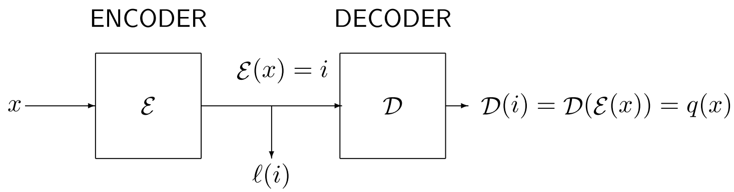
Quantization and lossless coding

Recall basic model of a quantizer



The components of the quantizer are

- An *encoder* $\mathcal{E} : A \to \mathcal{I}$. So far assumed $\mathcal{I} = \mathcal{Z} = \{0, 1, 2, \ldots\}$ or some subset
- A *decoder* $\mathcal{D}\mathcal{I} \to \hat{A}$.
- Length function $\ell(i)$ satisfying $\sum_{i \in \mathcal{I}} e^{-\ell(i)} \leq 1$.

Consider the special case:

- input alphabet discrete
- distortion measure satisfies $d(x, y) > 0$ if $x \neq y$
- require $D(q) = 0$.

noiseless or *lossless* compression

What is minimum achievable average rate with no distortion?

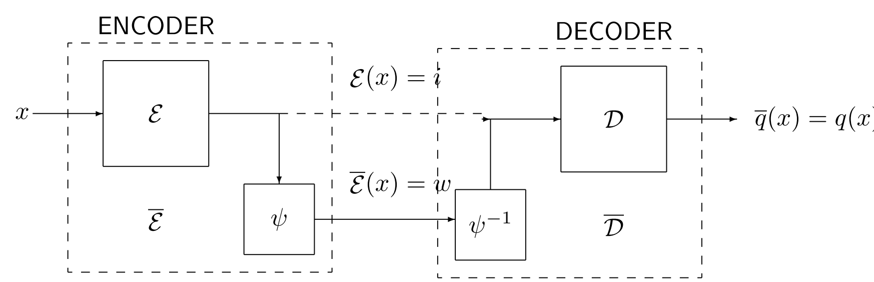
Helpful in 0 distortion compression, but also in general lossy quantization as a component of the overall quantizer. Consider the general case first.

Equivalent quantizers

Overall operation of q depends only on the cascade of encoder and decoder, not on their individual structure.

In particular, the index set $\mathcal{I}$ can be relabeled in any way preserving a unique label for each member of the index set.

Equivalently, overall quantization operation not be changed by an internal invertible mapping:



Two models q and $\overline{q}$ are *equivalent* if $\overline{q}(x) = q(x)$ for all $x \in A$.

Formally:

Lemma 7. *Given a quantizer $q = (\mathcal{E}, \mathcal{D})$ and a discrete set $\mathcal{W}$ for which $\psi : \mathcal{E}(A) \to \mathcal{W}$ is an invertible mapping onto $\mathcal{W}$, i.e., $\mathcal{W}$ is the range space of ψ and there is an inverse mapping $\psi^{-1} : \mathcal{W} \to \mathcal{E}(A)$ satisfying $\psi^{-1}(\psi(i)) = i$ for all $i \in \mathcal{E}(A)$. Then an equivalent model $\overline{q} = (\overline{\mathcal{E}}, \overline{\mathcal{D}})$ with index set $\overline{\mathcal{I}} = \mathcal{W}$ for the quantizer q consists of an encoder $\overline{\mathcal{E}} : A \to \mathcal{W}$ defined by $\psi(\mathcal{E}(x))$ and a decoder $\overline{\mathcal{D}} : \mathcal{W} \to \hat{A}$ defined by $\overline{\mathcal{D}}(w) = \mathcal{D}(\psi^{-1}(w))$ so that*

$$\overline{q}(x) = \mathcal{D}(\mathcal{E}(x)) = \overline{\mathcal{D}}(\overline{\mathcal{E}}(x)) = q(x).$$

Justifies use of $\mathcal{Z}$ as canonical index set — for any other discrete set there is an equivalent quantizer in terms of the overall input/output mapping which uses $\mathcal{Z}$ as the encoder output/decoder input alphabet.

Keep in mind the underlying index cost or weighting will differ. In fact equivalent quantizers motivate the definition of admissible length functions and codeword weightings.

The invertible mapping $\psi : \mathcal{I} \to \mathcal{W}$ is a *lossless coding* or *noiseless coding* of $\mathcal{I}$ onto $\mathcal{W}$.

Suppose each encoded input $\mathcal{E}(x) = i \in \mathcal{Z}$ is to be transmitted through or stored in a medium that accepts and delivers one or more symbols drawn from an m-ary alphabet, say $\mathcal{Z}_m = \{0, 1, \ldots, m-1\}$.

i.e. i is mapped by invertible function $\psi : \mathcal{I} \to \mathcal{W} \subset \mathcal{Z}_m^*$, space of all finite length sequences of symbols drawn from $\mathcal{Z}_m$.

The resulting m-ary sequence (channel codeword)

$$w = \psi(i) \in \mathcal{W} = \psi(\mathcal{I}) \subset \mathcal{Z}_m^*$$

Decoder applies inverse mapping $\psi^{-1} : \psi(\mathcal{I}) \to \mathcal{I}$ to reproduce the index i.

An equivalent quantizer overall, and provides natural notion of *cost* or *instantaneous rate* or *length function* of an index — the length length$(\psi(i))$ of the symbol sequence $\psi(i)$

E.g., if $m = 2$, then length$(\psi(i))$ = number of bits required to be transmitted or stored to uniquely specify the index selected by the encoder to the decoder.

Concentrate on the binary case, but keep in mind that ternary, quaternary, or other alphabets are also of interest in some problems and the ideas all generalize in a straightforward manner.

Since the length depends on the choice of m, denote

$$\ell_m(i) = \text{length}(\psi(i)).$$

"Neutral" length function normalizes the units to nats, that is,

$$\ell(i) = \ell_m(i) \ln m$$

so that $m^{\ell_m(i)} = e^{\ell(i)}$.

Often denote ℓ_m by ℓ if base clear from context.

Consider for example the three binary codings: where $\mathcal{I}$ is $\mathcal{Z}_{16}$.

$\mathcal{I}$	$\mathcal{W}_0$	$\mathcal{W}_1$	$\mathcal{W}_2$
0	0000	0	0
1	0001	1	10
2	0010	10	110
3	0011	11	1110
4	0100	100	11110
5	0101	101	111110
6	0110	110	1111110
7	0111	111	11111110
8	1000	1000	111111110
9	1001	1001	1111111110
10	1010	1010	11111111110
11	1011	1011	111111111110
12	1100	1100	1111111111110
13	1101	1101	11111111111110
14	1110	1110	111111111111110
15	1111	1111	111111111111111

All are invertible.

First is a *fixed-length code* or *fixed-rate code*, other two have variable-length words.

In the fixed-length code case, length$(\psi(i)) = \log_2 N(\mathcal{I})$ for all $i \in \mathcal{I} \Rightarrow$

$$R(\mathcal{E}) = E[\ell(\mathcal{E}(X))] = \log_2 N \text{ bits.}$$

Can also consider combined cost/penalty/rate definition as before:

$$r(i) = (1-\eta)\,\text{length}(\psi(i)) + \eta \log N(\mathcal{I}) \tag{61}$$

When dealing with the pure lossless coding case with variable rate codes allowed, $\eta = 0$ is appropriate since N determined by input alphabet.

Uniquely decodable codes

In practice apply quantizer to sequence of input symbols $\Rightarrow$ need more than invertibility, must have the property that *sequences* of channel codewords are also invertible: for every positive integer K and every sequence of indices $i^K = (i_0, i_1, \ldots, i_{K-1})$, the sequence of channel codewords $\psi(i^K) = (\psi(i_0), \psi(i_1), \ldots, \psi(i_{K-1}))$ uniquely determines i^K.

$\Leftrightarrow$ *uniquely decodable.*

Fixed-rate code example OK, but second code is not uniquely decodable:

Suppose receive 1011. Could be 1 0 11 (1 0 3 sent) or 10 11 (2 3 sent) or 1011 (11 sent).

Second variable-rate code is uniquely decodable.

McMillan (simplified by Karush) provided a simple necessary condition for the existence of a uniquely decodable code given a set of positive integers $\{\ell(i);\ i \in \mathcal{I}\}$ describing the allowed lengths of the channel codewords in m-ary symbols.

Lemma 8. *A necessary condition for the existence of an m-ary uniquely decodable code having a collection of codeword lengths in m-ary units $\{\ell(i);\ i \in \mathcal{I}\}$ is that the lengths satisfy the inequality*

$$\sum_{i \in \mathcal{I}} m^{-\ell(i)} \le 1. \tag{62}$$

(Kraft inequality)

The proof is for $m = 2$, general proof is similar.

Proof Suppose $\{\ell(i);\ i \in \mathcal{I}\}$ is a set of lengths in bits of channel codewords of a uniquely decodable lossless code with binary channel symbols.

Order so $\ell(i)$ is nondecreasing in i.

If the code with lengths $\{\ell(i);\ i \in \mathcal{I}\}$ is uniquely decodable, then so also must be the possibly smaller codebook with lengths $\{\ell(i);\ i \in \mathcal{I}, \ell(i) \le L\}$

Smaller code will be a uniquely decodable code for smaller subset of $\mathcal{I}$, say $\mathcal{I}^{(L)}$.

For any positive integer K, a sequence of indices $i = (i_0, \ldots, i_{K-1})$, $i_n \in \mathcal{I}^{(L)}$, must produce a sequence of binary channel codewords with length $\ell_{i_n};\ n = 0, 1, \ldots, K-1$.

Total length of the resulting sequence of channel codewords will be

$$\text{length}(\psi(i)) = \sum_{n=0}^{K-1} \ell_{i_n} = \sum_{\ell=1}^{L} N(K,\ell)\ell,$$

where $N(K,\ell)$ = number of input sequences $i = (i_0, \ldots, i_{K-1})$, $i_n \in \mathcal{I}^{(L)}$ which yield a channel sequence of total length ℓ.

Since the code is uniquely decodable,

$$N(K,\ell) \le 2^{\ell} \tag{63}$$

total number of length ℓ binary sequences.

The proof of the lemma combines this inequality with a simple

trick. To bound the sum

$$S_L = \sum_{i:\ell(i) \le L} 2^{-\ell(i)}.$$

consider instead the power

$$\begin{aligned} S_L^K &= \left(\sum_{i:\ell(i)\le L} 2^{-\ell(i)} \right)^K \\ &= \left(\sum_{i_0:\ell_{i_0}\le L} 2^{-\ell_{i_0}} \right) \times \left(\sum_{i_1:\ell_{i_1}\le L} 2^{-\ell_{i_1}} \right) \times \cdots \\ &\quad \times \left(\sum_{i_{K-1}:\ell_{i_{K-1}}\le L} 2^{-\ell_{i_{K-1}}} \right) \end{aligned}$$

$$\begin{aligned} &= \sum_{i_0:\ell_{i_0}\le L} \sum_{i_1:\ell_{i_1}\le L} \cdots \sum_{i_{K-1}:\ell_{i_{K-1}}\le L} 2^{-\sum_{n=0}^{K-1} \ell_{i_n}} \\ &= \sum_{\ell=1}^{L} N(K,\ell) 2^{-\ell}. \end{aligned}$$

Applying inequality (63) yields $S_L^K \le L$ or $S_L \le L^{1/K}$.

For a fixed L, this must hold for all K and hence it must also hold in the limit as $K \to \infty$, in which case the right hand side is 1.

Thus for every fixed finite L

$$S_L = \sum_{i:\ell(i)\le L} 2^{-\ell(i)} \le 1.$$

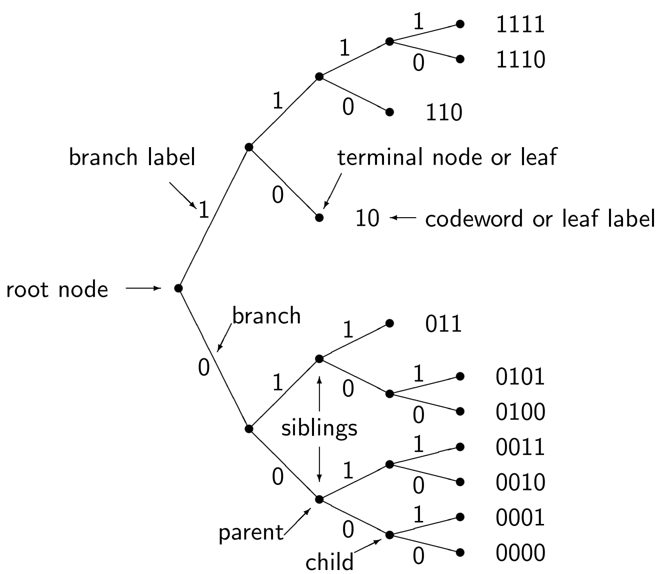
This inequality holds for all L and hence it must also hold true in the limit, which proves the lemma. □

Kraft is a *necessary* condition for uniquely decodable, hence optimization need consider only lengths satisfying Kraft's inequality.

Tree-structued codes

Simple and effective way to construct a uniquely decodable code is is to force the prefix-free property of the third example by use of tree-structured codes. It will be seen that no code without this property can do better.

Prefix-free/prefix/instantaneous/tree-structured codes can be simply visualized:



Tree has a depth of 4 ⇒ maximum length available = 4.

Any sequence of binary symbols labeling branches from the root node to a leaf or terminal node = a channel codeword.

No descendent node of a leaf can be a codeword ⇔ no codeword can be a prefix of another codeword. Conversely, any prefix code can be depicted as a binary tree of this form.

Simple necessary and sufficient condition for the existence of such a code for a given collection of channel codeword lengths $\ell(i)$, $i \in \mathcal{I}$:

Lemma 9. *There exists an m-ary tree-structured channel code (prefix code) having a collection of codeword lengths $\{\ell(i);\ i \in \mathcal{I}\}$ if and only if*

$$\sum_{i \in \mathcal{I}} m^{-\ell(i)} \leq 1.$$

Proof For $m = 2$.

Necessity follows from Lemma 8, but prove using tree sctructure.

Suppose binary tree exists with lengths $\ell(i)$; $i \in \mathcal{I}$ from root to leaves.

Assume that the $\ell(i)$ are in nondecreasing order: $\ell(0) \leq \ell(1) \leq \ell(2) \cdots$.

Fix L.

Consider the codeword with length $\ell(0)$.

Since a codeword, all of the descendent nodes are removed from the tree.

If these nodes had remained, there would have been at depth L in the tree $2^{L-\ell(0)}$ descendent nodes.

E.g., in example shortest codeword has $\ell(0) = 1$, so there would have been $2^{4-1} = 8$ descendants at level 4

Same is true for the remaining lengths up to and including lengths for which $\ell(i) = L$: the leaf with length $\ell(i)$ for $\ell(i) \leq L$ would have had $2^{L-\ell(i)}$ descendants at depth L if the tree had been allowed to grow.

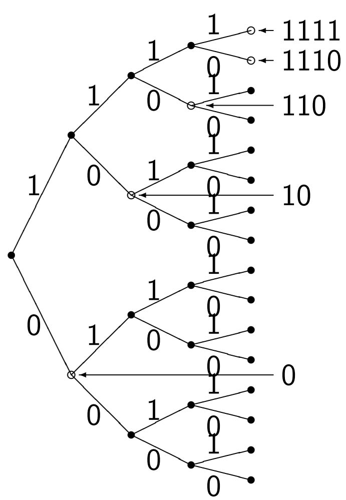
Since the collection of these descendants at depth L cannot contain more nodes than the total 2^L of the full tree at length L,

$$\sum_{i:\ell(i)\leq L} 2^{L-\ell(i)} \leq 2^L$$

or

$$\sum_{i:\ell(i)\leq L} 2^{-\ell(i)} \leq 1. \tag{64}$$

The left hand side of (64) is nondecreasing as L grows, so it must have a limit which is also less than 1 $\Rightarrow$ Kraft inequality



Same figure illustrates proof of sufficiency.

Given a set of ordered lengths $\ell(i); i \in \mathcal{I}$, pick an arbitrary node at depth $\ell(0)$ into the tree and declare it a leaf and make the binary pathmap to the node the channel codeword.

For any depth $L \geq \ell(0)$ this removes $2^{L-\ell(0)}$ nodes at depth L that might otherwise have been available.

In particular, at depth $\ell(1)$ in the tree $2^{\ell(1)-\ell(0)}$ nodes have been removed.

Provided this has not removed all of the possible $2^{\ell(1)}$ nodes at depth $\ell(1)$, that is, $2^{\ell(1)-\ell(0)} < 2^{\ell(1)}$ or $2^{-\ell(0)} < 1$, which is ensured by the Kraft inequality.

Continue in this manner.

Suppose words picked with lengths $\ell(0), \ell(1), \ell(N-1)$ and we wish to choose another of length $\ell(N)$, and the Kraft inequality holds for $\{\ell(0), \ell(1), \ell(N-1), \ell(N)\}$.

Existing codewords together cause the loss of $\sum_{i=0}^{N-1} 2^{\ell(N)-\ell(i)}$ nodes in the tree of depth $\ell(N)$, but at least one will remain if this total is strictly less than $2^{\ell(N)}$, that is, if

$$\sum_{i=0}^{N-1} 2^{\ell(N)-\ell(i)} < 2^{\ell(N)}$$

or

$$\sum_{i=0}^{N-1} 2^{-\ell(i)} < 1,$$

which will be the case since

$$\sum_{i=0}^{N} 2^{-\ell(i)} \leq 1.$$

If $\mathcal{I}$ if finite, this will terminate at some point. If it is infinite, there will always remain at least one open leaf at the next $\ell(N)$. □

Length function $\ell = \{\ell(i);\ i \in \mathcal{I}\}$ in m-ary units is m-admissible if it satisfies Kraft's inequality

$$\sum_{i \in \mathcal{I}} m^{-\ell(i)} \leq 1. \tag{65}$$

Alternatively,ℓ can be expressed in nats and Kraft's inequality is

$$\sum_{i \in \mathcal{I}} e^{-\ell(i)} \leq 1 \tag{66}$$

and that $\ell(i)/\ln m = (\log_m e)\ell(i)$ is an integer for all $i \in \mathcal{I}$.

Thus (66) is a *necessary* condition for the existence of a uniquely decodable code, but it is not sufficient unless the suitably normalized values are actually integers for a channel symbol alphabet size of interest.

Length function ℓ satisfying (66) will be said to be *admissible* whether or not it is m-admissible for some m. This justifies the use of the term in the definition of the length function component of a quantizer.

If a length function ℓ is admissible in any sense, then so is the resulting combined constraint instantaneous rate function $r_\eta(i) = (1-\eta)\ell(i) + \eta \ln N(\mathcal{I})$ in the same sense.

Shannon's lossless coding theorem

Consider lossless code ψ of a discrete random variable U with alphabet A_U and pmf $p(a) = \Pr(U = a), a \in A_U$.

E.g.,(1) in the general quantization setup $U = \mathcal{E}(X)$ and $A_U = \mathcal{I}$; (2) in the purely lossless compression problem, $U = X$ and $A_U = A$.

Consider $m = 2$, a binary channel symbol alphabet, and let $\ell(a) =$ length$(\psi(a))$ be the length function (in bits).

Lemma 3 with $w(a) = 2^{-\ell(a)} \Rightarrow$ (part of Shannon's famous first theorem)

Lemma 10. *Given a uniquely decodable noiseless variable length code with encoder ψ operating on a discrete random variable U with entropy $H(U)$, then the resulting average codeword length satisfies*

$$E[\ell(U)] \geq H(U); \tag{67}$$

i.e., the average length of the code can be no smaller than the entropy of the marginal pmf. The inequality is an equality iff

$$p(a) = 2^{-\ell(a)} \text{ for all } a \in A. \tag{68}$$

To achieve bound, (68) must be satisfied, which is only possible if all probabilities that are powers of $1/2$.

Best length function might not yield practicable lossless code.

More generally, for an m-ary lossless code, (68) will correspond to an actual set of lengths if $\ell_m(a) = -\log_m p(a)$ are all integers.

The average length can never be lower than the entropy. How close can one get?

Lemma 11. *There exists a uniquely decodable scalar noiseless code for a source with marginal pmf p for which the average codeword length satisfies*

$$E[\ell(U)] < H(p) + 1. \tag{69}$$

Proof: Given $p(a)$, for each a, choose $\ell(a)$ to satisfy

$$2^{-\ell(a)} \leq p(a) < 2^{-\ell(a)+1} \tag{70}$$

or, equivalently,

$$-\log p(a) \leq \ell(a) < -\log p(a) + 1 \tag{71}$$

Intuitively,

$$\ell(a) \approx -\log p(a).$$

Satisfy Kraft inequality since

$$\sum_a 2^{-\ell(a)} \leq \sum_a p(a) = 1,$$

$\Rightarrow$ there is a prefix-free (and hence a uniquely decodable) code with these lengths. From (71), the average length must satisfy the bound of the theorem. □

Combining Lemmas 10-11 yields Shannon's lossless coding theorem (Shannon's first theorem):

Theorem 1. Shannon's lossless coding theorem

For any uniquely decodable lossless code of a discrete source U with pmf p,

$$E[\ell(U)] \geq H(p); \tag{72}$$

and there exists a prefix-free code for which

$$E[\ell(U)] < H(p) + 1. \tag{73}$$

The Shannon code is simple and provides a useful general bound, but other codes can in general provide either superior performance or lower complexity implementations when the discrete alphabet is large.

Huffman codes

D. A. Huffman (1952) provided a constructive technique for an optimal lossless code given the underlying pmf.

The construction is based on optimality properties of lossless codes, which are considered next.

Lemma 12. *Suppose that ψ is a uniquely decodable variable length binary noiseless source code and that $\{\ell(a)$ is the collection of codeword lengths. Then there is a prefix-free code with the same lengths and the same average length.*

Follows since the lengths of a uniquely decodable code must satisfy the Kraft inequality $\Rightarrow$ there must exist a prefix-free code with these lengths (from the tree construction of the sufficiency proof for the Kraft inequality of Lemma 9).

Theorem 2. *An optimum binary prefix-free code has the following properties:*

(i) *If the codeword for input symbol a has length $\ell(a)$, then $p(a) > p(b)$ implies that $\ell(a) \leq \ell(b)$; that is, more probable input symbols have shorter (at least, not longer) codewords.*

(ii) *The two least probable input symbols have codewords which are equal in length and differ only in the final symbol.*

Proof:

(i) If $p(a) > p(b)$ and $\ell(a) > \ell(b)$, then exchanging codewords will cause a strict decrease in the average length. Hence the original code could not have been optimum.

(ii) Suppose the two codewords have different lengths. A prefix of a longer codeword can not itself be a codeword, and hence final symbol of the longer codeword can be deleted without confusion. This strictly decreases the average length of the code, #. $\Rightarrow$ the two least probable codewords must have equal length.

Suppose these two codewords differ in some position other than the final one. The final binary symbol could then be removed yielding a shorter code without confusion (the shorter codewords could still be distinguished and the prefix condition precludes the possibility of confusion with another codeword). This, however, yields a strict decrease in average length, #. □

The theorem provides an iterative design technique which Huffman demonstrated yields an optimal code and yields the entropy lower bound if the input probabilities are powers of $1/2$.

Order the input symbols in terms of probability, that is, $p(a_0) \geq p(a_1) \geq \cdots \geq p(a_{M-1})$.

2 least probable symbols must end up with codewords of the same length which differ only in the final binary symbol.

So begin a code tree with two terminal nodes with branches extending back to a common node. Label one branch 0 and the other 1. Consider these two input symbols to be *tied* together and form a single new symbol in a reduced alphabet A' with $M-1$ symbols in it.

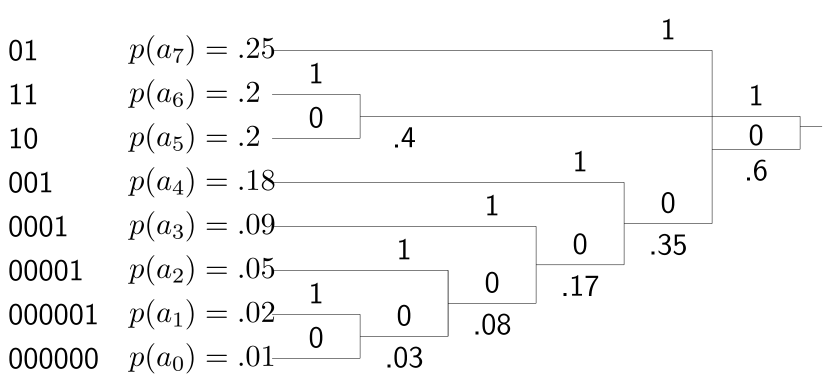
Next find an optimal code for the reduced alphabet A' with probabilities $p(a_m);\ m = 0, 1, \cdots, M-3$ and $p(a_{M-1}) + p(a_{M-2})$.

A prefix code for A' implies a prefix code for A by adjoining the final branch labels already selected. If the prefix code for A' is optimal, then so is the induced code for A.

Continue in this fashion:

- The probability of each node is found by adding up the probabilities of all input symbols connected to the node.
- At each step the two least probable nodes in the tree are found.
- These nodes are tied together and a new node is added with branches to each of the two low probability nodes and with one branch labeled 0 and the other 1.
- The procedure is continued until only a single node remains.

Example:



A variation on this technique works for nonbinary alphabets (and the performance can be better).

Vector lossless coding

Consider coding vector $U^N = (U_0, \ldots, U_{N-1})$ of samples from a stationary discrete random process $\{U_n\}$

$(U_n = \mathcal{E}(X_n)$ or $X_n)$

Will allow the vector dimension N to grow.

$p_{U^N} =$ pmf for a source vector $U^N = (X_0, \cdots, U_{N-1})$

From the lossless coding theorem, any uniquely decodable lossless code will have average length bounded below by

$$E[\psi(U^N)] \geq \frac{1}{N} H(U^N) \tag{74}$$

where

$$H(U^N) = H(p_{U^N}) = -\sum_{u^N} p_{U^N}(u^N) \log p_{U^N}(u^N).$$

and there exists a prefix-free code for which

$$E[\psi(U^N)] < \frac{1}{N} H(U^N) + \frac{1}{N}. \tag{75}$$

E.g., Huffman or Shannon code

If the input process is stationary, then the *entropy rate* of the source

$$\overline{H} = \inf_N \frac{H(U^N)}{N} = \lim_{N\to\infty} \frac{H(U^N)}{N}.$$

If the source is iid, then $\overline{H} = H(U_0)$.

Conclusion: the entropy rate $\overline{H}$ of the input process $\{U_n\}$ is an unbeatable lower bound to the average length that can be achieved using uniquely decodable lossles codes, and the bound can be approached arbitrarily closely by Huffman (or Shannon) coding sufficiently large vectors. Unfortunately, however, this does not provide a solution to the practical problem of lossless coding because the complexity of the lossless codes grows too high to be practicable as N gets large.

Solution: Lempel-Ziv, arithmetic codes effectively code long (variable-length) sequences in implementable manner.