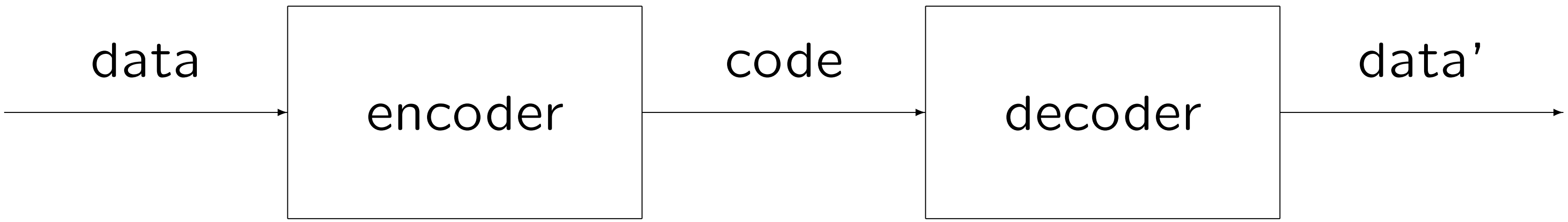
# Waiting and Weighting

(Two Universal Source Coding Concepts)

Frans Willems, Eindhoven University of Technology

# Universal Noiseless Source Coding




Properties:

- Assumption: binary data, binary code.
- Requirement: data' $\equiv$ data.
- Objective: length(code) $<$ length(data).
- Universality: source statistics *unknown* to encoder and decoder.

# Two Concepts

- **Waiting**
  We discuss waiting times, Kac's [1947] theorem, and its connection to universal source coding (Willems [1986,1989], and Wyner and Ziv [1989,1994]).

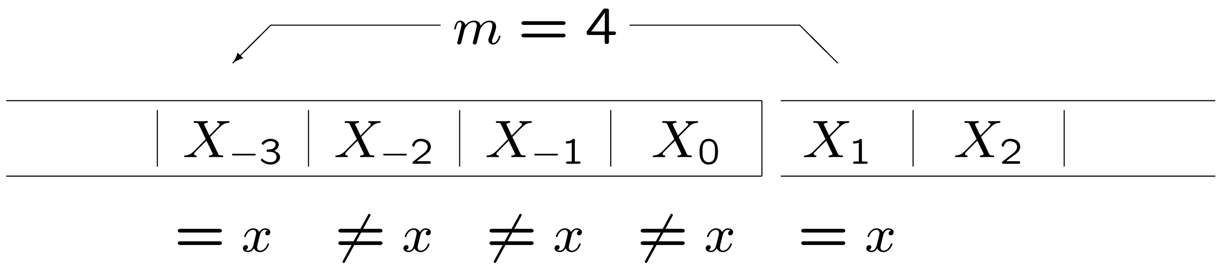
- **Weighting**
  We discuss arithmetic coding, weighted coding distributions, and the Context-Tree Weighting [1995] algorithm.

# Waiting Times

Consider the discrete stationary and ergodic source

$$\cdots, X_{-3}, X_{-2}, X_{-1}, X_0, X_1, X_2, \cdots.$$

Suppose that $X_1 = x$ for some symbol-value $x \in \mathcal{X}$ with $\Pr\{X_1 = x\} > 0$. We say that the *waiting time* of the $x$ that occurred at time $t = 1$ is $m$ if $X_{1-m} = x$ and $X_t \neq x$ for $t = 2 - m, \cdots, 0$.




Let $Q_m(x)$ be the conditional probability that the waiting time of this $x$ is $m$, given that $X_1 = x$. Hence

$$Q_m(x) = \Pr\{X_{1-m} = x, X_{2-m} \neq x, \cdots, X_0 \neq x | X_1 = x\}.$$

## Kac's Result

The *average* waiting time for symbol-value $x$ with $\Pr\{X_1 = x\} > 0$ is defined as

$$T(x) \triangleq \sum_{m=1,2,\cdots} mQ_m(x).$$

**Kac [1947]:** For stationary and ergodic sources

$$T(x) = \sum_{m=1,2,\cdots} mQ_m(x) = \frac{1}{\Pr\{X_1 = x\}}, \tag{1}$$

for any $x$ with $\Pr\{X_1 = x\} > 0$.

## Blocking

Let $L$ be a positive integer. When $\cdots, X_{-1}, X_0, X_1, X_2, \cdots$ is stationary and ergodic, then

$$\cdots, \begin{pmatrix} X_{-1} \\ X_0 \\ \cdots \\ X_{L-2} \end{pmatrix}, \begin{pmatrix} X_0 \\ X_1 \\ \cdots \\ X_{L-1} \end{pmatrix}, \begin{pmatrix} X_1 \\ X_2 \\ \cdots \\ X_L \end{pmatrix}, \begin{pmatrix} X_2 \\ X_3 \\ \cdots \\ X_{L+1} \end{pmatrix}, \cdots$$

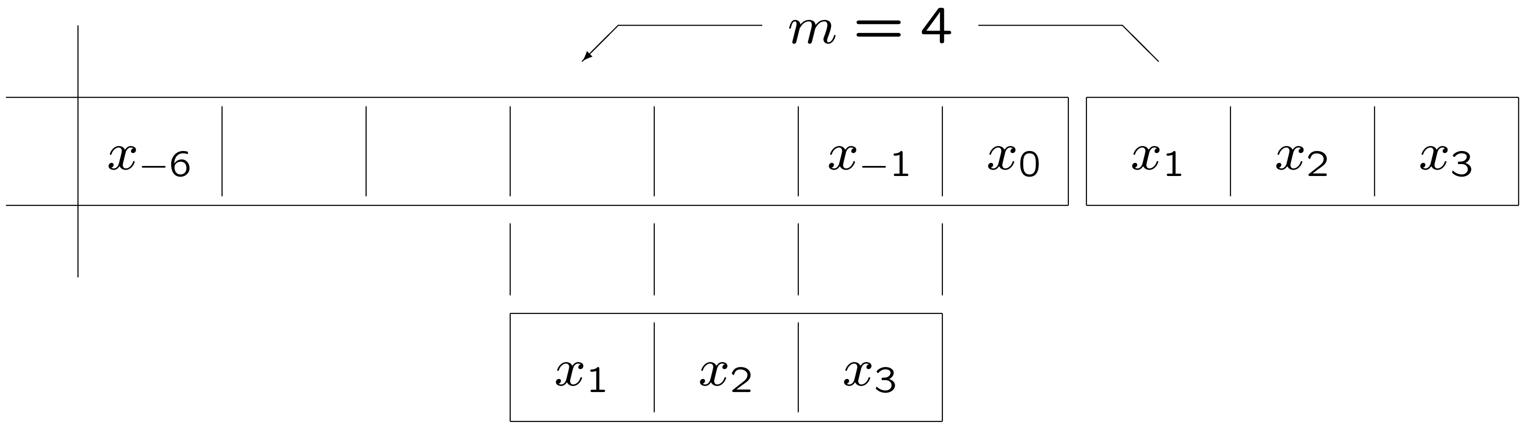
is stationary and ergodic too.

Therefore Kac's result holds also for "sliding" $L$-blocks. A waiting time equal to $m$ means that $m$ is the smallest positive integer for which

$$\begin{pmatrix} X_{1-m} \\ X_{2-m} \\ \cdots \\ X_{L-m} \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ \cdots \\ X_L \end{pmatrix}.$$

# A Universal Source Coding Method (Willems [1986,1989])

Suppose that our source is *binary* i.e. $X_t \in \{0,1\}$ for all integer $t$.




An encoder wants to transmit a source block $x_1^L \triangleq x_1, x_2, \cdots, x_L$ to a decoder. Both encoder and decoder have access to buffers containing all previous source symbols $\cdots, x_{-2}, x_{-1}, x_0$.

Using these previous source symbols the encoder can determine the waiting time of $x_1^L$. It is the smallest integer $m$ that satisfies

$$x_{1-m}^{L-m} = x_1^L,$$

where $x_{1-m}^{L-m} \triangleq x_{1-m}, x_{2-m}, \cdots, x_{L-m}$.

The waiting time $m$ is sent to the decoder. With $m$ and using the previous source symbols the decoder can reconstruct $x_1^L$.

Code for the waiting time $m$ for $L = 3$:

| $m$ | $p(m)$ | $c(m)$ | $l(m)$ |
|---|---|---|---|
| 1 | 00 | - | 2+0=2 |
| 2 | 01 | 0 | 2+1=3 |
| 3 | 01 | 1 | 2+1=3 |
| 4 | 10 | 00 | 2+2=4 |
| 5 | 10 | 01 | 2+2=4 |
| 6 | 10 | 10 | 2+2=4 |
| 7 | 10 | 11 | 2+2=4 |
| $\geq 8$ | 11 | $x_0x_1x_2$ | 2+3=5 |

In general we get fixed length codes with lengths $0, 1, \cdots, L-1$ and a "copy"-code with length $L$. We use a preamble $p(m)$ of $\lceil \log_2(L+1) \rceil$ bits to specify one of these $L+1$ alternative codes.

In general we get

$$l(m) = \begin{cases} \lceil \log_2(L+1) \rceil + \lfloor \log_2 m \rfloor & \text{if } m < 2^L, \\ \lceil \log_2(L+1) \rceil + L & \text{if } m \geq 2^L. \end{cases}$$
$$\leq \lceil \log_2(L+1) \rceil + \log_2 m.$$

**Note:** Buffers need only contain the previous $2^L - 1$ source symbols!

After processing the block $x_1^L$ the encoder and decoder can update their buffers. After that the next block

$$x_{L+1}^{2L} \triangleq x_{L+1}, x_{L+2}, \cdots, x_{2L}$$

is processed in a similar way, etc.

## Waiting-time algorithm: analysis

Assume that a certain $x_1^L$ occurred as first block. What is the average codeword length $L(x_1^L)$ for $x_1^L$?

$$\begin{aligned} L(x_1^L) &= \sum_{m=1,2,\cdots} Q_m(x_1^L) l(m) \\ &\leq \sum_{m=1,2,\cdots} Q_m(x_1^L) \left( \lceil \log_2(L+1) \rceil + \log_2 m \right) \\ &\overset{(a)}{\leq} \lceil \log_2(L+1) \rceil + \log_2 \left( \sum_{m=1,2,\cdots} m Q_m(x_1^L) \right) \\ &\overset{(b)}{=} \lceil \log_2(L+1) \rceil + \log_2 \frac{1}{\Pr\{X_1^L = x_1^L\}}. \end{aligned}$$

Here (a) follows Jensen's inequality ($E[f(X)] \leq f(E[X])$ for a convex-$\cap$ function $f(x)$ of $x$). Furthermore (b) follows from Kac's theorem.

The probability that $x_1^L$ occurred as first block is $\Pr\{X_1^L = x_1^L\}$. For the average codeword length $L(X_1^L)$ we get

$$\begin{aligned}
L(X_1^L) &= \sum_{x_1^L} \Pr\{X_1^L = x_1^L\} L(x_1^L) \\
&\leq \sum_{x_1^L} \Pr\{X_1^L = x_1^L\} \left( \lceil \log_2(L+1) \rceil + \log_2 \frac{1}{\Pr\{X_1^L = x_1^L\}} \right) \\
&= \lceil \log_2(L+1) \rceil + H(X_1^L).
\end{aligned}$$

For the rate $R_L$ we obtain

$$R_L = \frac{L(X_1^L)}{L} \leq \frac{H(X_1^L)}{L} + \frac{\lceil \log_2(L+1) \rceil}{L}.$$

## Achieving entropy

Since

$$\lim_{L \to \infty} \frac{H(X_1^L)}{L} \triangleq H_\infty(X)$$

and

$$\lim_{L \to \infty} \frac{\lceil \log_2(L+1) \rceil}{L} = 0$$

we may conclude that

$$\lim_{L \to \infty} R_L = H_\infty(X)$$

and therefore the waiting time algorithm achieves entropy.

Note that this method is **universal**. Although the statistics of the source are unknown, entropy is achieved.

## Relation between waiting times and entropy

Again assume that $\cdots, X_{-1}, X_0, X_1, X_2, \cdots$ is stationary and ergodic with entropy $H_\infty(X)$.

Let the random variable $M$ be the waiting time of the source block $X_1^L$.

**Wyner and Ziv [1989]**: Fix $\epsilon > 0$. Then

$$\lim_{L\to\infty} \Pr\left\{M \geq 2^{L(H_\infty(X)+\epsilon)}\right\} = 0. \tag{2}$$

This result was crucial in proving that the Ziv-Lempel [1977] algorithm achieves entropy (Wyner and Ziv [1994]).

## Intermezzo: Asymptotic Equipartion Property

Let $\cdots, X_{-1}, X_0, X_1, \cdots$ be stationary and ergodic with entropy $H_\infty(X)$.

Define for a fixed $\delta > 0$ the set of $\delta$-typical $L$-sequences

$$\mathcal{A}_\delta^L = \left\{ x_1^L : \left| \frac{1}{L} \log_2 \frac{1}{\Pr\{X_1^L = x_1^L\}} - H_\infty(X) \right| \leq \delta \right\}, \tag{3}$$

then (McMillan [1953]):

$$\lim_{L\to\infty} \Pr\{X_1^L \in \mathcal{A}_\delta^L\} = 1. \tag{4}$$

This is called the Asymptotic Equipartition Property (A.E.P.).

By definition for each $\delta$-typical $L$-sequence $x_1^L$ we have that

$$2^{-L(H_\infty(X)+\delta)} \leq \Pr\{X_1^L = x_1^L\} \leq 2^{-L(H_\infty(X)-\delta)}.$$

Therefore

$$\begin{aligned} 1 &\geq \sum_{x_1^L \in \mathcal{A}_\delta^L} \Pr\{X_1^L = x_1^L\} \\ &\geq \sum_{x_1^L \in \mathcal{A}_\delta^L} 2^{-L(H_\infty(X)+\delta)} \\ &= |\mathcal{A}_\delta^L| 2^{-L(H_\infty(X)+\delta)}, \end{aligned}$$

and consequently

$$|\mathcal{A}_\delta^L| \leq 2^{L(H_\infty(X)+\delta)}. \tag{5}$$

Thus the typical set contains only roughly $2^{LH_\infty(X)}$ sequences. Nevertheless it has probability almost equal to one.

## Proof of Wyner-Ziv theorem:

Consider the typical set $\mathcal{A}_\delta^L$ for $\delta = \epsilon/2$. Then

$$
\begin{aligned}
&\Pr\{M \geq 2^{L(H_\infty(X)+\epsilon)}\} \\
&= \Pr\{M \geq 2^{L(H_\infty(X)+\epsilon)} \wedge X_1^L \in \mathcal{A}_\delta^L\} + \Pr\{M \geq 2^{L(H_\infty(X)+\epsilon)} \wedge X_1^L \notin \mathcal{A}_\delta^L\}.
\end{aligned}
$$

First we consider the second term. Observe that

$$
\Pr\{M \geq 2^{L(H_\infty(X)+\epsilon)} \wedge X_1^L \notin \mathcal{A}_\delta^L\} \leq \Pr\{X_1^L \notin \mathcal{A}_\delta^L\} \to 0 \text{ for } L \to \infty \quad (6)
$$

by the AEP, see (4).

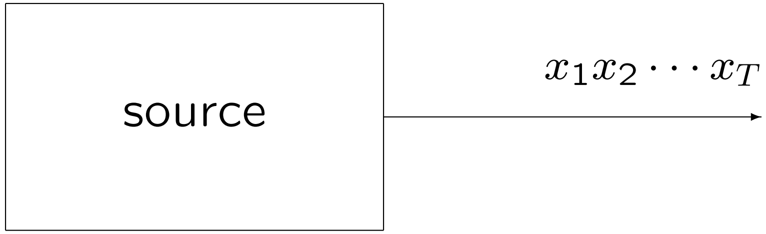
For the first term, if we use the notation $H_\infty \triangleq H_\infty(X)$ and $P(x_1^L) \triangleq \Pr\{X_1^L = x_1^L\}$, we can write

$$
\begin{aligned}
\Pr\{M \geq 2^{L(H_\infty(X)+\epsilon)} \wedge X_1^L \in \mathcal{A}_\delta^L\} &= \sum_{x_1^L \in \mathcal{A}_\delta^L} \sum_{m \geq 2^{L(H_\infty+\epsilon)}} P(x_1^L) Q_m(x_1^L) \\
&\leq \sum_{x_1^L \in \mathcal{A}_\delta^L} P(x_1^L) \sum_{m \geq 2^{L(H_\infty+\epsilon)}} \frac{m Q_m(x_1^L)}{2^{L(H_\infty+\epsilon)}} \\
&\leq \sum_{x_1^L \in \mathcal{A}_\delta^L} \frac{P(x_1^L)}{2^{L(H_\infty+\epsilon)}} \sum_{m=1,2,\cdots} m Q_m(x_1^L) \\
&= \sum_{x_1^L \in \mathcal{A}_\delta^L} \frac{P(x_1^L)}{2^{L(H_\infty+\epsilon)}} T(x_1^L) \\
&\overset{(a)}{=} \sum_{x_1^L \in \mathcal{A}_\delta^L} \frac{1}{2^{L(H_\infty+\epsilon)}} \\
&\overset{(b)}{\leq} \frac{2^{L(H_\infty+\delta)}}{2^{L(H_\infty+\epsilon)}} = 2^{-L\epsilon/2}.
\end{aligned}
$$

Here (a) follows from Kac's theorem (1) and (b) from the cardinality bound (5) for $\mathcal{A}_\delta^L$. Note finally that $\lim_{L\to\infty} 2^{-L\epsilon/2} = 0$.

# Weighting

## Binary sources, sequences




A *sequence* $x^T = x_1 x_2 \cdots x_T$ with components $\in \{0, 1\}$ is produced by the *source* with actual probability $P_a(x^T)$.

*Example:* Independent identically distributed (I.I.D.) source with parameter $\theta$. Let

$$\begin{aligned} P_a(1) &= \theta, \text{ and} \\ P_a(0) &= 1 - \theta, \end{aligned}$$

for some $0 \leq \theta \leq 1$. Then a sequence $x^T$ containing $a$ zeros and $b$ ones has

$$P_a(x^T) = (1 - \theta)^a \theta^b.$$

## Codes, redundancy

A *source code* assigns to source sequence $x^T$ a binary codeword $c(x^T)$ of length $L(x^T)$. These codewords must satisfy the prefix condition.

*Example:* $T = 2$.

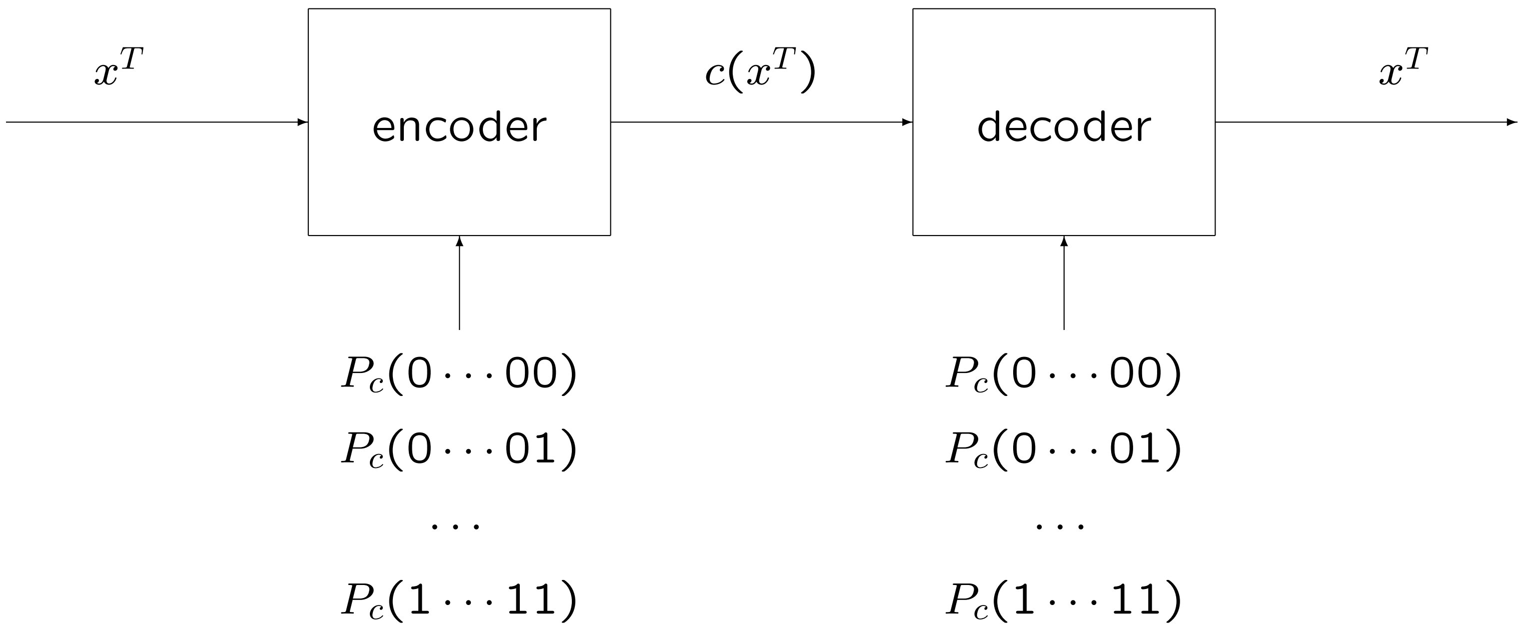
| $x^T$ | $c(x^T)$ | $L(x^T)$ |
|---|---|---|
| 00 | 0 | 1 |
| 01 | 10 | 2 |
| 10 | 110 | 3 |
| 11 | 111 | 3 |

The *individual redundancy* $\rho(x^T)$ of a sequence $x^T$ is now defined as

$$\rho(x^T) = L(x^T) - \log_2 \frac{1}{P_a(x^T)},$$

i.e. codeword-length minus *ideal* codeword-length.

# Arithmetic coding




Arithmetic coding is possible if we use *coding probabilities* $P_c(x^T)$ satisfying

$$P_c(x^T) > 0 \text{ for all } x^T, \text{ and} \sum_{x^T} P_c(x^T) = 1.$$

Now we obtain for the codeword-lengths

$$L(x^T) < \log_2 \frac{1}{P_c(x^T)} + 2.$$

PROBLEM:
How do we choose the coding probabilities $P_c(x^T)$ in the universal case?
We want them to be as large as possible (as close as possible to $P_a(x^T)$).

# I.I.D. source with unknown $\theta$

A good coding probability for a sequence $x^T$ that contains $a$ zeroes and $b$ ones is

$$P_e(a,b) \triangleq \int_{\theta=0,1} \frac{1}{\pi\sqrt{(1-\theta)\theta}} \cdot (1-\theta)^a \theta^b d\theta.$$

(Dirichlet **weighting**, Krichevsky-Trofimov estimator)

Properties:

- Lowerbound

$$\frac{P_c(x^T)}{P_a(x^T)} = \frac{P_e(a,b)}{\theta^a(1-\theta)^b} \geq \frac{1}{2\sqrt{T}}.$$

  for all $\theta$ and $x^T$ with $a$ zeros and $b$ ones.
  LOSS: At most a factor $2\sqrt{T}$.

- Probability of a sequence with $a+1$ zeroes and $b$ ones

$$P_e(a+1,b) = \frac{a+1/2}{a+b+1} \cdot P_e(a,b).$$

  $\Rightarrow$ sequential compression is simple, IMPORTANT!

The individual redundancy

$$\begin{aligned}\rho(x^T) &= L(x^T) - \log_2 \frac{1}{P_a(x^T)} \\ &< \log_2 \frac{1}{P_e(a,b)} + 2 - \log_2 \frac{1}{\theta^a(1-\theta)^b} \\ &= \log_2 \frac{\theta^a(1-\theta)^b}{P_e(a,b)} + 2 \leq \left(\frac{1}{2}\log T + 1\right) + 2.\end{aligned}$$

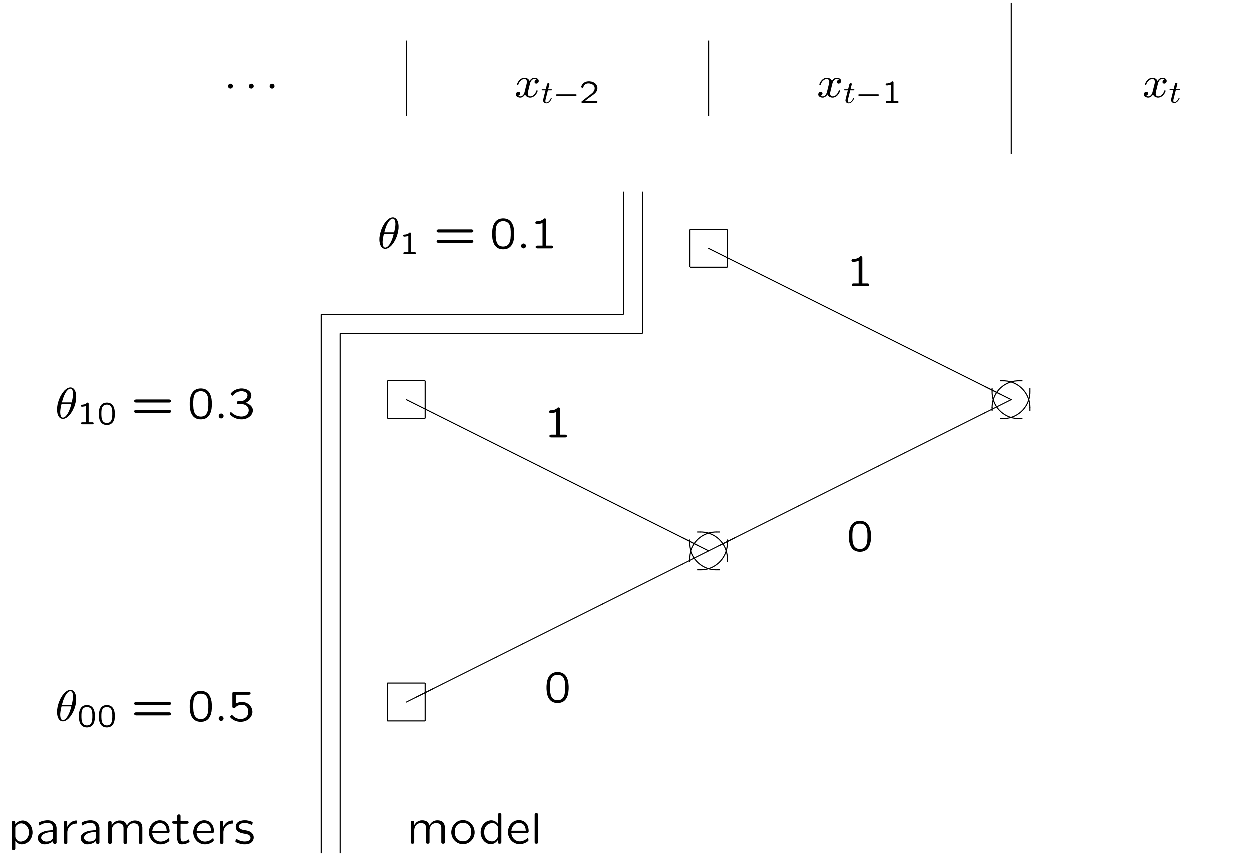
for all $\theta$ and $x^T$ with $a$ zeroes and $b$ ones.
$\Rightarrow$ PARAMETER REDUNDANCY $\leq \frac{1}{2}\log T + 1$ bits.

For the average codeword-length we obtain

$$\begin{aligned}L_{av} &< H(X^T) + \frac{1}{2}\log_2 T + 3, \\ &= T \cdot h(\theta) + \frac{1}{2}\log_2 T + 3.\end{aligned}$$

*Rissanen's lowerbound (1984):* redundancy $\frac{1}{2}\log_2 T$ bits/parameter is asymptotically optimal!

# Binary Tree Sources (Example)




$$
\begin{aligned}
P_a(X_t = 1|\cdots, X_{t-1} = 1) &= 0.1 \\
P_a(X_t = 1|\cdots, X_{t-2} = 1, X_{t-1} = 0) &= 0.3 \\
P_a(X_t = 1|\cdots, X_{t-2} = 0, X_{t-1} = 0) &= 0.5
\end{aligned}
$$

# Problem, Concepts

PROBLEM: What is a good coding distribution for sequences $x^T$ produced by a tree source with

- an unknown tree-model,
- and unknown parameters?

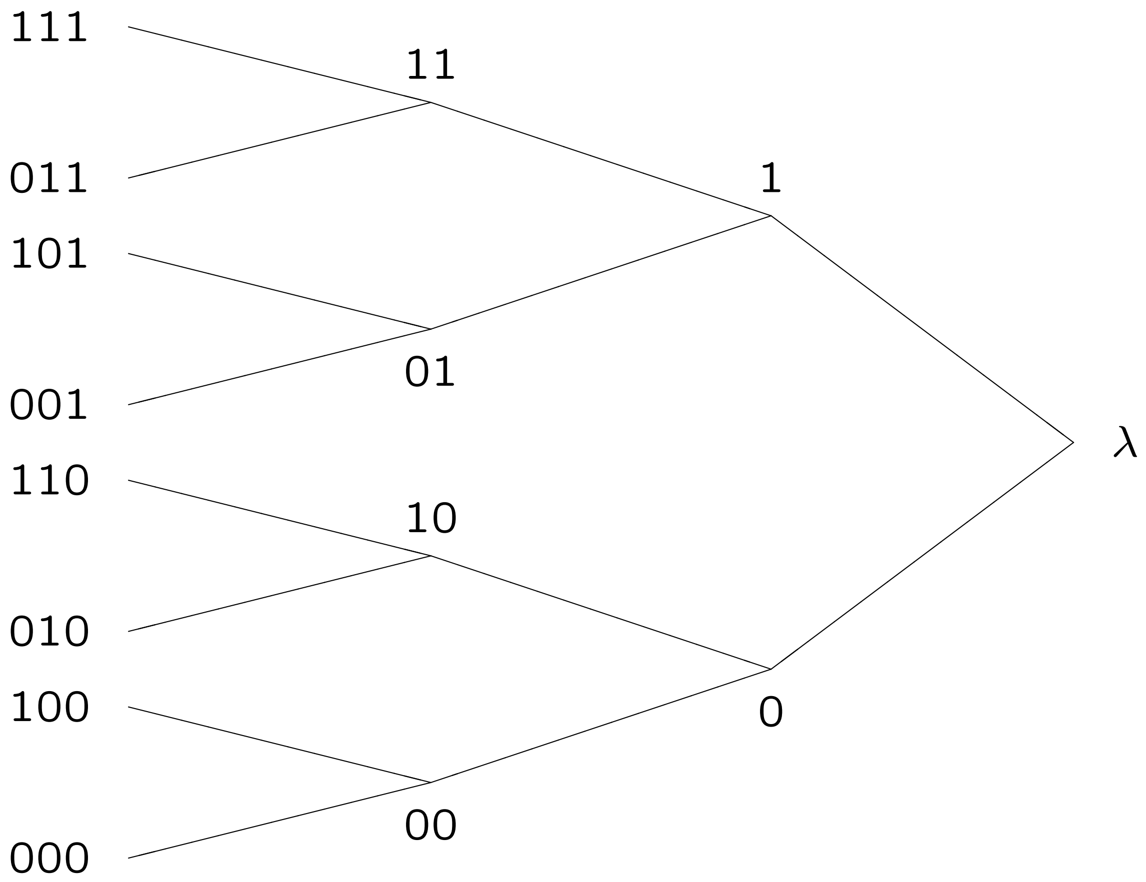
**Context-tree Weighting (Willems, Shtarkov, and Tjalkens [1995]):**

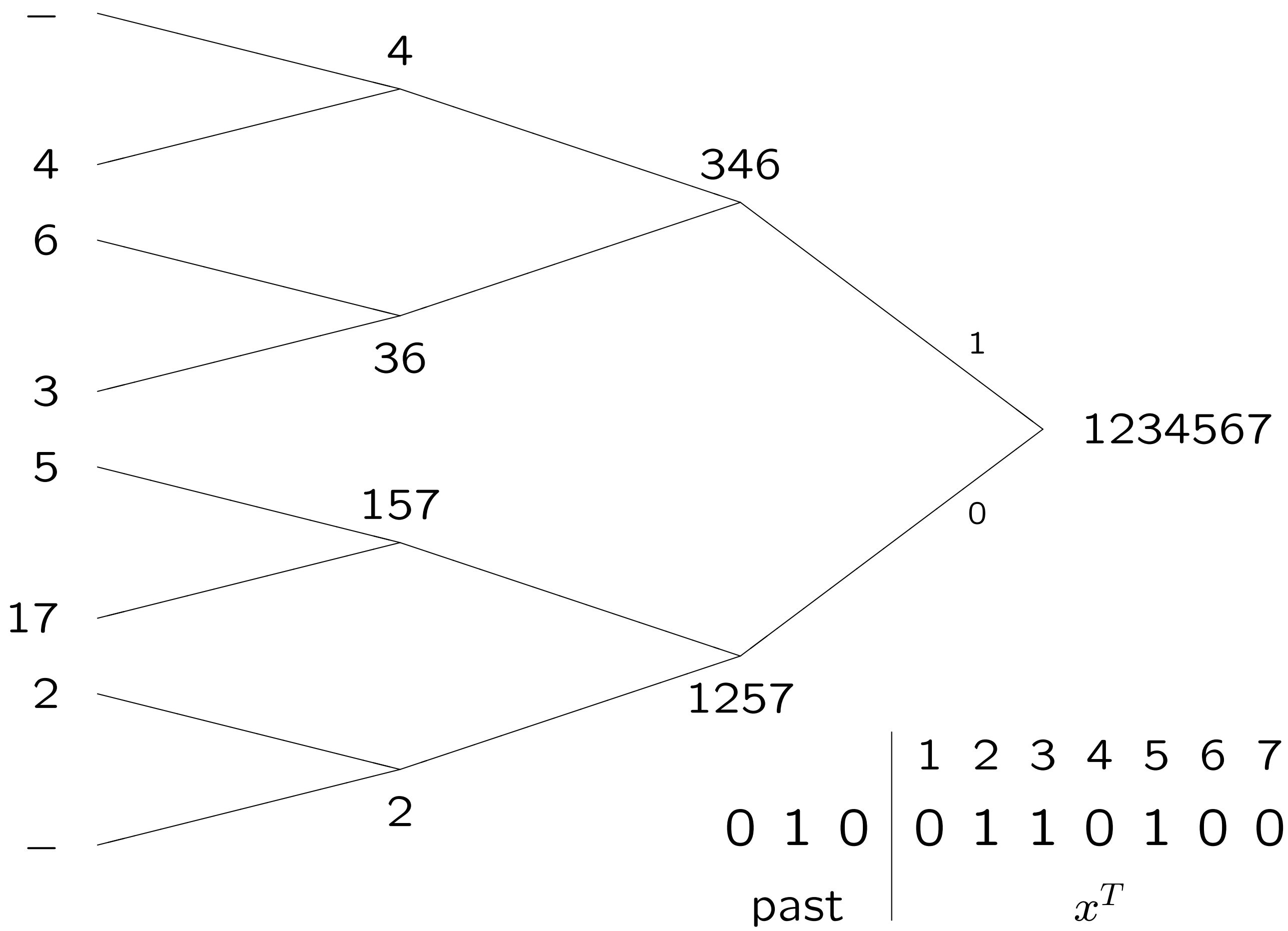
CONCEPTS:

- Context-tree (Rissanen [ ... ]),
- Combining,
- Weighting (folclore).

## Context-Tree

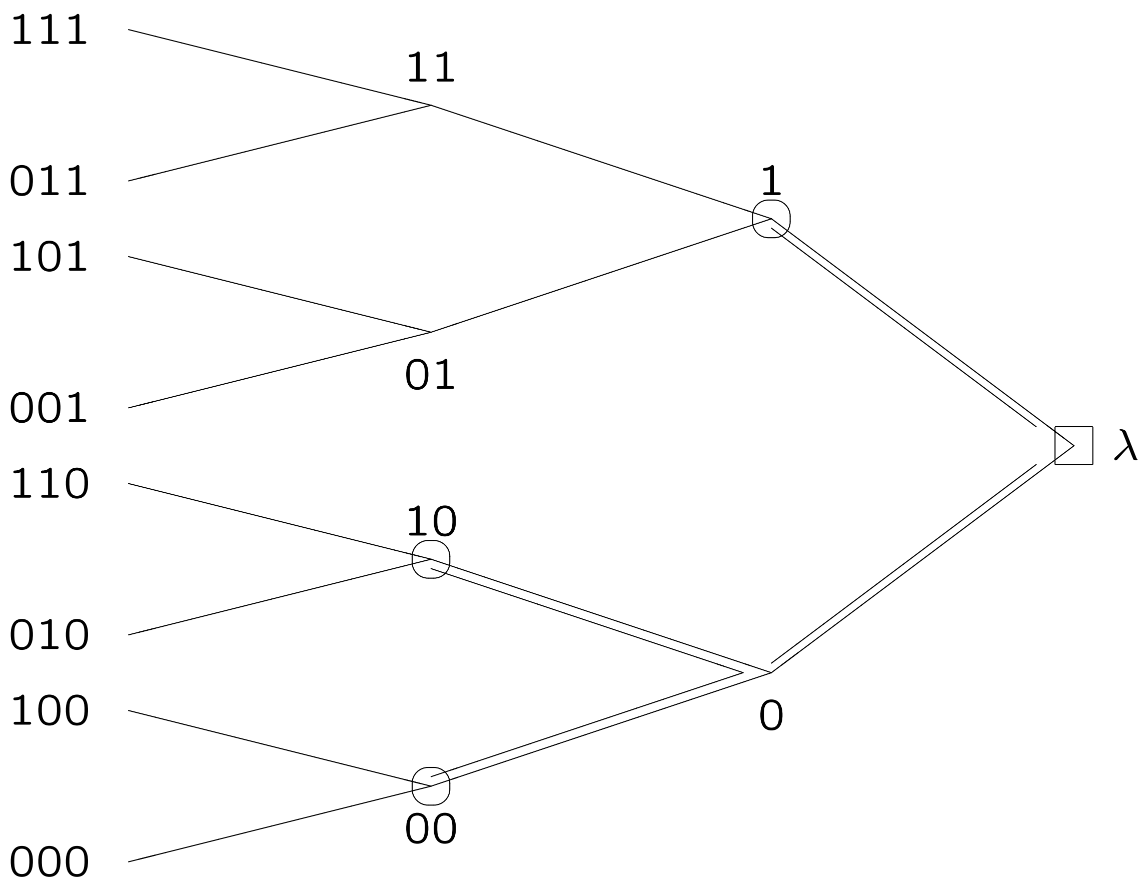
A tree-like data-structure with depth $D$. Node $s$ contains the sequence of source symbols that have occurred following context $s$.






Context-tree splits up sequences in subsequences.

# Leaves of the context-tree




Assume that the actual tree source fits into the context tree.

Then the subsequence corresponding to a leaf $s$ of the context tree is I.I.D.

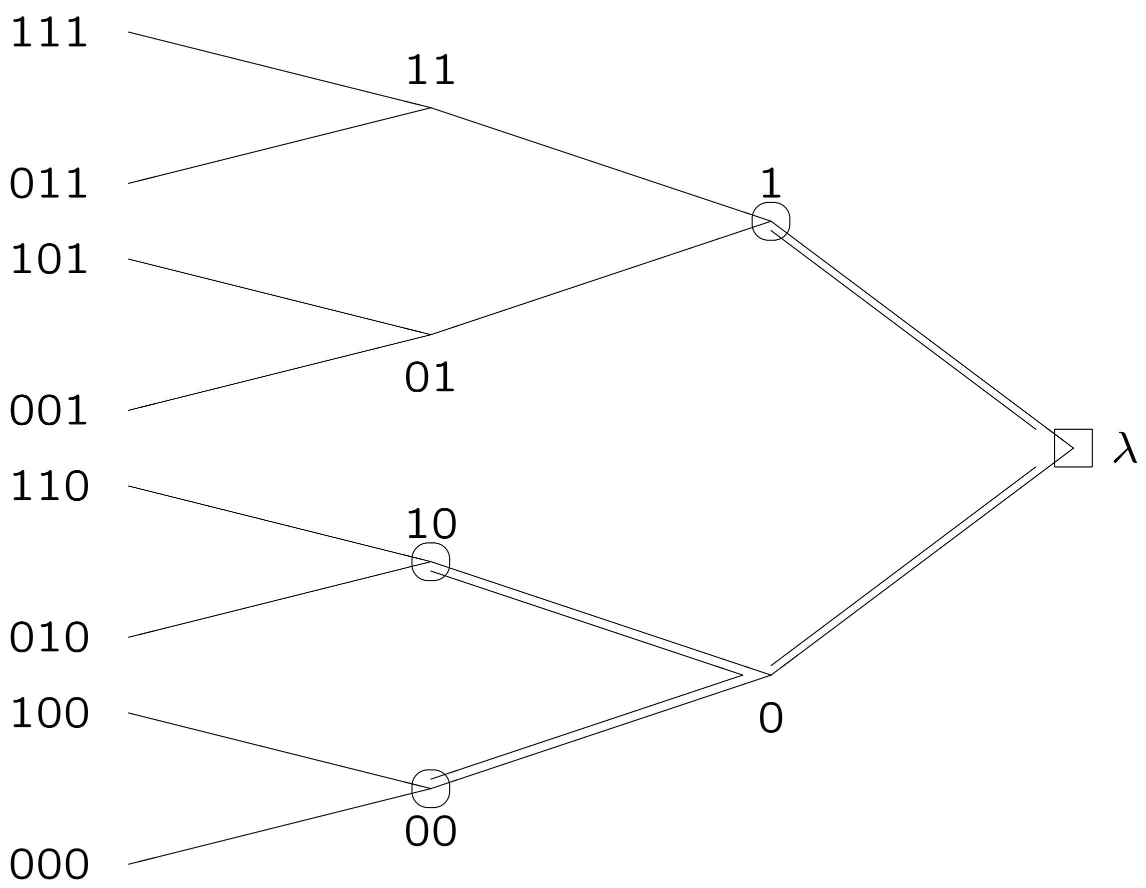
A good coding probability* for this subsequence is therefore

$$P_w^s = P_e(a_s, b_s),$$

where $a_s$ and $b_s$ are the number of zeroes and ones in this subsequence.

*We denote this probability by $P_w^s$ for a reason that will become clear later.

# Internal nodes of the context-tree




The subsequence corresponding to a node $s$ of the context tree is

- I.I.D. if the node $s$ is not an internal node of the actual tree-model,
- a combination of the subsequences corresponding to nodes $0s$ and $1s$, if $s$ is an internal node of the actual model.

## Combining

Suppose that sequence $y = y'y''$ is some combination of two independently generated subsequences $y'$ and $y''$.
Let $P_1(y')$ be a good coding probability for subsequence $y'$ and $P_2(y'')$ be a good coding probability for subsequence $y''$.

Then

$$P_{12}(y'y'') = P_1(y') \cdot P_2(y'').$$

is a good coding probability for $y = y'y''$.

## Weighting

Suppose that at least $P_1(y)$ or $P_2(y)$ is a good coding probability for sequence $y$.

Then the *weighted probability*

$$P_w(y) = \frac{P_1(y) + P_2(y)}{2}$$

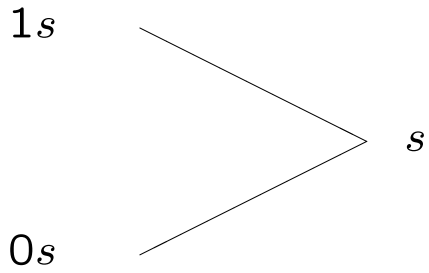
is at least (almost) as good as $P_1(y)$ and $P_2(y)$.

This is true because for $i = 1$ and 2

$$P_w(y) \geq \frac{P_i(y)}{2}.$$

LOSS: At most a factor 2.

# Recursion (internal nodes of context tree)




Suppose that $P_w^{0s}$ and $P_w^{1s}$ are good coding probabilities for the subsequences corresponding to $0s$ and $1s$.
If the subsequence that corresponds to node $s$

- is I.I.D., then a good coding probability for it would be

$$P_e(a_s, b_s).$$

- is a combination of the subsequences corresponding to $0s$ and $1s$, then a good coding probability for it would be

$$P_w^{0s} \cdot P_w^{1s}.$$

Weighting both alternatives yields the coding probability

$$P_w^s = \frac{P_e(a_s, b_s) + P_w^{0s} \cdot P_w^{1s}}{2}$$

for the subsequence that corresponds to node $s$.

Finally we find in the *root* $\lambda$ of the context-tree the coding probability $P_w^\lambda$ for the entire source sequence $x^T$.

IMPORTANT: $P_w^\lambda$ can be computed sequentially. Sequential (one-pass) compression is possible!

# Analysis (Example)

$$\begin{aligned} P_w^\lambda &\geq \frac{1}{2} P_w^0 \cdot P_w^1 \\ &\geq \frac{1}{2}\frac{1}{2} P_w^{00} \cdot P_w^{10} \cdot \frac{1}{2} P_e(a_1, b_1) \\ &\geq \frac{1}{2}\frac{1}{2}\frac{1}{2} P_e(a_{00}, b_{00}) \cdot \frac{1}{2} P_e(a_{10}, b_{10}) \cdot \frac{1}{2} P_e(a_1, b_1). \end{aligned}$$

Moreover

$$\begin{aligned} P_e(a_{00}, b_{00}) &\geq \frac{1}{2\sqrt{a_{00} + b_{00}}} (1 - \theta_{00})^{a_{00}} \theta_{00}^{b_{00}}, \\ P_e(a_{10}, b_{10}) &\geq \frac{1}{2\sqrt{a_{10} + b_{10}}} (1 - \theta_{10})^{a_{10}} \theta_{10}^{b_{10}}, \\ P_e(a_1, b_1) &\geq \frac{1}{2\sqrt{a_1 + b_1}} (1 - \theta_1)^{a_1} \theta_1^{b_1}. \end{aligned}$$

Here

$$P_a(x^T) = (1 - \theta_{00})^{a_{00}} \theta_{00}^{b_{00}} \cdot (1 - \theta_{10})^{a_{10}} \theta_{10}^{b_{10}} \cdot (1 - \theta_1)^{a_1} \theta_1^{b_1}.$$

# Total loss (Example)

- a factor 2 in every leaf and every internal node of the actual tree-model, i.e. $2^5$ in total,
- times a factor*

$$2\sqrt{(a_{00}+b_{00})}\cdot 2\sqrt{(a_{10}+b_{10})}\cdot 2\sqrt{(a_1+b_1)} \leq \left(2\sqrt{\frac{T}{3}}\right)^3.$$

- Hence

$$\frac{P_w^\lambda}{P_a(x^T)} \geq \frac{1}{2^5\cdot(2\sqrt{T/3})^3}.$$

- Total individual redundancy

$$\begin{aligned}\rho(x^T) = L(x^T) - \log_2\frac{1}{P_a(x^T)} &< \log_2\frac{1}{P_w^\lambda} + 2 - \log_2\frac{1}{P_a(x^T)} \\ &\leq 5 + 3\left(\frac{1}{2}\log_2\frac{T}{3}+1\right)+2.\end{aligned}$$

for all $(\theta_{00},\theta_{10},\theta_1)$ and all $x^T$.

*For simplicity assume that $a_s + b_s > 0$ for all leaves $s$ of the actual source.

# In general

For a tree source $\mathcal{S}$ with $|\mathcal{S}|$ leaves (parameters) the loss is

- a factor $2^{2|\mathcal{S}|-1}$
- times a factor $\left(2\sqrt{\frac{T}{|\mathcal{S}|}}\right)^{|\mathcal{S}|}$.

TOTAL REDUNDANCY:

$$\rho(x^T) < 2|\mathcal{S}| - 1 + \left(\frac{|\mathcal{S}|}{2}\log_2\frac{T}{|\mathcal{S}|} + |\mathcal{S}|\right) + 2 \text{ bits},$$

subdivided into three terms:

1. MODEL REDUNDANCY: $\leq 2|\mathcal{S}| - 1$,
2. PARAMETER REDUNDANCY: $\leq \frac{|\mathcal{S}|}{2}\log_2\frac{T}{|\mathcal{S}|} + |\mathcal{S}|$,
3. and CODING REDUNDANCY: $< 2$.

# Basic property the CTW method

- Implements a "weighting" over all tree-models with depth not exceeding $D$, i.e.

$$P_w^\lambda = \sum_{\mathcal{S} \in \mathcal{T_D}} P(\mathcal{S}) P_e(x^T|\mathcal{S}),$$

with

$$P_e(x^T|\mathcal{S}) = \Pi_{s \in \mathcal{S}} P_e(a_s, b_s),$$

and *a priori* tree-model probability

$$P(\mathcal{S}) = 2^{-(2|\mathcal{S}|-1)}.$$

- This leads to optimal redundancy behavior in individual sense.

- Straightforward analysis.

## Simulation (Example)

A sequence $x_1, x_2, x_3, \cdots$ is generated by a tree source with a certain model.

We now compute the terms $P(\mathcal{S})P_e(x^t|\mathcal{S})$ in the CTW-weighting for several models and $t = 1, 2, \cdots$. We plot

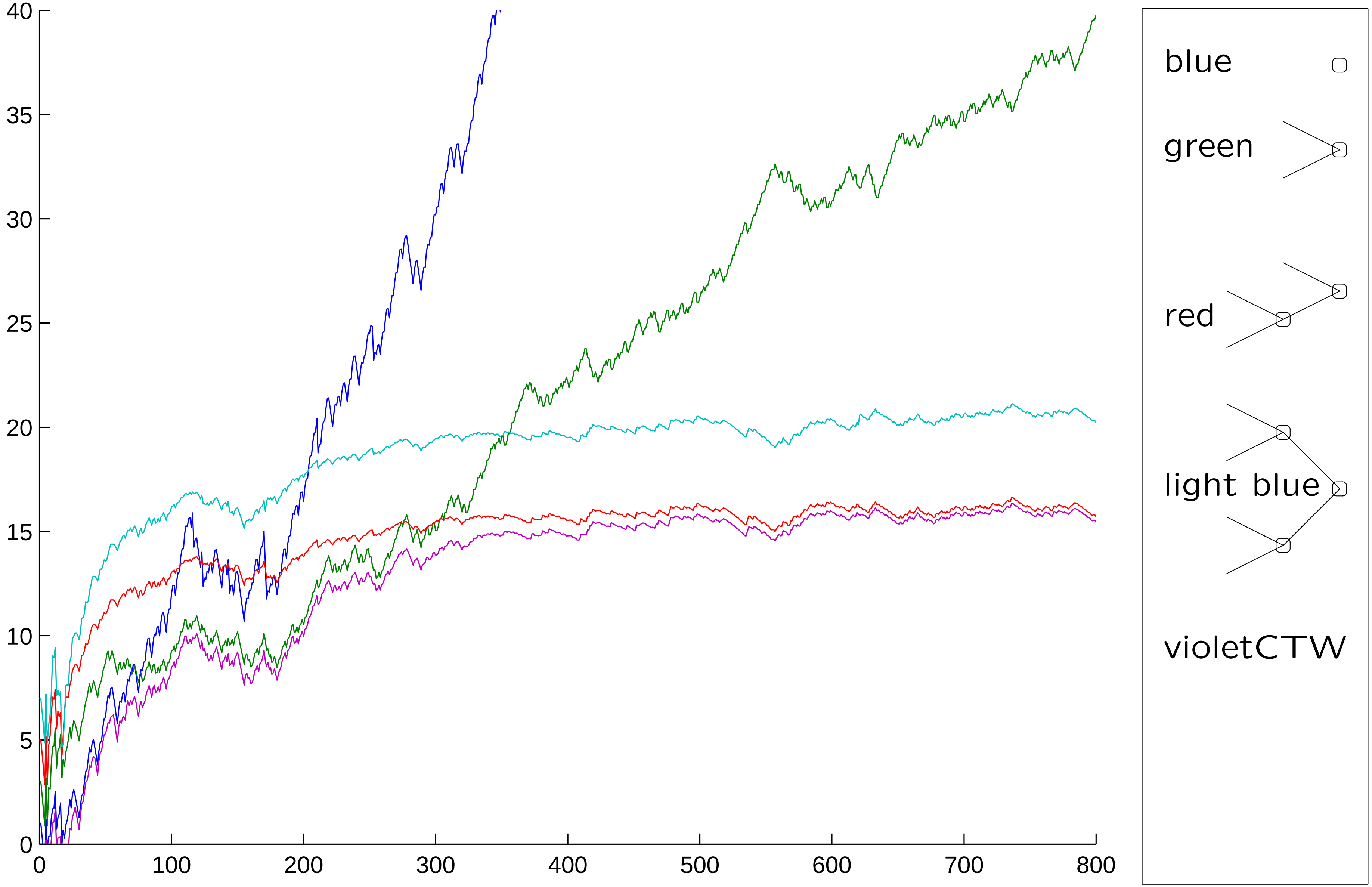
$$\log_2 \frac{1}{P(\mathcal{S})P_e(x^t|\mathcal{S})} - \log_2 \frac{1}{P_a(x^t)}.$$

We also compute the CTW-probability $P_w^\lambda$ and plot

$$\log_2 \frac{1}{P_w^\lambda} - \log_2 \frac{1}{P_a(x^t)}.$$

Then the actual model does not always contribute the most. The CTW-method always follows the model that gives the largest contribution!

However for $t \to \infty$ the actual model gives the largest contribution.



0
5
10
15
20
25
30
35
40
0
100
200
300
400
500
600
700
800
blue
green
red
light blue
violetCTW

## Conclusion

We have discussed Waiting and Weighting, which turned out to be useful concepts in Universal Source Coding.