High order entropy

Ben Langmead

JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

Department of Computer Science

High-order entropy

Zero order empirical entropy seems insufficient when context matters

Bigram frequency per 40,000 words

th	1.52	**en**	0.55	**ng**	0.18
he	1.28	**ed**	0.53	**of**	0.16
in	0.94	**to**	0.52	**al**	0.09
er	0.94	**it**	0.50	**de**	0.09
an	0.82	**ou**	0.50	**se**	0.08
re	0.68	**ea**	0.47	**le**	0.08
nd	0.63	**hi**	0.46	**sa**	0.06
at	0.59	**is**	0.46	**si**	0.05
on	0.57	**or**	0.43	**ar**	0.04
nt	0.56	**ti**	0.34	**ve**	0.04
ha	0.56	**as**	0.33	**ra**	0.04
es	0.56	**te**	0.27	**ld**	0.02
st	0.55	**et**	0.19	**ur**	0.02

http://math.cornell.edu/~mec/2003-2004/cryptography/subs/digraphs.html

High-order entropy

Can compress better if we consider ***context***

Let C change depending on surrounding symbols

gtgtatcggagcgctctgcgttatcgatcgcgatctggt

$C_{\mathtt{tcg}}(\mathtt{a})$ $C_{\mathtt{gcg}}(\mathtt{a})$ $C_{\mathtt{tcg}}(\mathtt{a})$ $C_{\mathtt{tcg}}(\mathtt{a})$ $C_{\mathtt{tct}}(\mathtt{a})$

For k symbols of context, we have codes $C_i \in \{ C_{\Sigma^k} \}$

High-order entropy

Could consider context to the right $\rightarrow$

gtgtatcggagcgctctgcgttatcgatcgcgatctggta

$C_{\mathtt{tcg}}(\mathrm{a})$ $C_{\mathtt{gcg}}(\mathrm{a})$ $C_{\mathtt{tcg}}(\mathrm{a})$ $C_{\mathtt{tcg}}(\mathrm{a})$ $C_{\mathtt{tct}}(\mathrm{a})$

Or context to the left $\leftarrow$

gtgtatcggagcgctctgcgttatcgatcgcgatctggta

$C_{\mathtt{tct}}(\mathrm{a})$ $C_{\mathtt{cgg}}(\mathrm{a})$ $C_{\mathtt{gtt}}(\mathrm{a})$ $C_{\mathtt{tcg}}(\mathrm{a})$ $C_{\mathtt{gcg}}(\mathrm{a})$

High-order entropy

How should we build each code $\{C_{\Sigma^k}\}$?

Same as before, but with frequencies *conditioned on context*

E.g. $C_{\mathtt{gca}}$ is built considering the number of times each symbol occurs just after $\mathtt{gca}$

High-order entropy

abracadabraabracadabra

Let $S_{\mathtt{a}}$ be the substring we get by concatenating characters just after the a's

$$S_{\mathtt{a}} = \mathtt{bcdbabcdb}$$

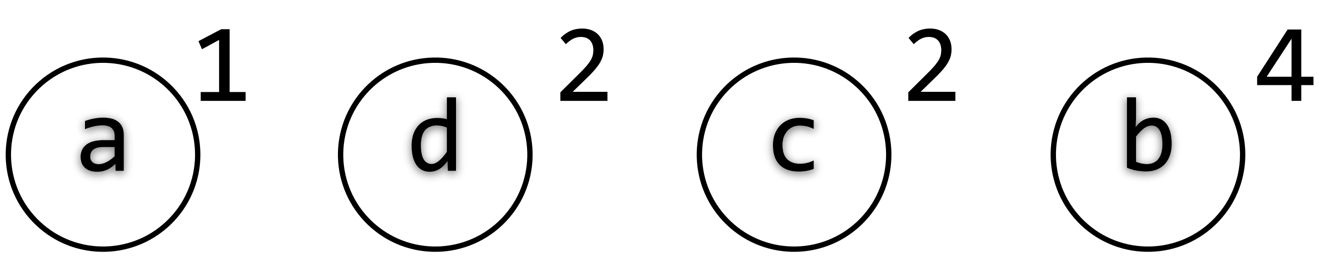
Build code $C_{\mathtt{a}}$ using frequencies in S_a:

$$\{\mathtt{a}:1,\ \mathtt{b}:4,\ \mathtt{c}:2,\ \mathtt{d}:2,\ \mathtt{r}:0\}$$

(r won't get a code)

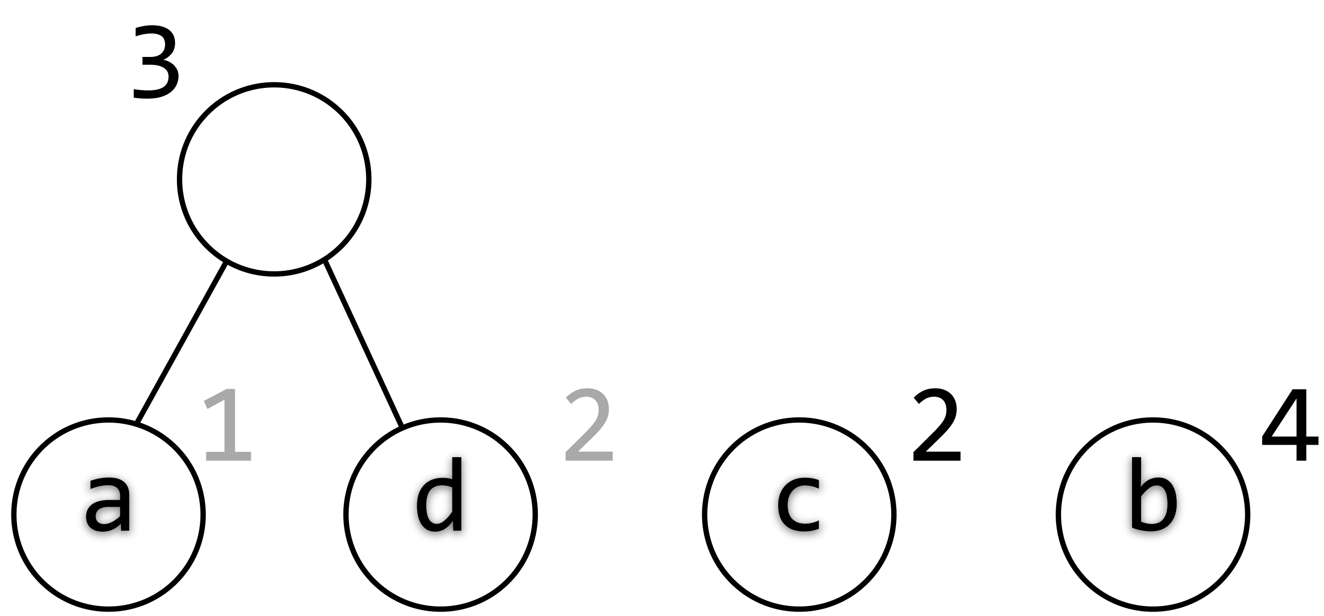
High-order entropy

$$\{\texttt{a} : 1,\ \texttt{b} : 4,\ \texttt{c} : 2,\ \texttt{d} : 2,\ \texttt{r} : 0\}$$



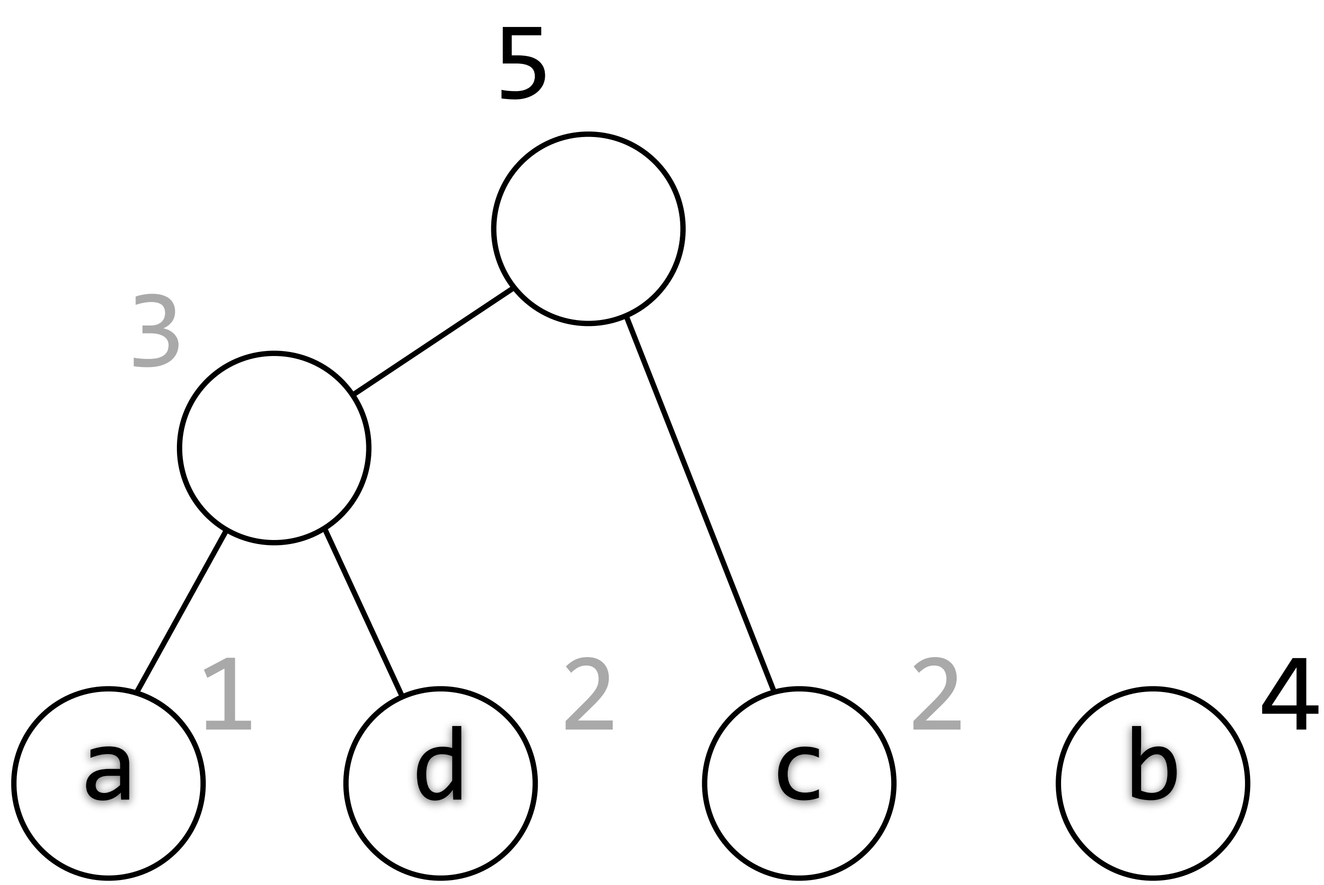
High-order entropy

$$\{\texttt{a} : 1,\ \texttt{b} : 4,\ \texttt{c} : 2,\ \texttt{d} : 2,\ \texttt{r} : 0\}$$



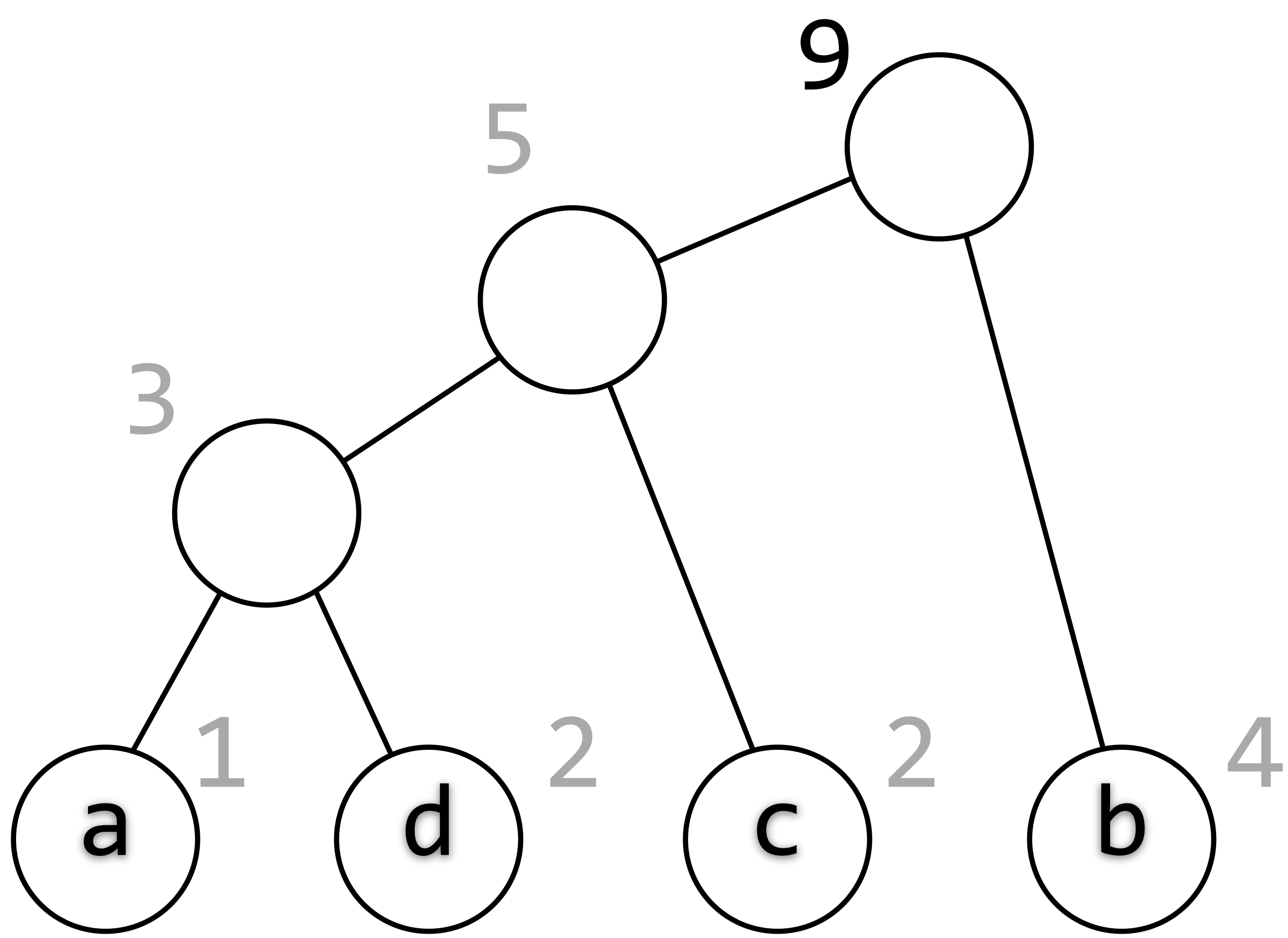
High-order entropy

$$\{\texttt{a} : 1,\ \texttt{b} : 4,\ \texttt{c} : 2,\ \texttt{d} : 2,\ \texttt{r} : 0\}$$



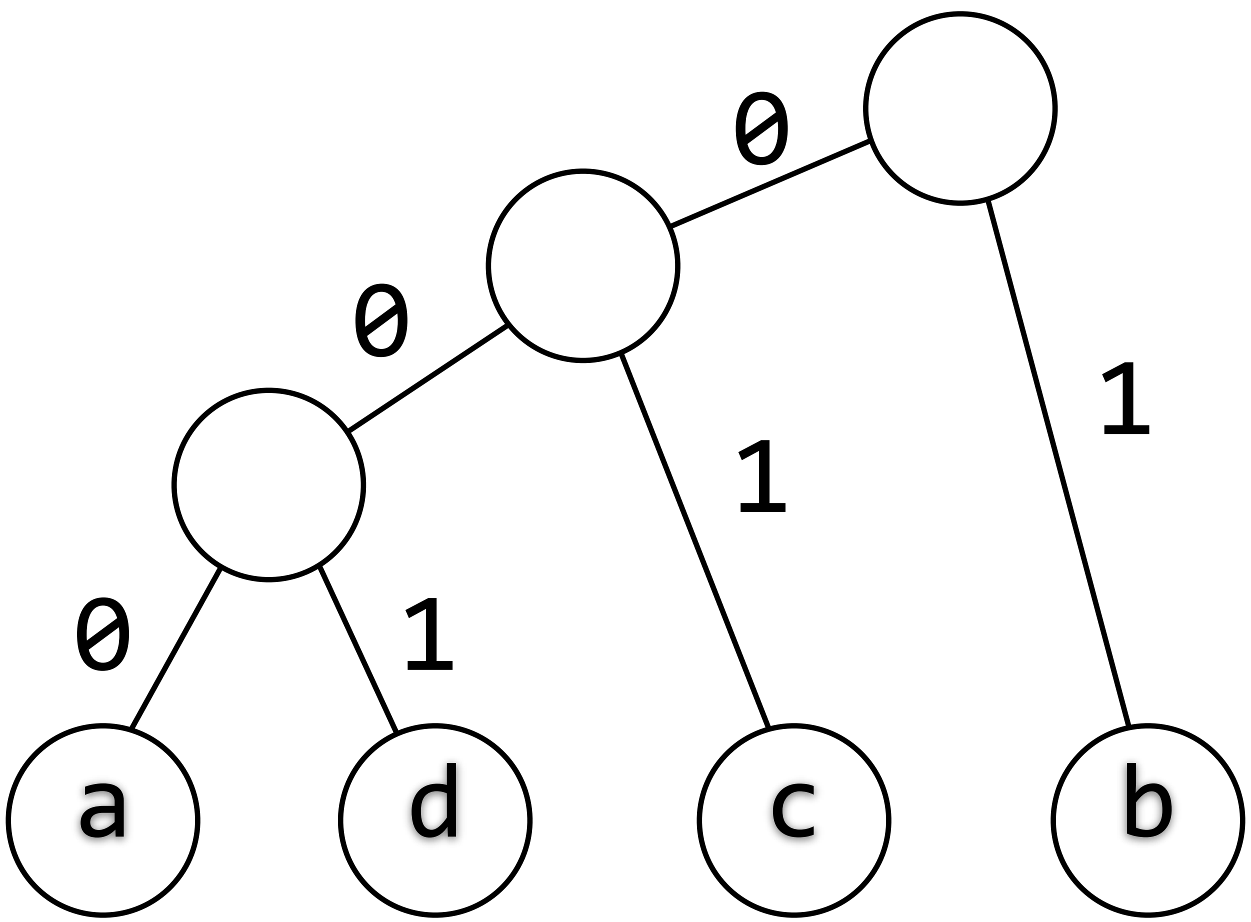
High-order entropy

$$\{\texttt{a}:1,\ \texttt{b}:4,\ \texttt{c}:2,\ \texttt{d}:2,\ \texttt{r}:0\}$$



High-order entropy

$$\{\texttt{a} : 1,\ \texttt{b} : 4,\ \texttt{c} : 2,\ \texttt{d} : 2,\ \texttt{r} : 0\}$$



$$C_{\texttt{a}}(\texttt{a}) = \texttt{000}$$

$$C_{\texttt{a}}(\texttt{d}) = \texttt{001}$$

$$C_{\texttt{a}}(\texttt{c}) = \texttt{01}$$

$$C_{\texttt{a}}(\texttt{b}) = \texttt{1}$$

High-order entropy

abracadabraabracadabra

$S_{\mathtt{a}} = \mathtt{bcdbabcdb}$

$\{\mathtt{a} : 1,\ \mathtt{b} : 4,\ \mathtt{c} : 2,\ \mathtt{d} : 2\}$

$C_{\mathtt{a}}(\mathtt{a}) = 000$

$C_{\mathtt{a}}(\mathtt{d}) = 001$

$C_{\mathtt{a}}(\mathtt{c}) = 01$

$C_{\mathtt{a}}(\mathtt{b}) = 1$

$S_{\mathtt{b}} = \mathtt{rrrr}\quad \{\mathtt{r} : 4\}$

(no code)

$S_{\mathtt{c}} = \mathtt{aa}\quad \{\mathtt{a} : 2\}$

(no code)

$S_{\mathtt{d}} = \mathtt{aa}\quad \{\mathtt{a} : 2\}$

(no code)

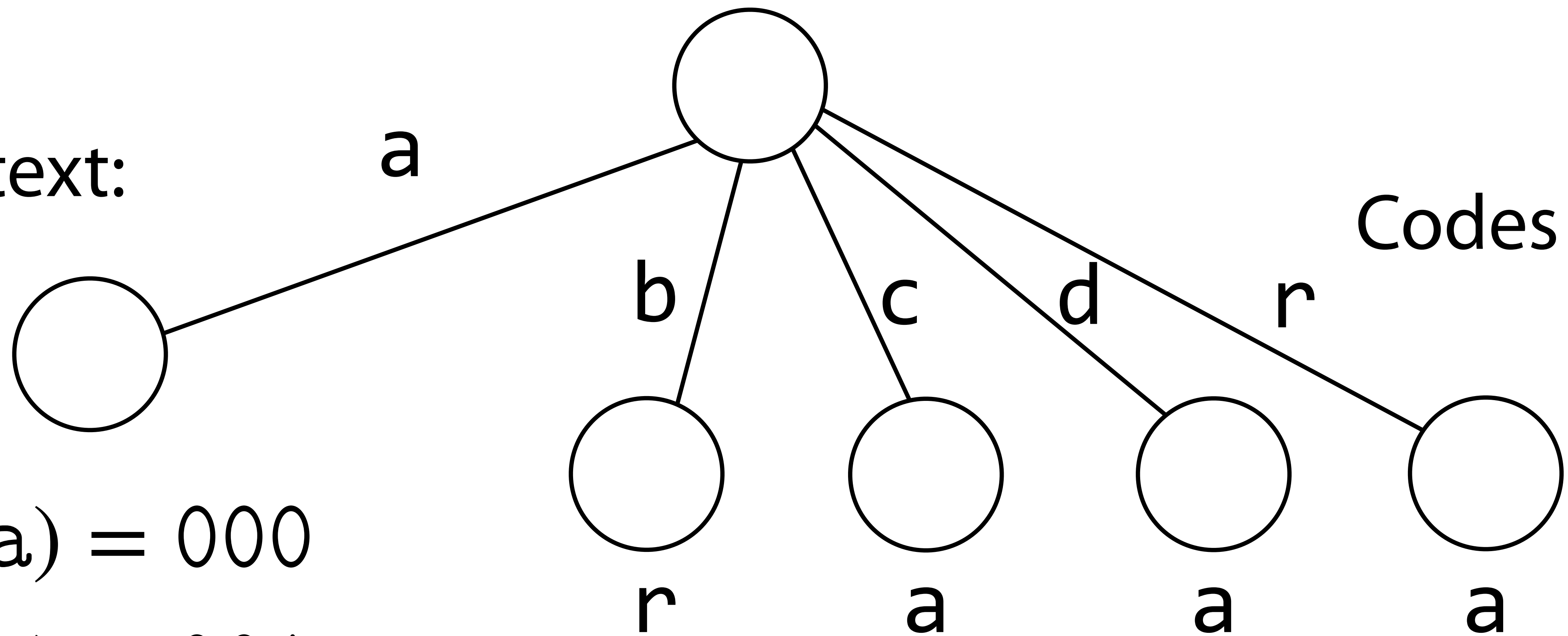
$S_{\mathtt{r}} = \mathtt{aaaa}\quad \{\mathtt{a} : 4\}$

(no code)

High-order entropy

`abracadabraabracadabra`

Context:



Codes based on H_0

(no codes required)

$C_{\mathtt{a}}(\mathtt{a}) = \mathtt{000}$

$C_{\mathtt{a}}(\mathtt{b}) = \mathtt{001}$

$C_{\mathtt{a}}(\mathtt{c}) = \mathtt{01}$

$C_{\mathtt{a}}(\mathtt{d}) = \mathtt{1}$

High-order entropy

`mississippimississippi`

$S_{\mathtt{i}} = \mathtt{sspmssp}$

$\{\mathtt{s}:4,\ \mathtt{p}:2,\ \mathtt{m}:1\}$

$C_{\mathtt{i}}(\mathtt{p}) = \mathtt{00}$

$C_{\mathtt{i}}(\mathtt{m}) = \mathtt{01}$

$C_{\mathtt{i}}(\mathtt{s}) = \mathtt{1}$

$S_{\mathtt{m}} = \mathtt{ii}$ $\{\mathtt{i}:2\}$

(no code)

$S_{\mathtt{p}} = \mathtt{pipi}$

$\{\mathtt{p}:2,\ \mathtt{i}:2\}$

$C_{\mathtt{p}}(\mathtt{p}) = \mathtt{0}$

$C_{\mathtt{p}}(\mathtt{i}) = \mathtt{1}$

$S_{\mathtt{s}} = \mathtt{sisisisi}$

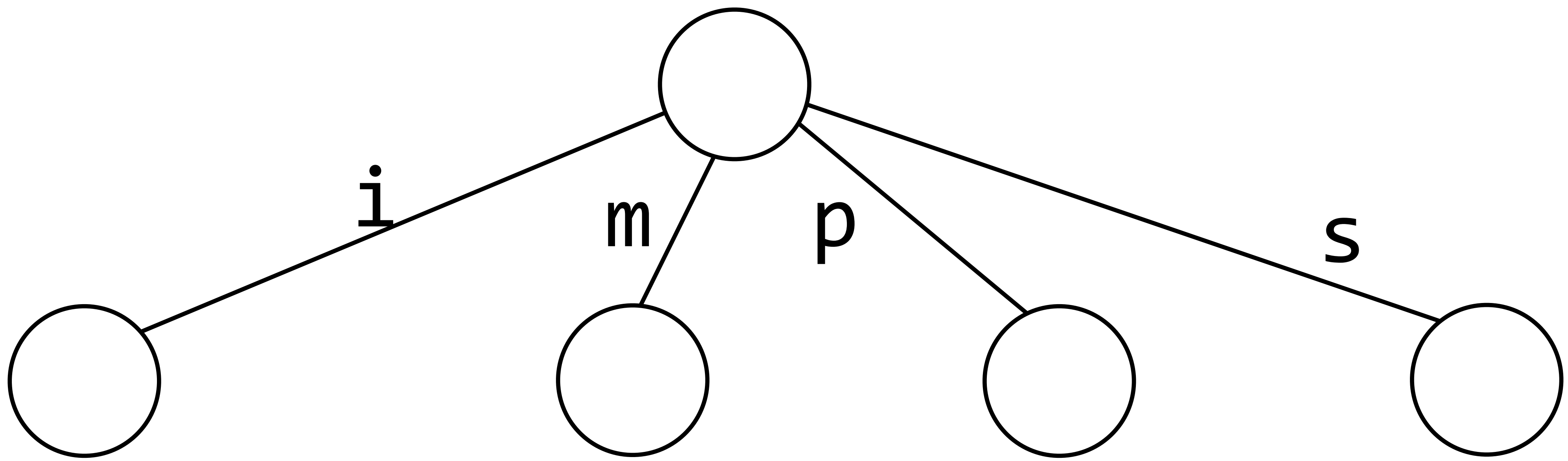
$\{\mathtt{s}:4,\ \mathtt{i}:4\}$

$C_{\mathtt{s}}(\mathtt{s}) = \mathtt{0}$

$C_{\mathtt{s}}(\mathtt{i}) = \mathtt{1}$

High-order entropy

`mississippimississippi`



$C_{\mathtt{i}}(\mathtt{p}) = \mathtt{00}$

$C_{\mathtt{i}}(\mathtt{m}) = \mathtt{01}$

$C_{\mathtt{i}}(\mathtt{s}) = \mathtt{1}$

i

(no code)

$C_{\mathtt{p}}(\mathtt{p}) = \mathtt{0}$

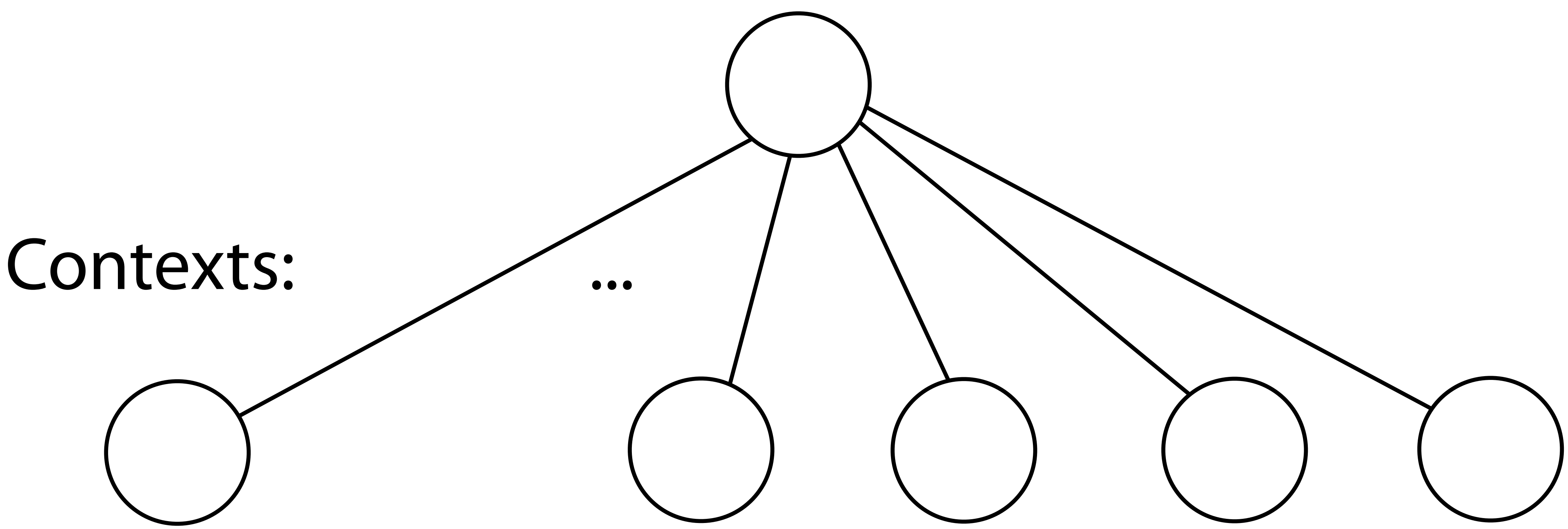
$C_{\mathtt{p}}(\mathtt{i}) = \mathtt{1}$

$C_{\mathtt{s}}(\mathtt{s}) = \mathtt{0}$

$C_{\mathtt{s}}(\mathtt{i}) = \mathtt{1}$

High-order entropy

Def'n of ***high-order empirical entropy*** H_k is similarly hierarchical



Codes achieving near-H_0 given context

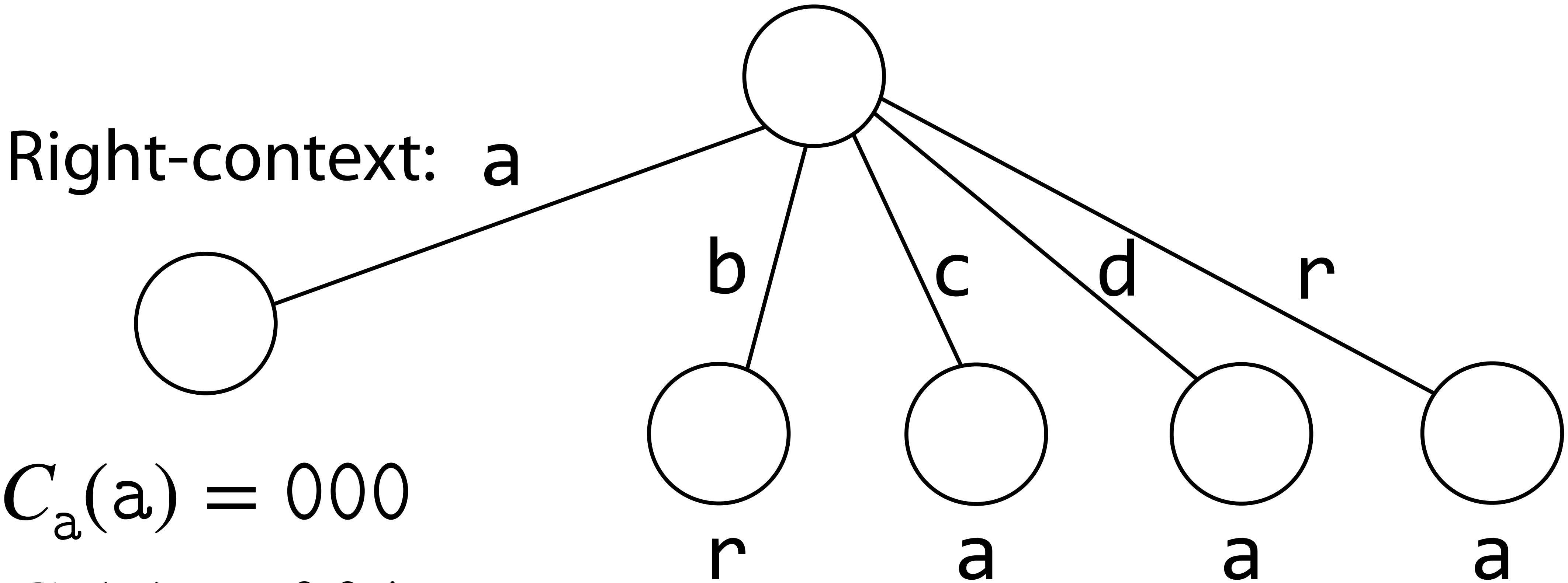
High-order entropy

H_k of a length-n string S is a weighted sum over all contexts of the zero order empirical entropy of symbols having that context

$$H_k(S) = \sum_{t \in \Sigma^k} \frac{|S_t|}{n} \cdot H_0(S_t) \qquad \text{for } k > 0$$

S is the entire string, S_t is the concatenation of symbols having context t

High-order entropy



$C_{\mathtt{a}}(\mathtt{a}) = \mathtt{000}$

$C_{\mathtt{a}}(\mathtt{d}) = \mathtt{001}$

$C_{\mathtt{a}}(\mathtt{c}) = \mathtt{01}$

$C_{\mathtt{a}}(\mathtt{b}) = \mathtt{1}$

Schemes like this can compress to $\leq n(H_k(S) + 1)$ bits

With added overhead of switching between many codes

High-order entropy

Collection	H0	H1	H2
CODE SOURCES	5.537 (69.21%)	4.038 (50.48%)	3.012 (37.65%)
MIDI	5.633 (70.41%)	4.734 (59.18%)	4.139 (51.74%)
PROTEINS	4.195 (52.44%)	4.173 (52.16%)	4.146 (51.82%)
DNA	1.982 (24.78%)	1.935 (24.19%)	1.925 (24.06%)
ENGLISH	4.529 (56.61%)	3.606 (45.08%)	2.922 (36.53%)
XML	5.230 (65.37%)	3.294 (41.17%)	2.007 (25.09%)

Empirical entropies for 6 texts. Values are bits-per-symbol, percentages are ratios compared to ASCII.

Table from: Prezza, Nicola. Compressed Computation for Text Indexing. Diss. PhD thesis, University of Udine, 2016.

High-order entropy

H_k encoding reaches into the string, extracting "structure" needed to compress well

k balances compression with overhead

Grouping principle at play

H_0-based methods are simpler, faster, require less memory, but can't find as much structure as H_k

...or...can they? *Order* to the rescue