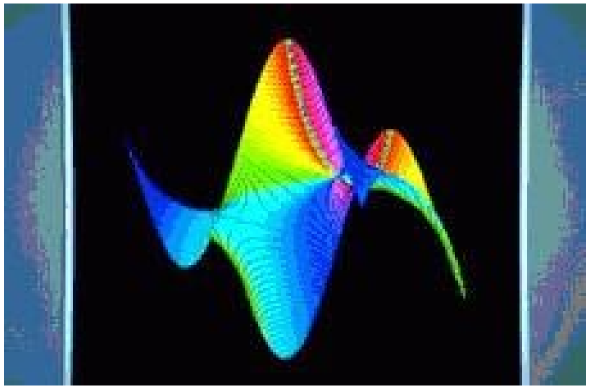
Ubiquitous Pattern Matching and Its Applications (Biology, Security, Multimedia)*

W. Szpankowski†
Department of Computer Science
Purdue University
W. Lafayette, IN 47907

September 14, 2005



*This research is supported by NSF and NIH.

†Joint work with P. Flajolet, A. Grama, R. Gwadera, S. Lonardi, M. Regnier, B. Vallee, M. Ward.

Outline of the Talk

1. Pattern Matching Problems
   - String Matching
   - Subsequence Matching (Hidden Words)
   - Self-Repetitive Pattern Matching
2. Biology – String Matching
   - Analysis (Languages and Generating Functions)
   - Finding Weak Signals and Artifacts in DNA
3. Information Security – Subsequence Matching
   - Some Theory (De Bruijn Automaton)
   - Reliable Threshold in Intrusion Detection
4. Multimedia Compression — Self-Repetitive Matching
   - Theoretical Foundation (Renyi's Entropy)
   - Data Structures and Algorithms
   - Video Compression (**Demo**)
   - Error Resilient LZ'77 (Suffix Trees)

Pattern Matching

Let $\mathcal{W}$ and T be (set of) strings generated over a finite alphabet $\mathcal{A}$.

We call $\mathcal{W}$ the pattern and T the text. The text T is of length n and is generated by a probabilistic source.

We shall write

$$T_m^n = T_m \dots T_n.$$

The pattern $\mathcal{W}$ can be a single string

$$\mathcal{W} = w_1 \dots w_m, \quad w_i \in \mathcal{A}$$

or a set of strings

$$\mathcal{W} = \{\mathcal{W}_1, \dots, \mathcal{W}_d\}$$

with $\mathcal{W}_i \in \mathcal{A}^{m_i}$ being a set of strings of length m_i.

Basic Parameters

Two basic questions are:

- **how many times $\mathcal{W}$ occurs in T**,
- **how long one has to wait until $\mathcal{W}$ occurs in T**.

The following quantities are of interest:

$O_n(\mathcal{W})$ — the number of times $\mathcal{W}$ occurs in T:

$$O_n(\mathcal{W}) = \#\{i : \; T_{i-m+1}^{i} = \mathcal{W}, \; m \leq i \leq n\}.$$

$W_{\mathcal{W}}$ — the first time $\mathcal{W}$ occurs in T:

$$W_{\mathcal{W}} := \min\{n : \; T_{n-m+1}^{n} = \mathcal{W}\}.$$

Relationship:

$$W_{\mathcal{W}} > n \quad \Leftrightarrow \quad O_n(\mathcal{W}) = 0.$$

Various Pattern Matching

(Exact) String Matching

In the exact string matching the pattern $\mathcal{W} = w_1 \dots w_m$ is a given string (i.e., consecutive sequence of symbols).

Generalized String Matching

In the generalized pattern matching a set of patterns (rather than a single pattern) is given, that is,

$$\mathcal{W} = (\mathcal{W}_0, \mathcal{W}_1, \dots, \mathcal{W}_d), \quad \mathcal{W}_i \in \mathcal{A}^{m_i}$$

where $\mathcal{W}_i$ itself for $i \geq 1$ is a subset of $\mathcal{A}^{m_i}$ (i.e., a set of words of a given length m_i).
The set $\mathcal{W}_0$ is called the forbidden set.

Three cases to be considered:

$\mathcal{W}_0 = \emptyset$ — one is interested in the number of patterns from $\mathcal{W}$ occurring in the text.

$\mathcal{W}_0 \neq \emptyset$ — we study the number of $\mathcal{W}_i$, $i \geq 1$ pattern occurrences under the condition that no pattern from $\mathcal{W}_0$ occurs in the text.

$\mathcal{W}_i = \emptyset$, $i \geq 1$, $\mathcal{W}_0 \neq \emptyset$ — restricted pattern matching.

Pattern Matching Problems

Hidden Words or Subsequence Pattern Matching

In this case we search in text for a subsequence $\mathcal{W} = w_1 \ldots w_m$ rather than a string, that is, we look for indices $1 \leq i_1 < i_2 < \cdots < i_m \leq n$ such that

$$T_{i_1} = w_1, \; T_{i_2} = w_2, \cdots, \; T_{i_m} = w_m.$$

We also say that the word $\mathcal{W}$ is "hidden" in the text.

For example:

$$\begin{aligned} \mathcal{W} &= \textbf{date} \\ T &= \textbf{hidden pattern} \end{aligned}$$

occurs four times as a subsequence in the text as hidden pattern but not even once as a string.

Self-Repetitive Pattern Matching

In this case the pattern $\mathcal{W}$ is part of the text:

$$\mathcal{W} = T_1^m.$$

We may ask when the first m symbols of the text will occur again. This is important in Lempel-Ziv like compression algorithms.

Example

Let $T = bababababb$, and $\mathcal{W} = abab$.

- $\mathcal{W}$ occurs **exactly three** times as a **string** at positions $\{2, 4, 6\}$

$$b\underbrace{abab}ababb.$$

- If $\mathcal{W} = \{abab, babb\}$, then $\mathcal{W}$ occurs four times.

$$bababababb.$$

- $\mathcal{W} = abab$ occurs **many** times as a **subsequence**.
  Here is one subsequence occurrence:

$$bababababb.$$

- $\mathcal{W}$ occurs first time at position 2, i.e., $W_{\mathcal{W}} = 2$:

$$bababababb.$$

- $\mathcal{W} = T_1T_2T_3 = bab$ occurs again (repeats itself) at position 5

$$bababababb.$$

Probabilistic Sources

Throughout the talk I will assume that the text is generated by a random source.

Memoryless Source

The text is a realization of an independently, identically distributed sequence of random variables (i.i.d.), such that a symbol $s \in \mathcal{A}$ occurs with probability $P(s)$.

Markovian Source

The text is a realization of a stationary Markov sequence of order K, that is, probability of the next symbol occurrence depends on K previous symbols.

Basic Thrust of our Approach

When searching for over-represented or under-represented patterns we must assure that such a pattern is not generated by randomness itself (to avoid too many false positives).

Outline of the Talk

1. Pattern Matching Problems
2. Biology – String Matching
   - Analysis (Languages and Generating Functions)
   - Finding Weak Signals and Artifacts in DNA
3. Information Security
4. Multimedia Compression

Application – Biology

Biological world is highly stochastic in its behavior and inhomogeneous or non-stationary (S. Salzberg).

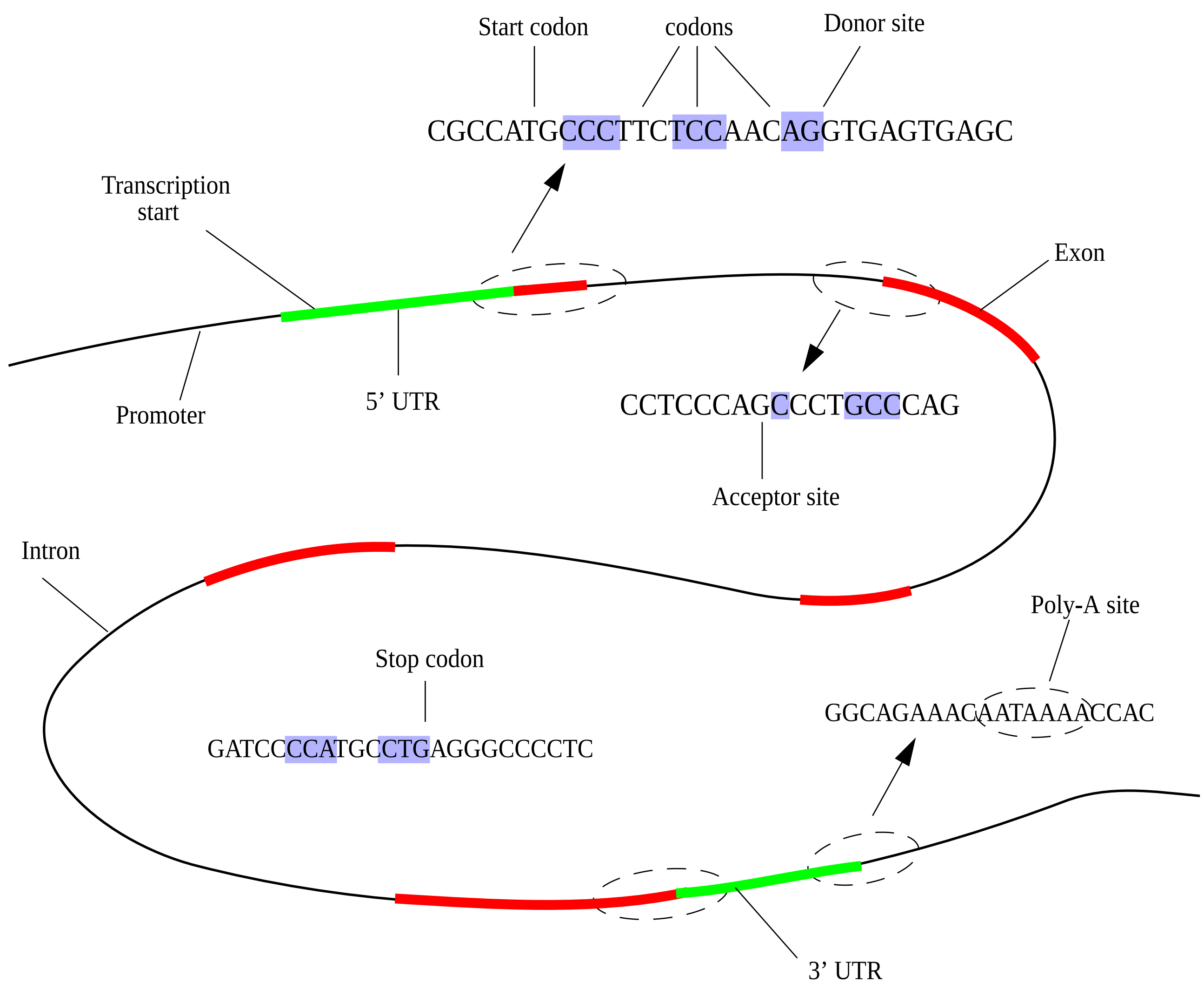


Figure 1: DNA with some signals shown.

Z Score vs p-values

In computational biology certain statistical tools are used to characterize underrepresented and overrepresented patterns. We illustrate it on $O_n(\mathcal{W})$.

Z-scores

$$Z(\mathcal{W}) = \frac{\mathbf{E}[O_n] - O_n(\mathcal{W})}{\sqrt{\mathbf{Var}[O_n(\mathcal{W})]}}$$

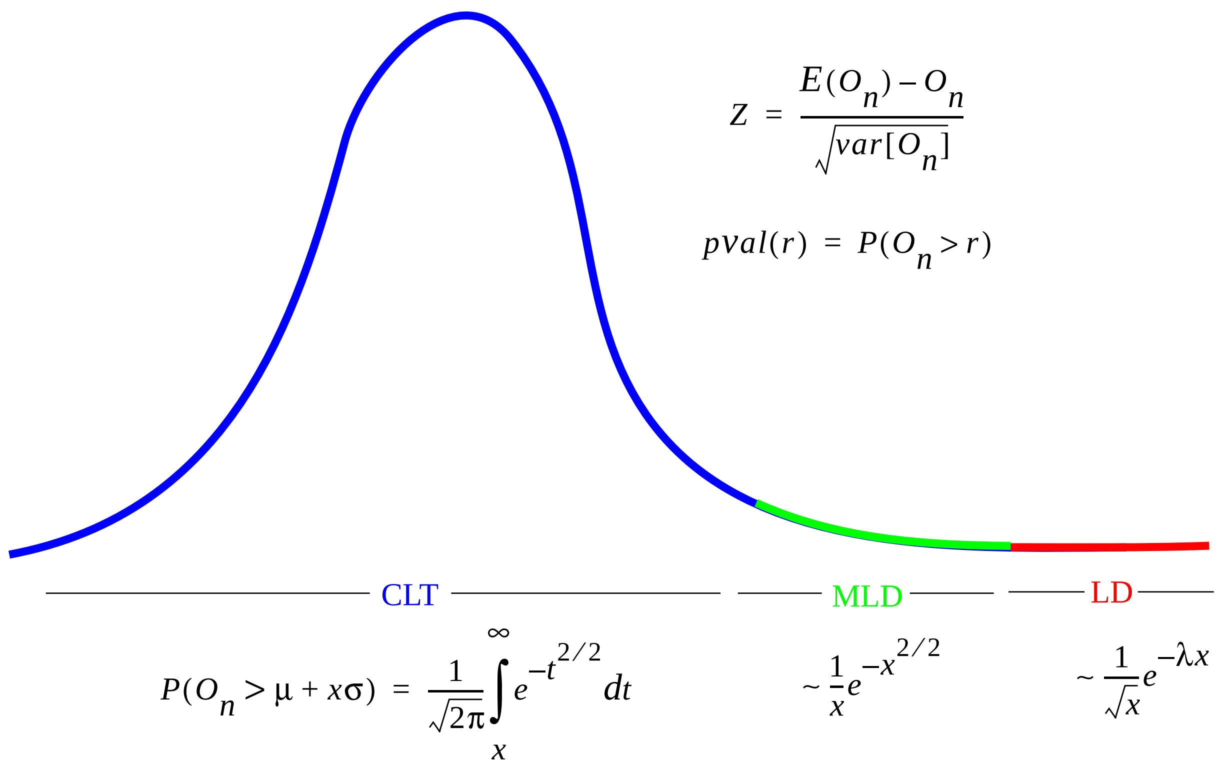
Z-score tells us how many standard deviations the observed value $O_n(\mathcal{W})$ is away from the mean.
This score makes sense only if one can prove that Z satisfies (at least asymptotically) the Central Limit Theorem (CLT), that is, Z is normally distributed.

p-values

$$pval(r) = P(O_n(\mathcal{W}) > \underbrace{\mathbf{E}[O_n] + x\sqrt{\mathbf{Var}[O_n]}}_{r}).$$

p values are used for very rare occurrences, far away from the mean (where CLT does not apply).
In order to compute p values one must apply either Moderate Large deviation (MLD) or Large Deviations (LD) results.

CLT vs LD



Let

$$P(O_n \geq n\alpha + x\sigma\sqrt{n})$$

Central Limit Theorem (CLT) – valid only in the square root off n vicinity of the mean, that is, for $x = O(1)$.

Moderate Large Deviations (MLD) – valid for $x \to \infty$ but $x = o(\sqrt{n})$.

Large Deviations (MLD) – valid for $x = O(\sqrt{n})$.

Z-scores and p values for A.thaliana

Table 1: Z score vs p-value of tandem repeats in *A.thaliana*.

Oligomer	Obs.	p-val (large dev.)	Z-sc.
AATTGGCGG	2	8.059×10^{-4}	48.71
TTTGTACCA	3	4.350×10^{-5}	22.96
ACGGTTCAC	3	2.265×10^{-6}	55.49
AAGACGGTT	3	2.186×10^{-6}	48.95
ACGACGCTT	4	1.604×10^{-9}	74.01
ACGCTTGG	4	5.374×10^{-10}	84.93
GAGAAGACG	5	0.687×10^{-14}	151.10

Remark: p values were computed using large deviations results of Regnier and S. (1998), and Denise and Regnier (2001) as we discuss below.

Some Theoretical Results (Single Pattern)

Here is an incomplete list of results on string pattern matching (given a pattern $\mathcal{W}$ find statistics of its occurrences):

- Feller (1968),
- Guibas and Odlyzko (1978, 1981),
- Prum, Rodolphe, and Turckheim (1995) – Markovian model, limiting distribution.
- Regnier & W.S. (1997,1998) – exact and approximate occurrences (memoryless and Markov models).
- P. Nicodéme, Salvy, & P. Flajolet (1999) – regular expressions.
- E. Bender and F. Kochman (1993) – general pattern matching.

Languages and Generating Functions

A language $\mathcal{L}$ is a collection of words satisfying some properties.

For any language $\mathcal{L}$ we define its generating function $L(z)$ as

$$L(z) = \sum_{u \in \mathcal{L}} P(u) z^{|u|}$$

where $P(w)$ is the stationary probability u occurrence, $|u|$ is the length of w.

For Markov sources we define $\mathcal{W}$-conditional generating function:

$$L_{\mathcal{W}}(z) = \sum_{u \in \mathcal{L}} P(u | u_{-m} = w_1 \cdots u_{-1} = w_m) z^{|u|}$$

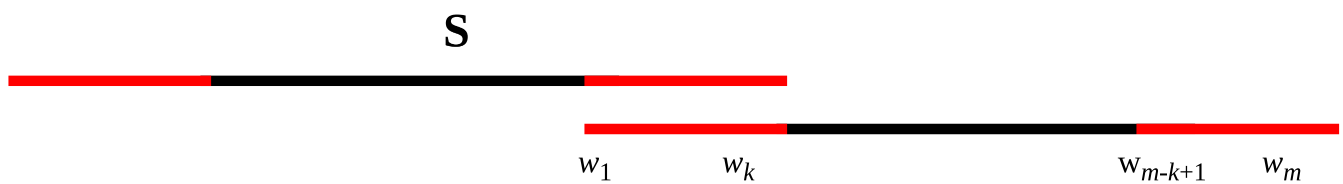
where u_{-i} stands for a symbol preceding the first character of u at distance i.

Autocorrelation Set and Polynomial

Given a pattern $\mathcal{W}$, we define the autocorrelation set $\mathcal{S}$ as:

$$\mathcal{S} = \{w_{k+1}^m : \ w_1^k = w_{m-k+1}^m\}, \quad w_1^k = w_{m-k+1}^m$$

and $\mathcal{WW}$ is the set of positions k satisfying $w_1^k = w_{m-k+1}^m$.



The generating function of $\mathcal{S}$ is denoted as $S(z)$ and we call it the autocorrelation polynomial.

$$S(z) = \sum_{k \in \mathcal{W}!\mathcal{W}} P(w_{k+1}^m) z^{m-k}.$$

Its $\mathcal{W}$-conditional generating function is denoted $S_{\mathcal{W}}(z)$. For example, for a Markov model we have

$$S_{\mathcal{W}}(z) = \sum_{k \in \mathcal{WW}} P(w_{k+1}^m | w_k^k) z^{m-k} \ .$$

Example

Example:

Let $\mathcal{W} = bab$ over alphabet $\mathcal{A} = \{a, b\}$.

$$\mathcal{WW} = \{1, 3\} \quad \text{and} \quad \mathcal{S} = \{\epsilon, ab\},$$

where ϵ is the empty word, since

b a b
    b a b

For the unbiased memoryless source

$$S(z) = 1 + P(ab)z^2 = 1 + \frac{z^2}{4}.$$

For the Markovian model of order one

$$S_{bab}(z) = 1 + P(ab|b)z^2 = 1 + p_{ba}p_{ab}z^2.$$

Language $\mathcal{T}_r$

We are interested in the following language:

$\mathcal{T}_r$ – set of words that contains exactly $r \geq 1$ occurrences of $\mathcal{W}$,

and its generating functions

$$\begin{aligned} T_r(z) &= \sum_{n \geq 0} \Pr\{O_n(\mathcal{W}) = r\} z^n, \quad r \geq 1, \\ T(z,u) &= \sum_{r=1}^{\infty} T_r(z) u^r = \sum_{r=1}^{\infty} \sum_{n=0}^{\infty} \Pr\{O_n(\mathcal{W}) = r\} z^n u^r \end{aligned}$$

for $|z| \leq 1$ and $|u| \leq 1$.

More Languages

(i) Let $\mathcal{T}$ be a language of words containing at least one occurrence of $\mathcal{W}$.

(ii) We define $\mathcal{R}$ as the set of words containing only one occurrence of $\mathcal{W}$, located at the right end. For example, for $\mathcal{W} = aba$

$$cc\,aba \in \mathcal{R}.$$

(iii) We also define $\mathcal{U}$ as

$$\mathcal{U} = \{u : \ \mathcal{W} \cdot u \cdot \in \mathcal{T}_1\}$$

that is, a word $u \in \mathcal{U}$ if $\mathcal{W} \cdot u$ has exactly one occurrence of $\mathcal{W}$ at the left end of $\mathcal{W} \cdot u$,

$$bba \in \mathcal{U}, \qquad ba \notin \mathcal{U}.$$

(iv) Let $\mathcal{M}$ be the language:

$$\mathcal{M} = \{u : \ \mathcal{W} \cdot u \in \mathcal{T}_2 \text{ and } \mathcal{W} \text{ occurs at the right of } \mathcal{W} \cdot u\},$$

that is, $\mathcal{M}$ is a language such that $\mathcal{W}\mathcal{M}$ has exactly two occurrences of $\mathcal{W}$ at the left and right end of a word from $\mathcal{M}$.

$$ba \in \mathcal{M} \qquad ababa$$

Basic Lemma

Lemma 1. *The language* $\mathcal{T}$ *satisfies the fundamental equation:*

$$\mathcal{T} = \mathcal{R} \cdot \mathcal{M}^* \cdot \mathcal{U} \quad .$$

Notably, the language $\mathcal{T}_r$ *can be represented for any* $r \geq 1$ *as follows:*

$$\mathcal{T}_r = \mathcal{R} \cdot \mathcal{M}^{r-1} \cdot \mathcal{U},$$

and

$$\mathcal{T}_0 \cdot \mathcal{W} = \mathcal{R} \cdot \mathcal{S} \quad .$$

Here, by definition $\mathcal{M}^0 := \{\epsilon\}$ *and* $\mathcal{M}^* := \bigcup_{r=0}^{\infty} M^r$*.*

Example: Let $\mathcal{W} = TAT$. The following string belongs $\mathcal{T}_3$:

$$\overbrace{CCTAT}^{\mathcal{R}}\underbrace{AT}_{\mathcal{M}}\underbrace{GATAT}_{\mathcal{M}}\overbrace{GGA}^{\mathcal{U}}.$$

More Results

Theorem 1. (i) *The languages $\mathcal{M}, \mathcal{U}$ and $\mathcal{R}$ satisfy:*

$$\begin{aligned}
\bigcup_{k\geq 1} \mathcal{M}^k &= \mathcal{A}^* \cdot \mathcal{W} + \mathcal{S} - \{\epsilon\} \ , \\
\mathcal{U} \cdot \mathcal{A} &= \mathcal{M} + \mathcal{U} - \{\epsilon\}, \\
\mathcal{W} \cdot \mathcal{M} &= \mathcal{A} \cdot \mathcal{R} - (\mathcal{R} - \mathcal{W}) \ ,
\end{aligned}$$

where $\mathcal{A}^*$ is the set of all words, $+$ and $-$ are disjoint union and subtraction of languages.

(ii) *The generating functions associated with languages $\mathcal{M}, \mathcal{U}$ and $\mathcal{R}$ satisfy for memoryless sources*

$$\begin{aligned}
\frac{1}{1 - M(z)} &= S_{\mathcal{W}}(z) + P(\mathcal{W}) \frac{z^m}{1 - z}, \\
U_{\mathcal{W}}(z) &= \frac{M(z) - 1}{z - 1}, \\
R(z) &= P(\mathcal{W}) z^m \cdot U_{\mathcal{W}}(z)
\end{aligned}$$

(Extension to Markov sources possible; cf. Regnier & WS.)

Main Results: Exact

Theorem 2. *The generating functions* $T_r(z)$ *and* $T(z,u)$ *are*

$$\begin{aligned} T_r(z) &= R(z)M_{\mathcal{W}}^{r-1}(z)U_{\mathcal{W}}(z)\ , \quad r \geq 1 \\ T(z,u) &= R(z)\frac{u}{1-uM(z)}U_{\mathcal{W}}(z) \\ T_0(z)P(\mathcal{W}) &= R(z)S_{\mathcal{W}}(z) \end{aligned}$$

where

$$\begin{aligned} M(z) &= 1 + \frac{z-1}{D_{\mathcal{W}}(z)}\ , \\ U_{\mathcal{W}}(z) &= \frac{1}{D_{\mathcal{W}}(z)}\ , \\ R(z) &= z^m P(\mathcal{W})\frac{1}{D_{\mathcal{W}}(z)}\ . \end{aligned}$$

with

$$D_{\mathcal{W}}(z) = (1-z)S_{\mathcal{W}}(z) + z^m P(\mathcal{W}).$$

Main Results: Asymptotics

Theorem 3. (i) *Moments. The expectation satisfies, for* $n \geq m$:

$$\mathbf{E}[O_n(\mathcal{W})] = P(\mathcal{W})(n - m + 1) \; ,$$

while the variance is

$$\mathbf{Var}[O_n(\mathcal{W})] = nc_1 + c_2.$$

with

$$\begin{aligned} c_1 &= P(\mathcal{W})(2S(1) - 1 - (2m-1)P(\mathcal{W}) \; , \\ c_2 &= P(\mathcal{W})((m-1)(3m-1)P(\mathcal{W}) \\ &- (m-1)(2S(1) - 1) - 2S'(1)). \end{aligned}$$

Distributions

(ii) *Case* $r = O(1)$. *Let* $\rho_{\mathcal{W}}$ *be the smallest root of*

$$D_{\mathcal{W}}(z) = (1 - z)S_{\mathcal{W}}(z) + z^m P(\mathcal{W}) = 0.$$

Then

$$\Pr\{O_n(\mathcal{W}) = r\} \quad \sim \quad \sum_{j=1}^{r+1} (-1)^j a_j \binom{n}{j-1} \rho_{\mathcal{W}}^{-(n+j)}$$

where

$$a_{r+1} = \frac{\rho_{\mathcal{W}}^m P(\mathcal{W}) \left(\rho_{\mathcal{W}} - 1\right)^{r-1}}{\left(D'_{\mathcal{W}}(\rho_{\mathcal{W}})\right)^{r+1}} \,,$$

and the remaining coefficients can be easily computed, too.

Central Limit and Large Deviations

(iii) *CLT*: *Case* $r = EO_n + x\sqrt{\mathbf{Var}O_n}$ *for* $x = O(1)$. *Then:*

$$\Pr\{O_n(\mathcal{W}) = r\} = \frac{1}{\sqrt{2\pi c_1 n}} e^{-\frac{1}{2}x^2} \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right) .$$

(iv) *Large Deviations*: *Case* $r = (1+\delta)EO_n$. *Let* $a = (1+\delta)P(\mathcal{W})$ *with* $\delta \neq 0$. *For complex* t, *define* $\rho(t)$ *to be the root of*

$$1 - e^t M_{\mathcal{W}}(e^\rho) = 0 ,$$

while ω_a *and* σ_a *are defined as*

$$\begin{aligned} -\rho'(\omega_a) &= a \\ -\rho''(\omega_a) &= \sigma_a^2 \end{aligned}$$

Then

$$\Pr\{O_n(\mathcal{W}) \quad \sim \quad (1+\delta)EO_n\} = \frac{e^{-(n-m+1)I(a)+\delta_a}}{\sigma_a\sqrt{2\pi(n-m+1)}}$$

where $I(a) = a\omega_a + \rho(\omega_a)$ *and* δ_a *is a constant.*

Biology – Weak Signals and Artifacts

Denise and Regnier (2002) observed that in biological sequence whenever a word is overrepresented, then its subwords are also overrepresented.
For example, if $\mathcal{W}_1 = AATAAA$, then

$$\mathcal{W}_2 = ATAAAN$$

is also overrepresented.

Overrepresented subword is called artifact.

It is important to disregard automatically noise created by artifacts.

Example:
1. Popular Alu sequence introduces artifacts noise.

2. Another example is χ-sequence $GNTGGTGG$ in H.influenzae (Nicodeme, 2000).

Discovering Artifacts

New Approach:

> Once a dominating signal has been detected, we look for a weaker signal by comparing the number of observed occurrences of patterns to the conditional expectations **not** the regular expectations.

In particular, using the methodology presented above Denise and Regnier (2002) were able to prove that

$$\mathbf{E}[O_n(\mathcal{W}_2)|O_n(\mathcal{W}_1) = k] \sim \alpha n$$

provided $\mathcal{W}_1$ is overrepresented, where α can be explicitly computed (often $\alpha = P(\mathcal{W}_2)$ is $\mathcal{W}_1$ and $\mathcal{W}_2$ do not overlap).

Polyadenylation Signals in Human Genes

Beaudoing et al. (2000) studied several variants of the well known AAUAAA polyadenylation signal in mRNA of humans genes. To avoid artifacts Beaudoing et al cancelled all sequences where the overrepresented hexamer was found.

Using our approach Denise and Regnier (2002) discovered/eliminated all artifacts and found new signals in a much simpler and reliable way.

Hexamer	Obs.	Rk	Exp.	Z-sc.	Rk	Cd.Exp.	Cd.Z-sc.	Rk
AAUAAA	3456	1	363.16	167.03	1			1
AAAUAA	1721	2	363.16	71.25	2	1678.53	1.04	1300
AUAAAA	1530	3	363.16	61.23	3	1311.03	6.05	404
UUUUUU	1105	4	416.36	33.75	8	373.30	37.87	2
AUAAAU	1043	5	373.23	34.67	6	1529.15	12.43	4078
AAAAUA	1019	6	363.16	34.41	7	848.76	5.84	420
UAAAAU	1017	7	373.23	33.32	9	780.18	8.48	211
AUUAAA	1013	I	373.23	33.12	10	385.85	31.93	3
AUAAAG	972	9	184.27	58.03	4	593.90	15.51	34
UAAUAA	922	10	373.23	28.41	13	1233.24	–8.86	4034
UAAAAA	922	11	363.16	29.32	12	922.67	9.79	155
UUAAAA	863	12	373.23	25.35	15	374.81	25.21	4
CAAUAA	847	13	185.59	48.55	5	613.24	9.44	167
AAAAAA	841	14	353.37	25.94	14	496.38	15.47	36
UAAAUA	805	15	373.23	22.35	21	1143.73	–10.02	4068

Outline of the Talk

1. Pattern Matching Problems
2. Biology
3. Information Security – Subsequence Matching
   - Some Theory (De Bruijn Automaton)
   - Reliable Threshold in Intrusion Detection
4. Multimedia Compression

Application – Information Security

Convert all color commands to black or
Since PostScript files are often extremely large, it makes sense to try to compress them with either the zip or gzip programs. In such a case, the eps file is replace by a file with extension zip or eps gz, or eps-gz. Two problems now arise: first LATEX cannot read such files to obtain the bounding box information, and secondly, the driver needs to unpack such a file to include it in the final output. This can be accomplished with, for example: Declare-GraphicsRule.eps.gzeps.eps.bbgunzip which stablizes the graphics type as eps with the bounding box information in the file of the same name and extension. Convrt all color commands to black or white.

Imagine that the file above is **audit file**. An attacker/attacker left a signature/signature as a **subsequence** in the file.

How to know whether this subsequence constitutes an attack or is merely a result of randomness?

How to minimize the number of false positives?

Subsequence Matching (Hidden Words)

A subsequence pattern occurrence or a hidden word occurrence is defined by a pair:

$$(\mathcal{W}, \mathcal{D})$$

– the pattern $\mathcal{W} = w_1 \cdots w_m$ is a word of length m;
– the **constraint** $\mathcal{D} = (d_1, \ldots, d_{m-1})$ such that m-tuple $I = (i_1, i_2, \ldots, i_m)$ satisfies

$$i_{j+1} - i_j \le d_j,$$

The I-tuple is called a position.

Let $\mathcal{P}_n(\mathcal{D})$ be the set of all positions subject to the separation constraint $\mathcal{D}$.
An occurrence of pattern $\mathcal{W}$ in the text T_n subject to $\mathcal{D}$ is a position $I = (i_1, i_2, \ldots, i_m)$ such that

$$T_{i_1} = w_1, \; T_{i_2} = w_2, \; \ldots, \; T_{i_m} = w_m.$$

Basic Equation

Unconstrained problem: $\mathcal{D} = (\infty, \ldots, \infty)$.

constrained problem: all d_j are finite.

Let $O_n(\mathcal{W})$ be the number of $\mathcal{W}$ occurrences in T. Observe that

$$O_n(\mathcal{W}) = \sum_{I \in \mathcal{P}_n(\mathcal{D})} X_I$$

where

$$X_I := [\![\mathcal{W} \text{ occurs at position } I \text{ in } T_n]\!]$$

with

$$[\![B]\!] = \begin{cases} 1 & \text{if the property } B \text{ holds,} \\ 0 & \text{otherwise.} \end{cases}$$

Below analysis is based on:
P. Flajolet, W.S., and B. Vallee, ICALP 2001 & JACM 2005.

Very Little Theory – Constrained Problem

Let us analyze the constrained subsequence problem. We reduce it to the generalized string matching problem using the de Bruijn automaton.

1. The $(\mathcal{W}, \mathcal{D})$ **constrained subsequence problem** will be viewed as the generalized string matching problem by assuming that $\mathcal{W}$ is the set of all possible patterns.

Example: If $(\mathcal{W}, \mathcal{D}) = a\#_2 b$, then

$$\mathcal{W} = \{ab, aab, abb\}.$$

2. de Bruijn Automaton.

Let $M = \max\{length(\mathcal{W})\} - 1$ (e.g., $M = 2$ in the above example). Define

$$\mathcal{B} = \mathcal{A}^M.$$

De Bruijn automaton is built over $\mathcal{B}$.

3. Let $b \in \mathcal{B}$ and $a \in \mathcal{A}$. Then the transition from the state b upon scanning symbol a of the text is to $\hat{b} \in \mathcal{B}$ such that

$$ba \mapsto \hat{b} = b_2 b_3 \cdots b_M a,$$

that is, the leftmost symbol of b is erased and symbol a is appended on the right. For example

$$\underbrace{abb}_{\mathcal{B}}\underbrace{a}_{\mathcal{A}} \mapsto \underbrace{bba}_{\mathcal{B}}.$$

4. The Transition Matrix

Let $\mathbf{T}(u)$ be complex-valued transition matrix define as:

$$[\mathbf{T}(u)]_{b,\hat{b}} := P(a) u^{O_{M+1}(ba) - O_M(b)} [\![\, \hat{b} = b_2 b_3 \cdots b_M a \,]\!]$$

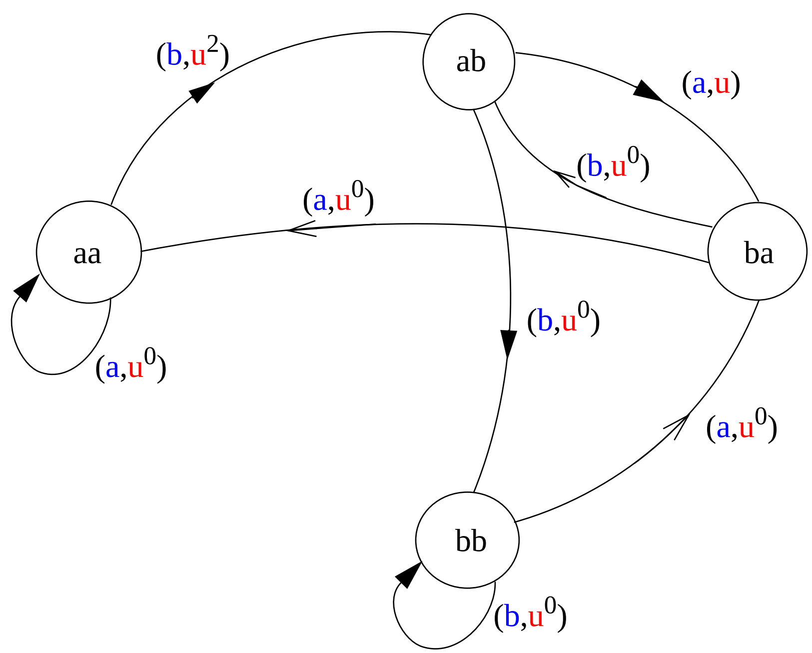
where $O_M(b)$ is the number of pattern occurrences in the text b.

Example

5. Example

Let $\mathcal{W} = \{ab, aab, aba\}$. Then $M = 2$, the de Bruijn graph is as below and the matrix $\mathbf{T}(u)$ is shown below

$$\mathbf{T}(u) = \begin{array}{c c} & \begin{array}{cccc} aa & ab & ba & bb \end{array} \\ \begin{array}{c} aa \\ ab \\ ba \\ bb \end{array} & \begin{pmatrix} P(a) & P(b)\,u^2 & 0 & 0 \\ 0 & 0 & P(a)\,u & P(b) \\ P(a) & P(b) & 0 & 0 \\ 0 & 0 & P(a) & P(b) \end{pmatrix} \end{array}.$$



Generating Functions

6. Using properties of product of matrices we conclude that

$$O_n(u) = \mathbf{E}[u^{O_n(\mathcal{W})}] = \mathbf{b}^t(u)\mathbf{T}^n(u)\vec{1}$$

where $\mathbf{b}^t(u)$ is an initial vector and $\vec{1} = (1, \ldots, 1)$.

7. Spectral Decomposition
Let $\lambda(u)$ be the largest eigenvalue of $\mathbf{T}(u)$ (which we know that it exists). Then

$$O_n(u) = c(u)\lambda^n(u)(1 + O(A^n))$$

for some $A < 1$. This proves that the generating function $O_n(u)$ satisfies the so called quasi-power law.

Final Results

8. Mean and Variance

$$\begin{aligned}\mathbf{E}[O_n(\mathcal{W})] &= n\Lambda'(0)+O(1) = nP(\mathcal{W})+O(1),\\ \mathbf{Var}[O_n(\mathcal{W})] &= n\Lambda''(0)+O(1) = n\sigma^2(\mathcal{W})+O(1)\end{aligned}$$

where $\Lambda(s) = \log\lambda(e^s)$

9. Central Limit Theorem

$$\Pr\left\{\frac{O_n - nP(\mathcal{W})}{\sigma(\mathcal{W})\sqrt{n}} \le x\right\} \sim \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{x} e^{-t^2/2}$$

10. Large deviations

If $\mathbf{T}(u)$ is primitive, then

$$\Pr\{O_n(\mathcal{W}) = a\mathbf{E}[O_n]\} \sim \frac{1}{\sigma_a\sqrt{2\pi n}}e^{-nI(a)+\theta_a}$$

where $I(a)$ can be explicitly computed, and θ_a is a known constant.

Reliable Threshold for Intrusion Detection

We argued that one needs a reliable threshold for intrusion detection. If false alarms are to be avoided, the problem is of finding a threshold $\alpha_0 = \alpha_0(\mathcal{W}; n, \beta)$ such that

$$P(O_n(\mathcal{W}) > \alpha_{th}) \leq \beta(= 10^{-5}).$$

Our results shows that

$$\alpha_{th} = nP(\mathcal{W}) + x_0\sigma(\mathcal{W})\sqrt{n}, \quad \beta = \frac{1}{\sqrt{2\pi}} \int_{x_0}^{\infty} e^{-t^2/2} dt \sim \frac{1}{x_0} e^{-x_0^2/2}.$$

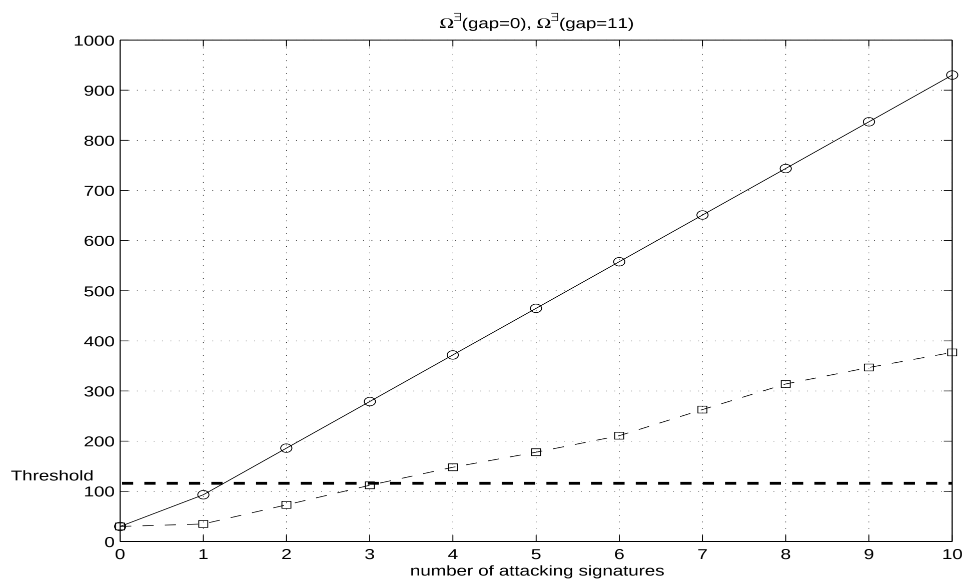


Figure 2: Pattern=wojciech, window=100 (cf. Gwadrea at al. (2004).

Outline of the Talk

1. Pattern Matching Problems
2. Biology
3. Information Security
4. Multimedia Compression — Self-Repetitive Matching
   - Theoretical Foundation (Renyi's Entropy)
   - Data Structures and Algorithms
   - Video Compression (**Demo**)
   - Error Resilient LZ'77 (Suffix Trees)

Application – Multimedia Compression

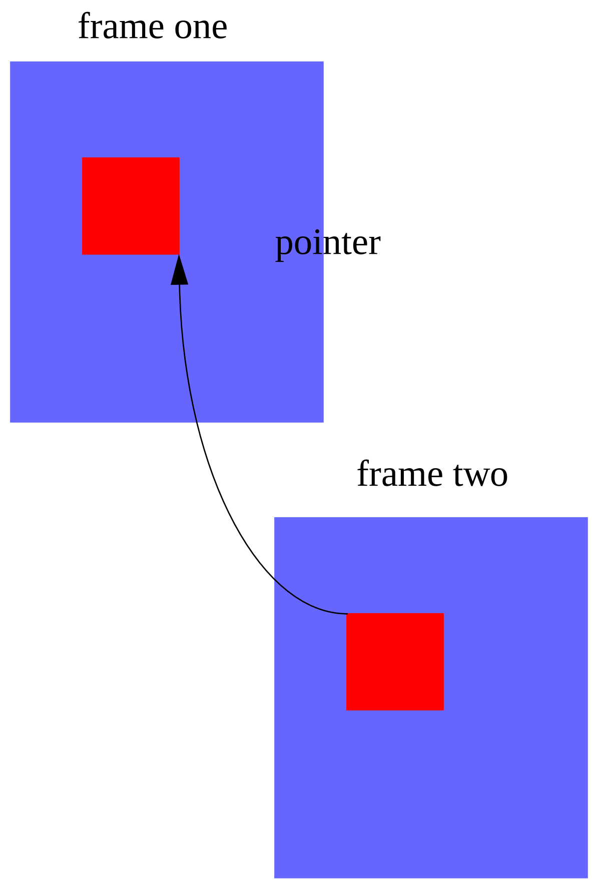
Is length of code $(X_1^M) < M$?

By how much?

Optimal compression

$$r_n(x_1^M) = \frac{1}{M}\sum_i \log_2 n + \theta(L_n^i)$$

$$L_n^i = |z_i|$$

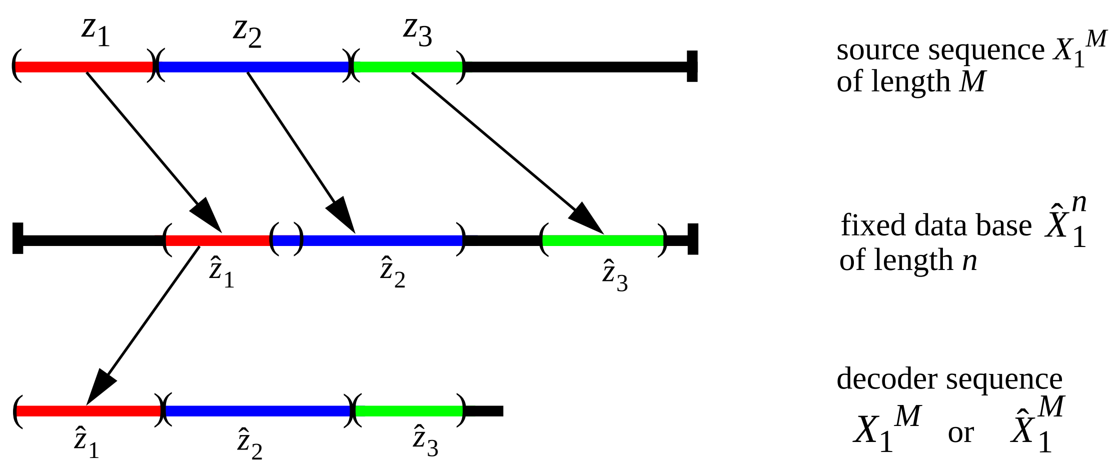


$$Code = (pointer, length, width)$$

Is the **code length** shorter than the **original file**?

1. T. Luczak and W.S., IEEE Inf. Theory, 1997.
2. M. Alzina, W.S., A. Grama, IEEE Image Proc., 2002.

Lossy Lempel-Ziv Scheme



Source sequence (e.g., second frame in a video stream) X_1^M is assumed to be of length M.

Fixed database (e.g., the first frame in video) X_1^n is of length n.

Code C_n of length $l(C_n)$ is a function from $\mathcal{A}^n$ to $\{0,1\}^*$ that represents the source sequence.

$$Code = (pointer, length).$$

Reproduction sequence $\hat{X}_1^n$ that approximates the source sequence (e.g., for a given D and a distortion measure $d(\cdot,\cdot)$ such that $d(X_1^n, \hat{X}_1^n) < D$).

Bit rate

$$r_n(X_1^M) = \frac{length(C_n(X_1^M))}{n}.$$

Some Definitions

Lossy Lempel-Ziv algorithm partitions according to Π_n the source sequence X_1^M into variable phrases $Z^1, \ldots, Z^{|\Pi_n|}$ of length $L_n^1, \ldots, L_n^{|\Pi_n|}$.

Code length: Since Code=(ptr, length) the length of the code for the source sequence X_1^M is

$$l_n(X_1^M) = \sum_{i=1}^{|\Pi_n|} \log n + \Theta(\log L_n^i)$$

and hence the bit rate is

$$r_n(X_1^M) = \frac{1}{M} \sum_{i=1}^{|\Pi_n|} \log n + \Theta(\log L_n^i).$$

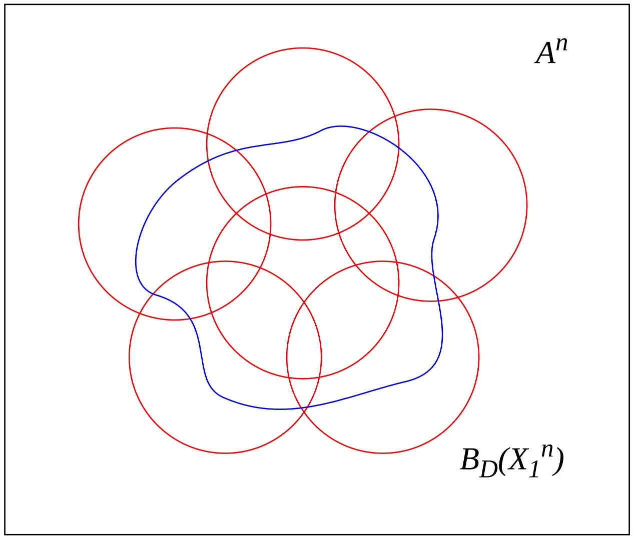
How much do we gain?

How much do we compress?

Generalized Shannon Entropy is defined as

$$\hat{r}_0(D) = \lim_{n\to\infty} \frac{\mathbf{E}_P[-\log P(B_D(X_1^n))]}{n},$$

where $B_D(x_1^n) = \{y_1^n : d(y_1^n, x_1^n) \le D\}$ is a ball of radius D with center x_1^n.



$P(B_D(X_1^n)) \sim 2^{-r_0(D)\cdot n}$

Theorem [T. Luczak and W.S, 1997] For Markov sources

$$\lim_{n\to\infty} \frac{L_n^1}{\log n} = \frac{1}{\hat{r}_0(D)}, \qquad \text{(pr.)}.$$

and

$$\lim_{n\to\infty} \lim_{M\to\infty} \mathbf{E}[r_n(X_1^M)] = \hat{r}_0(D).$$

We implemented 2D Pattern Matching Compression (2D-PMC) scheme that has three major encoding mechanisms:

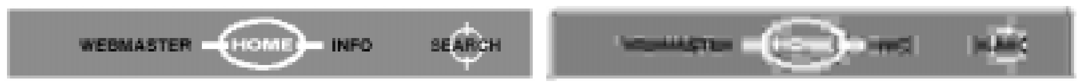
- 2D Pattern Matching
- Enhanced Run-Length Encoding
- Lossless Coding

2D pattern matching is the most efficient encoding. The basic idea is to find a two-dimensional region (rectangle) in the uncompressed part of the image that occurs approximately in the compressed part (i.e., database), and to store a pointer to it along with the width and the length of the repeated rectangle, as shown on the next slide.

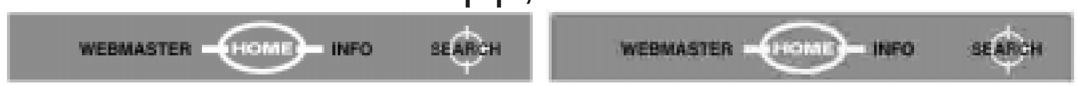
Run-length encoding (RLE) of images identifies regions of the image with constant pixel values. We enhance RLE by giving it the capability of coding regions in which pixel values can be (approximately) modeled by a planar function.

Sample of Image Compression Results

Pattern Matching Compression



Banner 0.29 bpp, 2DPMIC vs. JPEG



Banner 0.50 bpp, 2DPMIC vs. JPEG

(JPEG)

(2D-PMC)

Comparisons for Images

2D-PMIC				JPEG			
BPP	CR	RMSE	PSNR	BPP	CR	RMSE	PSNR
Image: Banner							
0.29	28.00	9.5	28.6	0.29	28.00	27.4	19.4
0.50	16.00	1.3	45.8	0.50	16.00	15.3	24.5
0.54	14.89	0.0	Inf	1.01	7.94	15.1	24.6
				2.00	4.00	15.1	24.6
Image: Basselope							
0.27	29.56	21.0	21.7	0.25	32.31	19.3	22.4
0.51	15.58	12.6	26.2	0.50	16.17	12.5	26.2
0.96	8.33	0.0	Inf	1.00	7.95	6.9	31.4
				2.01	4.08	2.6	39.7
Image: Lena							
0.25	32.01	10.8	27.5	0.25	32.30	8.9	29.1
0.49	29.30	8.7	29.3	0.50	16.03	5.8	32.9
1.05	7.61	5.6	33.1	1.00	8.04	4.2	35.7
1.94	4.13	3.6	37.1	2.01	3.81	2.7	39.4
Image: San Francisco							
0.25	32.00	17.0	23.5	0.25	32.00	15.5	24.3
0.50	16.00	13.1	25.8	0.50	16.00	10.6	27.6
1.05	7.59	6.7	31.6	1.00	8.02	6.8	31.5
2.03	3.95	2.9	38.8	2.01	3.98	3.5	37.2

Video Compression – Statistics

Video Pattern Matching Compression

Sample	MPG	PMC	Comp. Time		Decomp. Time	
			MPG	PMC	MPG	PMC
Claire	17.7	19.1	2	26	0.36	0.05
Football	111.9	90.9	3	29	0.34	0.09
Missa	20.4	20.2	9	23	0.32	0.03
PomPom	187.1	174.6	7	34	0.35	0.07
PingPong	113.8	104.9	8	39	0.35	0.03
Train	202.8	139.3	9	25	0.35	0.04

Table 2: Comparison of data rates (KB/s), compression, and decompression times. 2DPMC yields performance ranging from 7.9% worse to 31.3% better than MPEG2.

Outline of the Talk

1. Pattern Matching Problems
2. Biology
3. Information Security
4. Multimedia Compression — Self-Repetitive Matching
   - Theoretical Foundation
   - Data Structures and Algorithms
   - Video Compression
   - Error Resilient LZ'77 (Suffix Trees)

Error Resilient LZ'77 Scheme

In the LZ'77 there are many copies of the longest match that can be used to correct errors.
We denote by M_n the number of such copies.

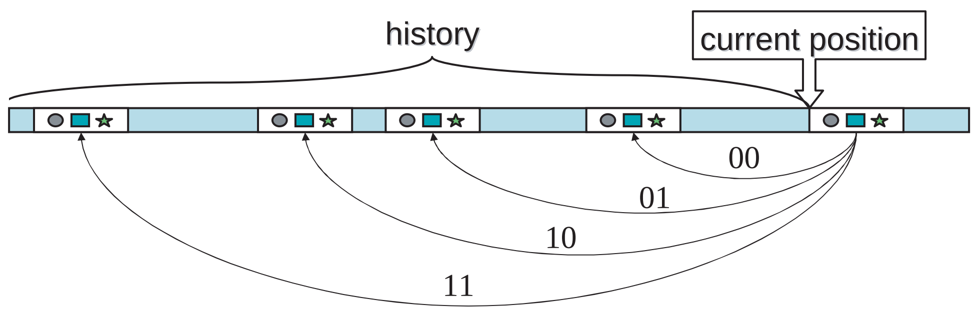


Figure 3: At this stage in LZ'77, we have $M_n = 4$.

Source Coding vs. Channel coding

Source Coding (i.e., Data Compression)

- Goal: Represent the source information with a minimum of symbols

Channel Coding (i.e., Error Correction)

- Goal: Represent the source information with a minimum of error probability in decoding

The goals of source and channel coding are conflicting:

Channel coding traditionally requires additional symbols to perform error correction.

Solution: Joint Source-Channel Coding.

Main Idea of the LZRS'77

Lonardi and W.S. in 2003 proposed a joint source-channel coding for LZ'77 by recovering parity bits needed for the Reed-Solomon channel coding from redundancy (multiple copies of longest match) of LZ'77.

Definition: Consider the stage at which n bits of a phrase have already been compressed by LZ'77. By M_n we denote the number of copies of the longest prefix of the uncompressed string that appear in the database.

By a judicious choice of pointers in the LZ'77 scheme, we can recover $\lfloor \log_2 M_n \rfloor$ bits at this stage.

In fact, if this greediness is relaxed (say, by looking for the 10th largest prefix, for instance), then the number of copies found in the database will increase significantly. This would allow even more errors to be corrected.

Encoder and Decoder of LZRS'77

We use the family of Reed-Solomon codes RS$(255, 255-2e)$ that contains blocks of 255 bytes, of which $255-2e$ are idata and $2e$ are parity.

Encoder: The data is broken into blocks of size $255-2e$. Then, blocks are processed in reverse order, beginning with the very last. When processing block i, the encoder computes first the Reed-Solomon parity bits for the block $i+1$ and then it embeds the extra bits in the pointers of block i.

Decoder: The decoder receives a sequence of pointers, preceded by the parity bits of the first block. It uses parity bits to correct block B_1. Once block B_1 is correct, it decompresses it using LZS'77. Redundant bits of block B_1 are used as parity bits to correct block B_2, etc.

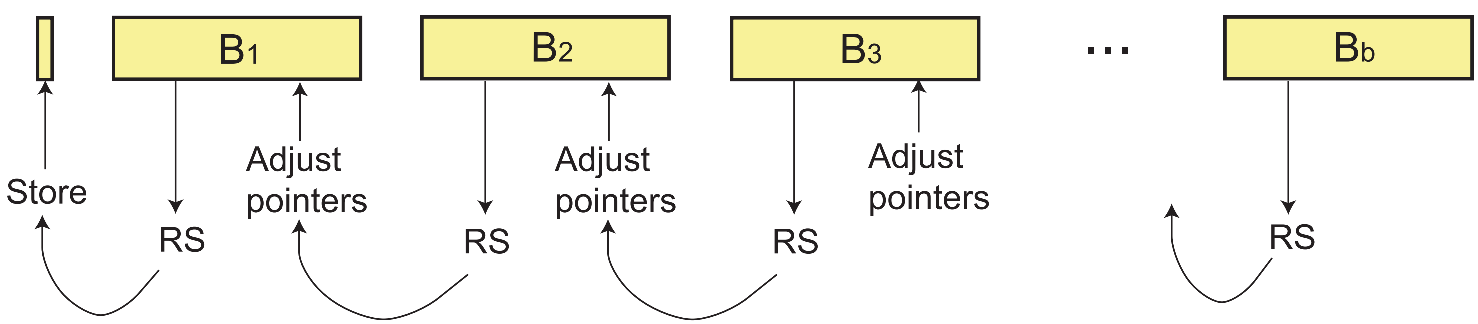


Figure 4: The right-to-left sequence of operations on the blocks for the encoder

Analysis of M_n Via Suffix Trees

Build a suffix tree from the first n suffixes of X (i.e., $X_1^\infty, X_2^\infty, \ldots, X_n^\infty$). Then insert the $(n+1)$st suffix, namely X_{n+1}^∞.
Observe: M_n is the size of the subtree that starts at the insertion point of the $(n+1)$st suffix.

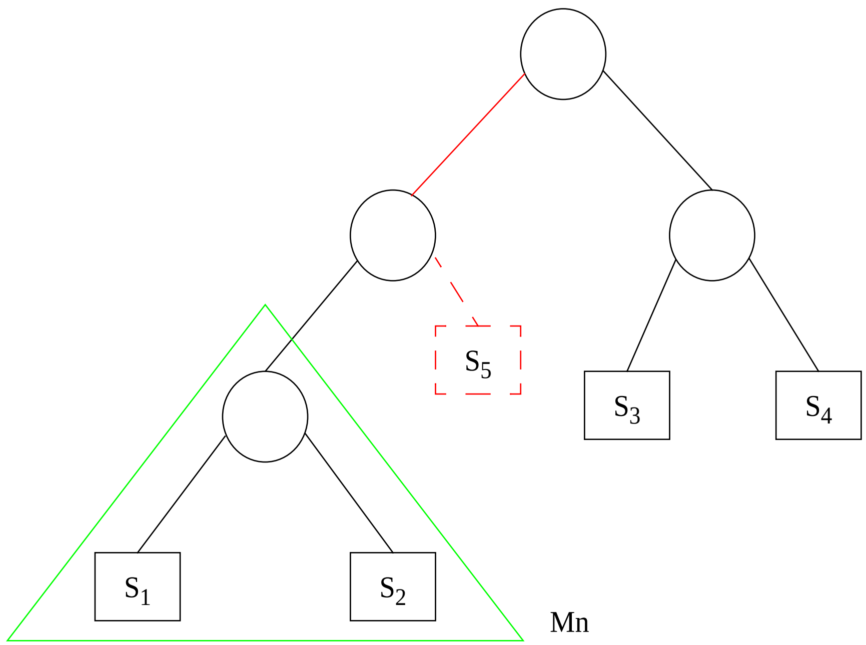


Figure 5: M_4 is the size of the subtree at the insertion point of S_5. Here $M_4^I = 2$.

Main Results

Theorem 4 (Ward, W.S., 2005). *Let* $z_k = \frac{2kr\pi i}{\ln p}\ \forall k \in \mathbb{Z}$, *where* $\frac{\ln p}{\ln q} = \frac{r}{s}$ *for some relatively prime* $r, s \in \mathbb{Z}$ *(we are most interested in the situation where* $\frac{\ln p}{\ln q}$ *is rational). Then*

$$\begin{aligned} E[(M_n)^j] &= \Gamma(j)\frac{q(p/q)^j + p(q/p)^j}{h} \\ &\quad + \delta_j(\log_{1/p} n) - \frac{1}{2}n\left(\frac{d^2}{dz^2}\delta_j(\log_{1/p} z)\right)\Bigg|_{z=n} + O(n^{-2}) \end{aligned}$$

where Γ *is the Euler gamma function and*

$$\delta_j(t) = \sum_{k\neq 0} -\frac{e^{2kr\pi i t}\Gamma(z_k + j)\left(p^j q^{-z_k-j+1} + q^j p^{-z_k-j+1}\right)}{p^{-z_k+1}\ln p + q^{-z_k+1}\ln q}.$$

We emphasize $-\frac{1}{2}n\left(\frac{d^2}{dz^2}\delta_j(\log_{1/p} z)\right)\Big|_{z=n}$ is $O(n^{-1})$. Also δ_j is a periodic function that has small magnitude and exhibits fluctuation when $\frac{\ln p}{\ln q}$ is rational.

Distribution of M_n

Theorem 5 (Ward, W.S., 2005). *Let $z_k = \frac{2kr\pi i}{\ln p}$ $\forall k \in \mathbb{Z}$, where $\frac{\ln p}{\ln q} = \frac{r}{s}$ for some relatively prime $r, s \in \mathbb{Z}$. Then*

$$E[u^{M_n}] = -\frac{q\ln(1-pu) + p\ln(1-qu)}{h} + \delta(\log_{1/p} n, u) + O(n^{-2})$$

where

$$\delta(t,u) = \sum_{k\neq 0} -\frac{e^{2kr\pi it}\Gamma(z_k)\left(q(1-pu)^{-z_k} + p(1-qu)^{-z_k} - p^{-z_k+1} - q^{-z_k+1}\right)}{p^{-z_k+1}\ln p + q^{-z_k+1}\ln q}.$$

and Γ is the Euler gamma function.

Corollary 1. *It follows immediately that*

$$E[u^{M_n}] = \sum_{j=1}^{\infty}\left[\frac{p^j q + q^j p}{jh} + \sum_{k\neq 0} -\frac{e^{2kr\pi i \log_{1/p} n}\Gamma(z_k)(p^j q + q^j p)(z_k)^{\overline{j}}}{j!(p^{-z_k+1}\ln p + q^{-z_k+1}\ln q)}\right]u^j + O(n^{-1})$$

and

$$P(M_n = j) = \frac{p^j q + q^j p}{jh} + \sum_{k\neq 0} -\frac{e^{2kr\pi i \log_{1/p} n}\Gamma(z_k)(p^j q + q^j p)(z_k)^{\overline{j}}}{j!(p^{-z_k+1}\ln p + q^{-z_k+1}\ln q)} + O(n^{-1}).$$