STRING ATTRACTORS: A UNIFYING THEORY OF REPETITIVENESS

Dominik Kempa
Department of Computer Science, University of Helsinki
Helsinki, Finland

Nicola Prezza
Department of Computer Science, University of Pisa
Pisa, Italy
DTU Compute, Technical University of Denmark
Kongens Lyngby, Denmark

A well-known fact in the field of text compression is that entropy is a weak model when the input contains repetitions. To address this, decades of research generated myriads of the so-called dictionary compressors. We show that these techniques are different solutions to the same, elegant, combinatorial problem: to find a small set of positions capturing all text's substrings. We call such set a string attractor.

We show reductions between dictionary compressors and attractors which allows us to uncover new relations between the output sizes of different dictionary compressors. We then provide matching lower and upper bounds for the random access problem on attractors and show that they transfer to many compressors. Finally, we study the computational complexity of the problem of finding the smallest attractor.

DICTIONARY COMPRESSION

Encoding of string that replaces repetitions with pointers to other occurrences. Useful since entropy compression is not sensitive to repetitions:

$$nH_k(T^t) \approx t \cdot nH_k(T)$$

Highly repetitive data:

- software repositories (github)
- genomic databases (1000-Genomes project)
- versioned documents (Wikipedia)

EXAMPLES

Lempel-Ziv factorization $LZ77(T)$ is a greedy partition of T into *longest previous factors* (LPFs). LPF at position i is the longest factor $T[i..i+\ell)$ occurring also at some position $j < i$.

Example:

i	0	1	2	3	4	5	6	7	8	9
T[i]	A	B	A	B	B	A	B	B	A	B

LPF[2] = AB (j = 0)
LPF[5] = ABBAB (j = 2)

LZ(T): | A | B | A B | B A B B A B |

Output: LZ77 = (0,A), (0,B), (2,0), (6,1)

Run-length Burrows–Wheeler transform is an invertible text transformation defined as follows.

Input: text T = BANANA$

1. Build a matrix with the text *rotations* as rows

B	A	N	A	N	A	$
A	N	A	N	A	$	B
N	A	N	A	$	B	A
A	N	A	$	B	A	N
N	A	$	B	A	N	A
A	$	B	A	N	A	N
$	B	A	N	A	N	A

2. Sort the rows

						L
$	B	A	N	A	N	A
A	$	B	A	N	A	N
A	N	A	$	B	A	N
A	N	A	N	A	$	B
B	A	N	A	N	A	$
N	A	$	B	A	N	A
N	A	N	A	$	B	A

3. Apply run-length compression to L = ANNB$AA (the last column)

Output: RLBWT = (1,A), (2,N), (1,B), (1,$), (2,A)

Other (less known) dictionary compressors:

- (run-length) grammars (SLP)
- collage systems
- macro schemes
- word graphs (CDAWG)

APPLICATIONS

Compression: reducing the size of data before archiving or transfer, e.g., over the network. Examples: 7-zip, gzip = LZ77, bzip = RLBWT.

Compressed computation: supporting operations on compressed text, i.e., using a data structure taking space close to dictionary-compressed text. Example operations:

- random access
- pattern matching (counting/reporting) queries

STRING ATTRACTORS

New combinatorial object generalizing all known dictionary compression methods.

Definition ([2])

A set $\Gamma \subseteq [1..n]$ is a *string attractor* of $T \in \Sigma^n$ if every $T[i..j]$ has an occurrence $T[i'..j']=T[i..j]$ with $j'' \in [i'..j']$ for some $j'' \in \Gamma$.

Example:

$$T = \text{CDABCCDABCCA}$$

Let γ^*= the size of minimum attractor. Here we have $\gamma^*=|\Gamma^*|=|\{3,6,10,11\}|$, since the alphabet size $\sigma=4=|\Gamma^*|$, and any Γ must satisfy $|\Gamma| \geq \sigma$.

REDUCTIONS

Theorem: compressors → attractors ([2])

Let $T \in \Sigma^n$ and let α be the output size of any the following dictionary compressors on T:

(1) (RL)SLP, (2) collage system, (3) LZ77,
(4) macro scheme, (5) RLBWT, (6) CDAWG.

Claim: T has a string attractor of size $\mathcal{O}(\alpha)$.

Exercise/example

Starting positions of LZ77 phrases form a string attractor. Why?

A	B	A	B	A B B A B

Thus, dictionary compressors can be viewed as algorithms approximating γ^*. The approximation ratio is a nice way to evaluate the power of a given compression method.

Theorem: attractors → compressors ([2])

Given a string $T \in \Sigma^n$ and a string attractor Γ of size γ for T, we can build

- a macro scheme for T of size $\mathcal{O}(\gamma \log(n/\gamma))$,
- a collage system for T of size $\mathcal{O}(\gamma \log(n/\gamma))$,
- an SLP for T of size $\mathcal{O}(\gamma \log^2(n/\gamma))$.

Consequence: many new (and easier proofs of existing) relations between sizes of dictionary compressors, for example,

$$z \in \mathcal{O}(r \log^2(n/r)),$$

where z (resp. r) is the size of LZ77 (resp. RLBWT).

UNIVERSAL DATA STRUCTURES

Theorem ([2])

Let Γ be a size-γ attractor for $T \in [1..\sigma]^n$. We can store a data structure of $\mathcal{O}(\gamma \operatorname{polylog} n)$ w-bit words that can extract any length-ℓ substring of T in $\mathcal{O}(\ell \log(\sigma)/w + \log n/\log\log n)$ time.

The above reductions transfer this to concrete compressors. One can prove [2], the resulting structures are optimal for LZ77, collage systems, SLPs, RLSLPs, and macro schemes.

Recently, universal indexing was also achieved.

Theorem ([3])

If $T \in \Sigma^n$ has an attractor of size γ, then we can build a data structure of size $\mathcal{O}(\gamma \log(n/\gamma))$ that, given a pattern $P[1..m]$, outputs all its occurrences in T in $\mathcal{O}(m \log n + \text{occ} \log^{\epsilon} n)$ time.

COMPUTATIONAL COMPLEXITY

Theorem ([2])

The decision version of MINIMUMATTRACTOR is NP-complete.

The problem remains difficult (also to approximate) even in the simplified case.

Definition ([2])

$\Gamma \subseteq [1..n]$ is a *k-attractor* of string $T \in \Sigma^n$ if every $T[i..j]$ of length $\leq k$ has an occurrence $T[i'..j']=T[i..j]$ with $j'' \in [i'..j']$ for some $j'' \in \Gamma$.

Theorem ([2])

For any $k \geq 3$:

- Decision version of MINIMUM-k-ATTRACTOR is NP-complete.
- MINIMUM-k-ATTRACTOR is APX-hard.

Good news: MINIMUM-k-ATTRACTOR can be efficiently approximated up to $\mathcal{O}(\log k)$ factor [1, 2].

SUMMARY OF STRING ATTRACTORS

- Common principle underlying all known dictionary compressors
- Rigorous way to measure text compressibility and evaluate/analyze compressors
- Reductions between attractors and compressors imply new (and easy proofs of known) relations between compressibility measures
- Data structures on attractors + above reductions = Universal data structures

OPEN PROBLEMS

- Complexity of MINIMUM-2-ATTRACTOR
- Can MINIMUMATTRACTOR be approximated up to $\mathcal{O}(1)$ factor
- Faster approximation algorithms
- Generalizations (trees, graphs)

REFERENCES

[1] Dominik Kempa, Alberto Policriti, Nicola Prezza, and Eva Rotenberg. String attractors: Verification and optimization. In *arXiv*. 1803.01695.

[2] Dominik Kempa and Nicola Prezza. At the roots of dictionary compression: String attractors. In *Proc. STOC*, 2018.

[3] Gonzalo Navarro and Nicola Prezza. Universal compressed text indexing. In *arXiv*. 1803.09520.