

Huffman Code

Autor: Jochen Ziegenbalg

Email: ziegenbalg.edu@gmail.com

Internet: <https://jochen-ziegenbalg.github.io/materialien/>

■ Basics

■ Constructing the Huffman Code

Remarks on the implementation: The construction of the Huffman code is based on a given alphabet with frequencies; more precisely on a list of pairs {frequency, letter} - see for instance the list FTGerman above. (Having the frequency value in the first and the letter in the second position is only a matter of convenience.) This list is referred to as the underlying frequency table FT - which is the only input parameter of the function HuffmanTree.

The Huffman tree is constructed as a binary root-tree. Its subtrees are of the form {value, {left child, right child}} or {value, letter}. The latter ones are the "leaves" of the Huffman tree.

The construction of the Huffman tree is done by a series of transformations on FT as follows:

- * FT is given in sorted form (sorted by the values of the subtrees).
- * The two subtrees with the lowest values (i.e. the first and the second subtree of FT are merged into another subtree, its value being the sum of the latter two.)
- * This new subtree replaces the two subtrees from which it was constructed.
- * This procedure is continued until there is only one subtree.

```
LowestValue1[T_] := First[T[[1]]];
LowestValue2[T_] := First[T[[2]]];

HuffmanTree[FT_] :=
  (If[verbose, Print[FT]];
   If[Length[FT] == 1, FT,
    HuffmanTree[Sort[
      Prepend[Delete[FT, {{1}, {2}}], {LowestValue1[FT] +
        LowestValue2[FT], {FT[[1]], FT[[2]]}}]]])
```

```
verbose = True
```

```
True
```



```

LeftChild[HT_] := { ((First[HT])[ [2] ]) [ [1] ] };

RightChild[HT_] := { ((First[HT])[ [2] ]) [ [2] ] };

LeafP[HT_] := Not[ListQ[(HT[ [1] ]) [ [2] ] ] ] ;
(* Leaf-Property *)

Letter[Pair_] := Pair[ [2] ];

HuffmanCodeTable[HT_] :=
  If[LeafP[HT],
    {Letter[First[HT]], "1"}, Sort[HCT[HT, ""]]];

HCT[HT_, code_] :=
  Which[HT == {}, {},
    LeafP[HT], Return[{{Letter[First[HT]], code}}],
    True,
    Join[
      HCT[LeftChild[HT], StringJoin[code, "0"]],
      HCT[RightChild[HT], StringJoin[code, "1"]] ]

```

```
HuffmanCodeTable[{{0.6, "X"}}]
```

```
{X, 1}
```

```
HuffmanCodeTable[{{1, {{0.4, "U"}, {0.6, "V"}}}]
```

```
{{U, 0}, {V, 1}}
```

```
HuffmanCodeTable[HuffmanTree[FTGerman]]
```

```
{ {A, 1000}, {B, 111011}, {C, 01101}, {D, 0101},
  {E, 110}, {F, 011110}, {G, 01000}, {H, 0000}, {I, 1111},
  {J, 111010001}, {K, 011111}, {L, 11100}, {M, 01110},
  {N, 001}, {O, 01001}, {P, 11101001}, {Q, 11101000010},
  {R, 1011}, {S, 1010}, {T, 1001}, {U, 0001}, {V, 1110101},
  {W, 011001}, {X, 11101000011}, {Y, 1110100000}, {Z, 011000}

```

■ Some further utility functions

FTGerman

```
{ {0.012, Q}, {0.023, X}, {0.023, Y}, {0.222, J},
  {0.655, P}, {1.041, V}, {1.299, Z}, {1.346, W},
  {1.463, F}, {1.626, K}, {2.527, B}, {2.633, G},
  {2.644, O}, {2.832, C}, {2.949, M}, {3.3, L}, {4.517, H},
  {4.727, U}, {5.324, D}, {6.109, A}, {6.471, T},
  {6.704, S}, {7.7, R}, {8.238, I}, {9.771, N}, {15.844, E} }
```

```
Cases[FTGerman, {_, "D"}]
```

```
{{5.324, D}}
```

```
Frequency[FT_, symb_] :=
  First[First[Cases[FT, {_, symb}]]]
```

```
Frequency[FTGerman, "D"]
```

```
5.324
```

```
HuffmanCode[FT_, symb_] := Last[First[
  Cases[HuffmanCodeTable[HuffmanTree[FT]], {symb, _}]]]
```

```
HuffmanCode[FTGerman, "D"]
```

```
0101
```

```
MeanCodeLength[FT_] :=
```

```
Module[{alph, freq, codes, mean = 0},
  alph = Sort[(Transpose[FT])][[2]]];
  freq = Map[Function[x, Frequency[FT, x]], alph];
  codes = Map[Function[x, HuffmanCode[FT, x]], alph];
  (* Print[{alph, freq, codes}]; *)
  Do[mean = mean + freq[[i]] * StringLength[codes[[i]]],
    {i, 1, Length[alph]}];
  mean = 0.01 * mean;
  Return[mean]
```

```
MeanCodeLength[FTGerman]
```

```
4.12501
```

■ Some Examples

In the following examples, a word is given. The word determines the respective alphabet and the frequencies of the letters.

■ Example: ABRAKADABRA

ExABRAKADABRA =

```
Sort[{{5/11, "A"}, {2/11, "B"}, {2/11, "R"},
      {1/11, "K"}, {1/11, "D"}}] // N
```

```
{0.0909091, D}, {0.0909091, K},
{0.181818, B}, {0.181818, R}, {0.454545, A}}
```

HuffmanTree[ExABRAKADABRA]

```
{1., {{0.454545, A},
      {0.545455, {{0.181818, {{0.0909091, D}, {0.0909091, K}}},
      {0.363636, {{0.181818, B}, {0.181818, R}}}}}}}
```

% // TableForm

	0.454545	A		
1.	0.545455	0.181818	0.0909091	D
			0.0909091	K
		0.363636	0.181818	B
			0.181818	R

HuffmanCodeTable[HuffmanTree[ExABRAKADABRA]]

```
{A, 0}, {B, 110}, {D, 100}, {K, 101}, {R, 111}}
```

■ Example: MISSISSIPPIDAMPFER

ExMISSISSIPPIDAMPFER =

```
Sort[{{2/18, "M"}, {4/18, "I"}, {4/18, "S"},
      {3/18, "P"}, {1/18, "D"}, {1/18, "A"},
      {1/18, "F"}, {1/18, "E"}, {1/18, "R"}}] // N
```

```
{0.0555556, A}, {0.0555556, D}, {0.0555556, E},
{0.0555556, F}, {0.0555556, R}, {0.1111111, M},
{0.1666667, P}, {0.2222222, I}, {0.2222222, S}}
```

HuffmanTree[ExMISSISSIPPIDAMPFER]

```
{1., {{0.444444, {{0.222222, I}, {0.222222, S}}}, {0.555556,
  {{0.222222, {{0.111111, {{0.0555556, A}, {0.0555556, D}}},
    {0.111111, {{0.0555556, E}, {0.0555556, F}}}}}},
  {0.333333, {{0.166667, P}, {0.166667,
    {{0.0555556, R}, {0.111111, M}}}}}}}}
```

% // TableForm

	0.444444	0.222222	I		
		0.222222	S		
			0.111111	0.0555556	A
				0.0555556	D
1.		0.222222	0.111111	0.0555556	E
	0.555556			0.0555556	F
			0.166667		P
		0.333333		0.0555556	R
			0.166667	0.111111	M

HuffmanCodeTable[HuffmanTree[ExMISSISSIPPIDAMPFER]

```
{{A, 1000}, {D, 1001}, {E, 1010}, {F, 1011},
  {I, 00}, {M, 1111}, {P, 110}, {R, 1110}, {S, 01}}
```

■ Example: MISSISSIPPISCHIFF**ExMISSISSIPPISCHIFF =**

```
Sort[{{1/17, "M"}, {5/17, "I"}, {5/17, "S"}, {2/17,
  "P"}, {1/17, "C"}, {1/17, "H"}, {2/17, "F"}}] // N
{{0.0588235, C}, {0.0588235, H}, {0.0588235, M},
  {0.117647, F}, {0.117647, P}, {0.294118, I}, {0.294118, S}}
```

Apply[Plus, Map[First, %]]

```
1.
```

HuffmanTree[ExMISSISSIPPISCHIFF]

```
{1., {{0.411765, {{0.176471, {{0.0588235, M}, {0.117647, F}}},
  {0.235294, {{0.117647, P},
    {0.117647, {{0.0588235, C}, {0.0588235, H}}}}}}},
  {0.588235, {{0.294118, I}, {0.294118, S}}}}}}
```

```
% // TableForm
```

```

      0.176471 0.0588235 M
      0.117647 F
1.    0.411765 0.117647 P
      0.235294 0.117647 0.0588235 C
      0.0588235 H
      0.588235 0.294118 I
      0.294118 S

```

```
TableForm[HuffmanTree[ExMISSISSIPPISCHIFF]
```

```

      0.176471 0.0588235 M
      0.117647 F
1.    0.411765 0.117647 P
      0.235294 0.117647 0.0588235 C
      0.0588235 H
      0.588235 0.294118 I
      0.294118 S

```

```
HuffmanCodeTable[HuffmanTree[ExMISSISSIPPISCHIFF]
```

```

{{C, 0110}, {F, 001}, {H, 0111},
 {I, 10}, {M, 000}, {P, 010}, {S, 11}}

```

■ Example: Ex01

```
Ex01 =
```

```
(* input word:
```

```
AAAAAABBBBDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEEFFFGGGG *)
```

```
Sort[{{6 / 49, "A"}, {4 / 49, "B"}, {12 / 49, "D"},
```

```
{19 / 49, "E"}, {3 / 49, "F"}, {5 / 49, "G"}}] // N
```

```
{{0.0612245, F}, {0.0816327, B}, {0.102041, G},
```

```
{0.122449, A}, {0.244898, D}, {0.387755, E}}
```

```
Apply[Plus, Map[First, %]]
```

```
1.
```

```
HuffmanTree[Ex01]
```

```

{{1., {{0.387755, E}, {0.612245, {{0.244898, D},
 {0.367347, {{0.142857, {{0.0612245, F}, {0.0816327, B}}},
 {0.22449, {{0.102041, G}, {0.122449, A}}}}}}}}}}

```

```
% // TableForm
```

```

0.387755 E
0.244898 D
1. 0.612245 0.367347 0.142857 0.0612245 F
0.0816327 B
0.22449 0.102041 G
0.122449 A

```

```
TableForm[HuffmanTree[Ex01]]
```

```

0.387755 E
0.244898 D
1. 0.612245 0.367347 0.142857 0.0612245 F
0.0816327 B
0.22449 0.102041 G
0.122449 A

```

```
HuffmanCodeTable[HuffmanTree[Ex01]]
```

```
{A, 1111}, {B, 1101}, {D, 10}, {E, 0}, {F, 1100}, {G, 1110}}
```

■ Example Ex02

```
Ex02 =
```

```

{{1., {{0.36, "E"}, {0.64, {{0.29, {{0.14, {{0.04,
"A"}, {0.1, "C"}}}, {0.15, "B"}}},
{0.35, {{0.15, "D"}, {0.2, "F"}}}}}}}}
{{1., {{0.36, E},
{0.64, {{0.29, {{0.14, {{0.04, A}, {0.1, C}}}, {0.15, B}}},
{0.35, {{0.15, D}, {0.2, F}}}}}}}}

```

```
% // TableForm
```

```

0.36 E
0.29 0.14 0.04 A
1. 0.64 0.15 0.1 C
0.35 0.15 D
0.2 F

```

```
HuffmanCodeTable[Ex02]
```

```
{A, 1000}, {B, 101}, {C, 1001}, {D, 110}, {E, 0}, {F, 111}}
```


■ Example: Ex03

```

ExSchulz =
  Sort[{{0.4, "W"}, {0.3, "S"}, {0.2, "G"}, {0.1, "H"}}]
  {{0.1, H}, {0.2, G}, {0.3, S}, {0.4, W}}

HuffmanTree[ExSchulz]
  {{1., {{0.4, W}, {0.6, {{0.3, S}, {0.3, {{0.1, H}, {0.2, G}}}}}}}}

% // TableForm
      0.4  W
1.      0.3  S
      0.6  0.3  0.1  H
          0.3  0.2  G

HuffmanCodeTable[HuffmanTree[ExSchulz]]
  {{G, 111}, {H, 110}, {S, 10}, {W, 0}}

```

■ Example: Ex04

```

ExRechenberg = Sort[{{0.36, "E"}, {0.22, "B"},
  {0.16, "A"}, {0.14, "C"}, {0.12, "D"}}]
  {{0.12, D}, {0.14, C}, {0.16, A}, {0.22, B}, {0.36, E}}

HuffmanTree[ExRechenberg]
  {{1., {{0.38, {{0.16, A}, {0.22, B}}},
    {0.62, {{0.26, {{0.12, D}, {0.14, C}}}, {0.36, E}}}}}}

% // TableForm
      0.38  0.16  A
      0.22  B
1.      0.62  0.26  0.12  D
          0.14  C
          0.36  E

HuffmanCodeTable[HuffmanTree[ExRechenberg]]
  {{A, 00}, {B, 01}, {C, 101}, {D, 100}, {E, 11}}

```

■ Example: Ex05

```

ExSchoening = Sort[{{0.44, "A"}, {0.13, "B"},
  {0.12, "C"}, {0.16, "D"}, {0.09, "E"}, {0.05, "F"}}]
{{0.05, F}, {0.09, E}, {0.12, C}, {0.13, B}, {0.16, D}, {0.44, A}}

HuffmanTree[ExSchoening]
{{0.99, {{0.44, A}, {0.55, {{0.25, {{0.12, C}, {0.13, B}}},
  {0.3, {{0.14, {{0.05, F}, {0.09, E}}}, {0.16, D}}}}}}}}

% // TableForm
      0.44  A
          0.25  0.12  C
0.99      0.13  0.13  B
          0.55  0.14  0.05  F
          0.3   0.09  0.09  E
          0.16  0.16  D

HuffmanCodeTable[HuffmanTree[ExSchoening]]
{{A, 0}, {B, 101}, {C, 100}, {D, 111}, {E, 1101}, {F, 1100}}

```

■ Example: Ex06

```

ExUrff1 = Sort[Map[PairSwitch,
  {"A", 0.2}, {"B", 0.12},
  {"C", 0.08}, {"D", 0.2}, {"E", 0.4}]]]
{{0.08, C}, {0.12, B}, {0.2, A}, {0.2, D}, {0.4, E}}

HuffmanTree[ExUrff1]
{{1., {{0.4, {{0.2, A}, {0.2, D}}},
  {0.6, {{0.2, {{0.08, C}, {0.12, B}}}, {0.4, E}}}}}}

% // TableForm
      0.4  0.2  A
          0.2  D
1.      0.2  0.08  C
          0.12  0.12  B
          0.4  E

HuffmanCodeTable[HuffmanTree[ExUrff1]]
{{A, 00}, {B, 101}, {C, 100}, {D, 01}, {E, 11}}

```

```
100 * MeanCodeLength[ExUrff1]
```

```
2.2
```

For comparison ("Probe"):

```
{{"A", "111"}, {"B", "1101"}, {"C", "1100"}, {"D", "10"}, {"E", "0"}}
```

```
3 * 0.2 + 4 * 0.12 + 4 * 0.08 + 2 * 0.2 + 1 * 0.4
```

```
2.2
```

```
ExUrff2 = Sort[Map[PairSwitch,
```

```
  {"A", 0.1}, {"B", 0.12},
```

```
  {"C", 0.18}, {"D", 0.2}, {"E", 0.4}]]]
```

```
{ {0.1, A}, {0.12, B}, {0.18, C}, {0.2, D}, {0.4, E} }
```

```
HuffmanTree[ExUrff2]
```

```
{ {1., { {0.4, E}, {0.6, { {0.22, { {0.1, A}, {0.12, B} }},
  {0.38, { {0.18, C}, {0.2, D} } } } } }
```

```
% // TableForm
```

```

      0.4  E
1.    0.6  0.22  0.1  A
      0.38  0.12  0.12  B
      0.18  0.18  0.18  C
      0.2   0.2   0.2   D
```

```
HuffmanCodeTable[HuffmanTree[ExUrff2]]
```

```
{ {A, 100}, {B, 101}, {C, 110}, {D, 111}, {E, 0} }
```

■ Example: Deutsch mit Leerzeichen

■ Some Utilities