]}. \tag{34}$$

Die Funktion $f: \mathcal{W} \to \mathcal{C}$

Wir indizieren das Wörterbuch $\mathcal{W} = \{w_1, w_2, \dots, w_n\}$ und den Kode $\mathcal{C} = \{c_1, c_2, \dots, c_m\}$ mit $m \geq n$ so dass

$$P_X^{\mathcal{W}}(w_1) \geq P_X^{\mathcal{W}}(w_2) \geq \cdots \geq P_X^{\mathcal{W}}(w_n), \tag{35}$$

$$P_Z^{\mathcal{C}}(c_1) \geq P_Z^{\mathcal{C}}(c_2) \geq \cdots \geq P_Z^{\mathcal{W}}(c_m). \tag{36}$$

Wir definieren dann f durch $f: w_i \mapsto c_i$, wir bilden also Wörter kleiner Wahrscheinlichkeit ab auf Kodewörter mit kleiner Kodewort-Zielwahrscheinlichkeit.

Datenkompression

Das Designziel

Für eine gegebene Datenquelle $\boxed{P_X}$ wollen wir die Rate

$$R = \frac{\mathrm{H}(P_X^{\mathcal{W}})]}{\mathrm{E}[\ell(Y)]} \tag{37}$$

maximieren für eine binäre Ausgabe.

Informationsdivergenz

Wir wählen als Zielverteilung die Gleichverteilung P_U auf $\mathcal{Z} = \{0, 1\}$ und entwickeln die Informationsdivergenz.

$$\mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}}) \overset{(a)}{=} \sum_{j\in\mathcal{B}} Q(j)\, \mathrm{D}(P_{S_j} \| P_U) \tag{38}$$

$$= \sum_{j\in\mathcal{B}} Q(j) \left[\sum_{a\in\mathcal{S}_j} P_{S_j}(a) \log_2 \frac{P_{S_j}(a)}{\frac{1}{2}} \right] \tag{39}$$

$$= \sum_{j\in\mathcal{B}} Q(j) \left[1 - \mathrm{H}(P_{S_j}) \right] \tag{40}$$

$$\overset{(b)}{=} \mathrm{E}[\ell(Y)] - \mathrm{H}(P_X^{\mathcal{W}}). \tag{41}$$

In (a) haben wir das Leaf Divergence Lemma benutzt und in (b) das Path Length Lemma und das Leaf Entropy Lemma. Q sind Knotenwahrscheinlichkeiten für den Baum von Kode $\mathcal{C}$ mit Blattverteilung $P_X^{\mathcal{W}}$.

Grenzen

- Übertragunsrate:

$$\frac{\mathrm{H}(P_X^{\mathcal{W}})}{\mathrm{E}[\ell(Y)]} = 1 - \frac{\mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}})}{\mathrm{E}[\ell(Y)]} \leq 1. \tag{42}$$

- Erwartete Kodewortlänge:

$$\mathrm{E}[\ell(Y)] = \mathrm{H}(P_X^{\mathcal{W}}) + \mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}}) \geq \mathrm{H}(P_X^{\mathcal{W}}). \tag{43}$$

Wir können entweder die Rate maximieren oder die Informationsdivergenz pro bit minimieren über alle Wörterbücher $\mathcal{W}$ und Kodes $\mathcal{C}$. *Ungelöstes Problem!*
Beachte: Wir erreichen die maximale Rate, falls $\mathrm{D}(P_X^{\mathcal{W}} \| P_U^{\mathcal{C}}) = 0$, die von uns als Zielverteilung gewählte Gleichverteilung P_U maximiert also tatsächlich die Rate!

Huffman Coding

Für Huffman Coding fixieren wir $\mathcal{W} = \mathcal{X}$. Die Grenzen sind nun

- Übertragunsrate:

$$\frac{\mathrm{H}(P_X)}{\mathrm{E}[\ell(Y)]} = 1 - \frac{\mathrm{D}(P_X \| P_U^{\mathcal{C}})}{\mathrm{E}[\ell(Y)]} \leq 1. \tag{44}$$

- Erwartete Kodewortlänge:

$$\mathrm{E}[\ell(Y)] = \mathrm{H}(P_X) + \mathrm{D}(P_X \| P_U^{\mathcal{C}}) \geq \mathrm{H}(P_X). \tag{45}$$

Um die Rate zu maximieren, können wir nun entweder die erwartete Länge oder die Informationsdivergenz minimieren.

Huffman Coding[3]

- Das verbleibende Problem: Wähle $\mathcal{C}$ so, dass die erwartete Kodewortlänge

$$\mathrm{E}[\ell(Y)] = \sum_{x \in \mathcal{X}} P_X(x)\ell[f(x)] \tag{46}$$

minimiert wird.

[3]Siehe [3].

Huffman Coding: Aufgabe

Um die Notation zu vereinfachen, schreiben wir die Wahrscheinlichkeiten als $p_1, p_2, \ldots, p_n$ und die Kodewortlängen als $\ell_1, \ell_2, \ldots, \ell_n$.

Aufgabe 18. Zeigen Sie folgende Eigenschaften eines optimalen Kodes.

1. Falls $p_i < p_j$ dann $\ell_i \geq \ell_j$.
2. Ein optimaler Kode ist vollständig.
3. Nehmen Sie an $p_1 \geq p_2 \geq \cdots \geq p_{n-1} \geq p_n$. Es gibt dann einen optimalen code mit

$$\ell_n = \ell_{n-1} = \max_i \ell_i, \tag{47}$$

das heißt die Blätter mit den Längen ℓ_n, ℓ_{n-1} sind Geschwister mit einem gemeinsamen Elternknoten.

Huffman Coding: der Algorithmus

Die Pfadlängen seien optimal und erfüllen Eigenschaft (47). Sei L eine Zufallsvariable auf der Menge der Blätter. Die Längen $\ell_1, \ell_2, \ldots, \ell_{n-2}, \ell_{n-1} - 1$ sind Pfadlängen eines neuen Baumes mit dem Elternknoten der Blätter mit Längen ℓ_n, ℓ_{n-1} als Blatt. Dieses neue Blatt hat Wahrscheinlichkeit $p_n + p_{n-1}$. Der neue Baum hat $n-1$ Blätter. Sei L' eine Zufallsvariable auf der Menge der Blätter des neuen Baumes.
Wegen des Path Length Lemmas haben wir

$$\mathsf{E}[\ell(L)] = \mathsf{E}[\ell(L')] + p_n + p_{n-1}. \tag{48}$$

Da die Pfadlängen des Baumes mit n Blättern optimal sind, müssen die Pfadlängen des neuen Baumes mit $n-1$ Blättern ebenfalls optimal sein, d.h. $\mathsf{E}[\ell(L')]$ minimieren. Wir können den optimalen Baum also konstruieren, indem wir rekursiv die Blätter kleinster Wahrscheinlichkeit zu einem Elternknoten zusammenfassen.

Huffman Coding: Implementierung

Aufgabe 19.

1. Implementieren Sie `function l = vlk.huffman(p)`. Die Eingabe `p` ist ein Verteilungsvektor und die Ausgabe `l` ist ein Vektor mit optimalen Kodewortlängen.
2. Konstruieren Sie den zugehörigen Baum, indem Sie mittels der Kodewortlängen einen gewurzelten Baum erzeugen.

Verteilungsanpassung für rauschfreie Kanäle

Problemstellung

- Am Eingang haben wir gleichverteilte Informationsbits $\boxed{P_U}$.
- Die Ausgangssymbole $a \in \mathcal{Z}$ haben unterschiedliche Längen $v(a)$. Die Werte von $v(a)$ sind reell und positiv, müssen aber nicht ganzzahlig sein.
- Wir wollen mit maximaler Rate übertragen.
- Die Längenfunktion v definiert einen *diskreten rauschfreien Kanal*.

Zielverteilung

Definiere P_Z als

$$P_Z(a) = 2^{-Cv(a)}, \qquad \text{für jedes } a \in \mathcal{Z} \tag{49}$$

wobei C so gewählt ist, dass

$$\sum_{a \in \mathcal{Z}} 2^{-Cv(a)} = 1. \tag{50}$$

Aufgabe 20. Implementieren Sie `function C = vlk.capacityDNC(v)`. Die Eingabe `v` ist ein Spaltenvektor mit positiven Längen. *Hinweis:* Benutzen Sie Bisektion.

Zielverteilung

Wir entwickeln die Informationsdivergenz:

$$\begin{aligned} \mathrm{D}(P_Y \| P_Z) &= \sum_{a \in \mathcal{Z}} P_Y(a) \log_2 \frac{P_Y(a)}{2^{-Cv(a)}} &(51)\\ &= C \sum_{a \in \mathcal{Z}} P_Y(a) v(a) - \mathrm{H}(P_Y) &(52)\\ &= C\, \mathrm{E}[v(Y)] - \mathrm{H}(P_Y). &(53) \end{aligned}$$

Wir haben also

$$R = \frac{\mathrm{H}(P_Y)}{\mathrm{E}[v(Y)]} = C - \frac{\mathrm{D}(P_Y \| P_Z)}{\mathrm{E}[v(Y)]}. \tag{54}$$

C ist also die maximale Rate (auch *Kapazität* des rauschfreien Kanals v genannt) und wird erreicht, falls $P_Y = P_Z$. Unsere Zielverteilung ist also P_Z.

Verteilungsanpassung

Wir entwickeln die Informationsdivergenz.

$$\mathrm{D}(P_U^{\mathcal{W}} \| P_Z^{\mathcal{C}}) \overset{(a)}{=} \sum_{i\in\mathcal{B}} Q(i)\,\mathrm{D}(P_{S_i} \| P_Z) \tag{55}$$

$$\overset{(b)}{=} \sum_{i\in\mathcal{B}} Q(i)\{C\,\mathrm{E}[\Delta v(S_i)] - \mathrm{H}(P_{S_i})\} \tag{56}$$

$$\overset{(c)}{=} C\,\mathrm{E}[v(Y)] - \mathrm{H}(P_U^{\mathcal{W}}). \tag{57}$$

Wir benutzen in (a) das Leaf Divergence Lemma, in (b) benutzen wir (53) und in (c) das LANSIT und das Leaf Entropy Lemma. Wir haben also

$$R = \frac{\mathrm{H}(P_U^{\mathcal{W}})}{\mathrm{E}[v(Y)]} = C - \frac{\mathrm{D}(P_U^{\mathcal{W}} \| P_Z^{\mathcal{C}})}{\mathrm{E}[v(Y)]}. \tag{58}$$

Die Rate zu maximieren bzw die Informationsdivergenz pro erwartete Kodewortlänge zu minimieren über das Wörtebuch $\mathcal{W}$ *und* den Kode $\mathcal{C}$ ist ein ungelöstes Problem.

Fixierter Kode

Wir fixieren den Kode. Das verbleibende Problem ist

$$\frac{\mathrm{D}(P_U^{\mathcal{W}} \| P_Z)}{\mathrm{E}[v(Y)]} \tag{59}$$

zu minimieren über das Wörterbuch $\mathcal{W}$.

Äquivalentes Problem

Wir nehmen zunächst an, wir würden das Minimum δ kennen, also

$$\frac{\mathrm{D}(P_U^{\mathcal{W}} \| P_Z)}{\mathrm{E}[v(Y)]} \geq \delta \tag{60}$$

mit Gleichheit, falls $\mathcal{W}$ optimal ist. Äquivalent sind

$$\begin{aligned} \mathrm{D}(P_U^{\mathcal{W}} \| P_Z) &\geq \delta\, \mathrm{E}[v(Y)] && (61)\\ \Leftrightarrow \mathrm{D}(P_U^{\mathcal{W}} \| P_Z) - \delta\, \mathrm{E}[v(Y)] &\geq 0 && (62)\\ \Leftrightarrow \sum_{a \in \mathcal{W}} P_U^{\mathcal{W}}(a) \left[\log_2 \frac{P_U^{\mathcal{W}}(a)}{P_Z(a)} - \delta v(a)\right] &\geq 0 && (63)\\ \Leftrightarrow \sum_{a \in \mathcal{W}} P_U^{\mathcal{W}}(a) \log_2 \frac{P_U^{\mathcal{W}}(a)}{P_Z(a) 2^{\delta v(a)}} &\geq 0 && (64)\\ \Leftrightarrow \mathrm{D}(P_U^{\mathcal{W}} \| P_Z \circ 2^{\delta v}) &\geq 0. && (65) \end{aligned}$$

Geometric Huffman Coding[4]

Aus (65) wissen wir, dass wir $\mathrm{D}(P_U^{\mathcal{W}} \| T)$ minimieren müssen für $T = P_Z \circ 2^{\delta v}$. T ist eine nicht-negative Funktion auf $\mathcal{Z}$, aber nicht unbedingt eine Verteilung. *Geometric Huffman Coding* berechnet das optimale Wörterbuch $\mathcal{W}$. Der Algorithmus ist ähnlich zu Huffman Coding.

- Seien $T(a) \geq T(b)$ die kleinsten Funktionswerte. Wir unterscheiden zwei Fälle.
  1. $T(a) \geq 4T(b)$. Wir entfernen das Blatt b.
  2. $T(a) < 4T(b)$. Wir vereinigen die Blätter a und b in einem Elternknoten e. Wir setzen $T(e) = 2\sqrt{T(a)T(b)}$.

Wir wiederholen diese Prozedur bis wir beim Elternknoten angekommen sind. Der konstruierte Baum liefert das optimale Wörterbuch $\mathcal{W}$.

[4] Beweis für Optimalität in [4],[5, Section 3.2.3]. Siehe auch [6].

Bestimmung von δ[5]

Mit dem folgenden Algorithmus können wir δ und das optimale Wörterbuch $\mathcal{W}$ bestimmen.

Normalized Geometric Huffman Coding

$\hat{\mathcal{W}} \leftarrow \underset{\mathcal{W}}{\operatorname{argmin}} \operatorname{D}(P_U^{\mathcal{W}} \| P_Z)$

repeat

$\quad \hat{\delta} \leftarrow \frac{\operatorname{D}(P_U^{\hat{\mathcal{W}}} \| P_Z)}{\operatorname{E}[v(Y)]}$

$\quad \hat{\mathcal{W}} \leftarrow \underset{\mathcal{W}}{\operatorname{argmin}} \operatorname{D}(P_U^{\mathcal{W}} \| P_Z \circ 2^{\delta v})$

until $\hat{\delta} = \frac{\operatorname{D}(P_U^{\hat{\mathcal{W}}} \| P_Z)}{\operatorname{E}[v(Y)]}$

$\delta \leftarrow \hat{\delta}, \mathcal{W} \leftarrow \hat{\mathcal{W}}$

[5] Den Beweis für die Korrektheit des Algorithmus finden Sie in [5, Section 4.1.1].

Kodierung für Rauschfreie Kanäle: Aufgabe

Aufgabe 21. Implementieren Sie Normalized Geometric Huffman Coding in `function l = vlk.nghuffman(w)`. Sie können Geometric Huffman Coding verwenden. Es ist in `function l = vlk.ghuffman(p)` implementiert.

Erweiterungen

- Datenkompression mit fixiertem Kode: Tunstall Coding [7],[8, Section 2.3.4].
- Verteilungsanpassung mit fixiertem Wörterbuch: Siehe [9].

Literatur I

[1] R. A. Rueppel and J. L. Massey, "Leaf-average node-sum interchanges in rooted trees with applications," in *Communications and Cryptography: Two sides of One Tapestry*, R. E. Blahut, D. J. Costello Jr., U. Maurer, and T. Mittelholzer, Eds. Kluwer Academic Publishers, 1994.

[2] G. Böcherer and R. A. Amjad, "Informational divergence and entropy rate on rooted trees with probabilities," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2014. [Online]. Available: http://arxiv.org/abs/1310.2882

[3] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[4] G. Böcherer and R. Mathar, "Matching dyadic distributions to channels," in *Proc. Data Compression Conf.*, 2011, pp. 23–32.

[5] G. Böcherer, "Capacity-achieving probabilistic shaping for noisy and noiseless channels," Ph.D. dissertation, RWTH Aachen University, 2012. [Online]. Available: http://www.georg-boecherer.de/capacityAchievingShaping.pdf

Literatur II

[6] ——, "Geometric Huffman coding," http://www.georg-boecherer.de/ghc, Dec. 2010.

[7] B. Tunstall, "Synthesis of noiseless compression codes," Ph.D. dissertation, 1967.

[8] J. L. Massey, "Applied digital information theory I," lecture notes, ETH Zurich. [Online]. Available: http://www.isiweb.ee.ethz.ch/archive/massey_scr/adit1.pdf

[9] R. A. Amjad and G. Böcherer, "Fixed-to-variable length distribution matching," in *IEEE Int. Symp. Inf. Theory (ISIT)*, 2013. [Online]. Available: http://arxiv.org/abs/1302.0019