Refined upper bounds on the size of the condensed neighbourhood of sequences

Cedric Chauve, Marni Mishna, France Paquet-Nadeau

Department of Mathematics, Simon Fraser University

August 30, 2021

Overview

Motivation

Expected time complexity analysis for an approximate pattern matching algorithm:
E. W. Myers: A sublinear algorithm for approximate keyword searching. Algorithmica, 1994.
G. Myers: What's Behind Blast. Models and Algorithms for Genome Evolution, 2013.

Overview

Motivation

Expected time complexity analysis for an approximate pattern matching algorithm:
E. W. Myers: A sublinear algorithm for approximate keyword searching. Algorithmica, 1994.
G. Myers: What's Behind Blast. Models and Algorithms for Genome Evolution, 2013.

Sequence neighbourhood

Complexity driven by the maximum size of the neighbourhood of k-mers.

Overview

Motivation

Expected time complexity analysis for an approximate pattern matching algorithm:
E. W. Myers: A sublinear algorithm for approximate keyword searching. Algorithmica, 1994.
G. Myers: What's Behind Blast. Models and Algorithms for Genome Evolution, 2013.

Sequence neighbourhood

Complexity driven by the maximum size of the neighbourhood of k-mers.

Results

Improved upper bound on the maximum size of sequences neighbourhood.

Sequence neighbourhood

Definition

Given a sequence w of length k on an alphabet Σ (with $|\Sigma| = s$), the d-neighbourhood of w, denoted by $N(d, w)$, is the set of all sequences on Σ at Levenshtein distance of w at most d:

$$N(d, w) := \{v \mid d_{Lev}(v, w) \leq d\}.$$

Sequence neighbourhood

Definition

Given a sequence w of length k on an alphabet Σ (with $|\Sigma| = s$), the d-neighbourhood of w, denoted by $N(d, w)$, is the set of all sequences on Σ at Levenshtein distance of w at most d:

$$N(d, w) := \{v \mid d_{Lev}(v, w) \leq d\}.$$

Definition

The condensed neighbourhood of w, denoted by $CN(d, w)$, is the subset of this neighbourhood comprising sequences that have none of their prefixes in the neighbourhood:

$$CN(d, w) := \{v \mid v \in N(d, w) \text{ s.t. } \nexists u \in N(d, w) \text{ prefix of } v\}.$$

Approximate pattern matching

Problem statement

Given a (long) text of length n, a (short) pattern of length p, and an integer $e < p$, find in the text all the occurrences of sequences that are at distance at most e from the pattern (e-approximate pattern occurrences).

Approximate pattern matching

Problem statement

Given a (long) text of length n, a (short) pattern of length p, and an integer $e < p$, find in the text all the occurrences of sequences that are at distance at most e from the pattern (e-approximate pattern occurrences).

Algorithm [Myers, 1994]

- For a well chosen value k, splits the pattern into non-overlapping k-mers.

Approximate pattern matching

Problem statement

Given a (long) text of length n, a (short) pattern of length p, and an integer $e < p$, find in the text all the occurrences of sequences that are at distance at most e from the pattern (e-approximate pattern occurrences).

Algorithm [Myers, 1994]

- For a well chosen value k, splits the pattern into non-overlapping k-mers.
- Compute for each such k-mer its condensed neighbourhood.

Approximate pattern matching

Problem statement

Given a (long) text of length n, a (short) pattern of length p, and an integer $e < p$, find in the text all the occurrences of sequences that are at distance at most e from the pattern (e-approximate pattern occurrences).

Algorithm [Myers, 1994]

- For a well chosen value k, splits the pattern into non-overlapping k-mers.
- Compute for each such k-mer its condensed neighbourhood.
- Search (through a pre-built index) occurrences of the sequences in these neighbourhoods in the text.

Approximate pattern matching

Problem statement

Given a (long) text of length n, a (short) pattern of length p, and an integer $e < p$, find in the text all the occurrences of sequences that are at distance at most e from the pattern (e-approximate pattern occurrences).

Algorithm [Myers, 1994]

- For a well chosen value k, splits the pattern into non-overlapping k-mers.
- Compute for each such k-mer its condensed neighbourhood.
- Search (through a pre-built index) occurrences of the sequences in these neighbourhoods in the text.
- For any such occurrence, try to extend it into an approximate pattern occurrence by dynamic programing.

Approximate pattern matching: expected time complexity

Definition

We denote by $CN(s, k, d)$ the maximum size of a condensed d-neighbourhood over all sequences w of length k on an alphabet Σ of size s:

$$CN(s, k, d) := \max_{w \in \Sigma^k} |CN(d, w)| .$$

[E. W. Myers: A sublinear algorithm for approximate keyword searching. Algorithmica, 1994]
Approximate pattern matching can be solved in expected time

$$O\left(e \cdot CN(s, k, d) \cdot \left(1 + k\frac{n}{s^k}\right) + h \cdot e \cdot p\right)$$

where h is the expected number of e-approximate pattern occurrences, which is optimal for $k = \log_s(n)$, and, for $s = 4$, is sub-linear if $\epsilon := e/p \leq 1/3$.

Problem and motivation

Approximate pattern matching can be solved in expected time

$$O\left(e \cdot CN(s,k,d) \cdot \left(1 + k\frac{n}{s^k}\right) + h \cdot e \cdot p\right).$$

Problem and motivation

Approximate pattern matching can be solved in expected time

$$O\left(e \cdot CN(s,k,d) \cdot \left(1 + k\frac{n}{s^k}\right) + h \cdot e \cdot p\right).$$

Problem statement

Given s, k, d, find an upper-bound for $CN(s,k,d)$, the maximum size of a condensed d-neighbourhood over all sequences of length k on an alphabet of size s.

Problem and motivation

Approximate pattern matching can be solved in expected time

$$O\left(e \cdot CN(s,k,d) \cdot \left(1 + k\frac{n}{s^k}\right) + h \cdot e \cdot p\right).$$

Problem statement

Given s, k, d, find an upper-bound for $CN(s,k,d)$, the maximum size of a condensed d-neighbourhood over all sequences of length k on an alphabet of size s.

Motivation

Improve the sub-linearity window for the expected time complexity of the approximate pattern matching algorithm.

Known results, Myers

Let

$$B(s,k,d,c) := \left(\frac{c+1}{c-1}\right)^k c^d s^d$$

$$c^\star := 1+\sqrt{2}$$

and

$$M(s,k,d) := \frac{c^\star}{c^\star-1} B(s,k,d,c^\star)$$

Then

$$CN(s,k,d) \leq M(s,k,d).$$

Moreover, if

$$pow(\epsilon) := \log_s\left(\frac{c^\star+1}{c^\star-1}\right) + \epsilon \log_s(c^\star) + \epsilon,\ \ k := \lceil \log_s(n) \rceil$$

then for $\epsilon = e/p$,

$$CN(s,k,d) \in O\left(n^{pow(\epsilon)}\right)$$

which leads to the sub-linear expected time complexity if $\epsilon \leq 1/3$.

New results

Conjecture

Let

$$A(s,k,d) := \frac{(2s-1)^d k^d}{d!}.$$

Then

$$CN(s,k,d) \leq A(s,k,d).$$

Experimentally, we obtain the following result.

Proposition

Let $s \in \{1,\ldots,4\}$, $k \in \{1,\ldots,50\}$, $d \in \{1,\ldots,4\}$. Then

$$CN(s,k,d) \leq \frac{(2s-1)^d k^d}{d!}.$$

Application: approximate pattern matching complexity

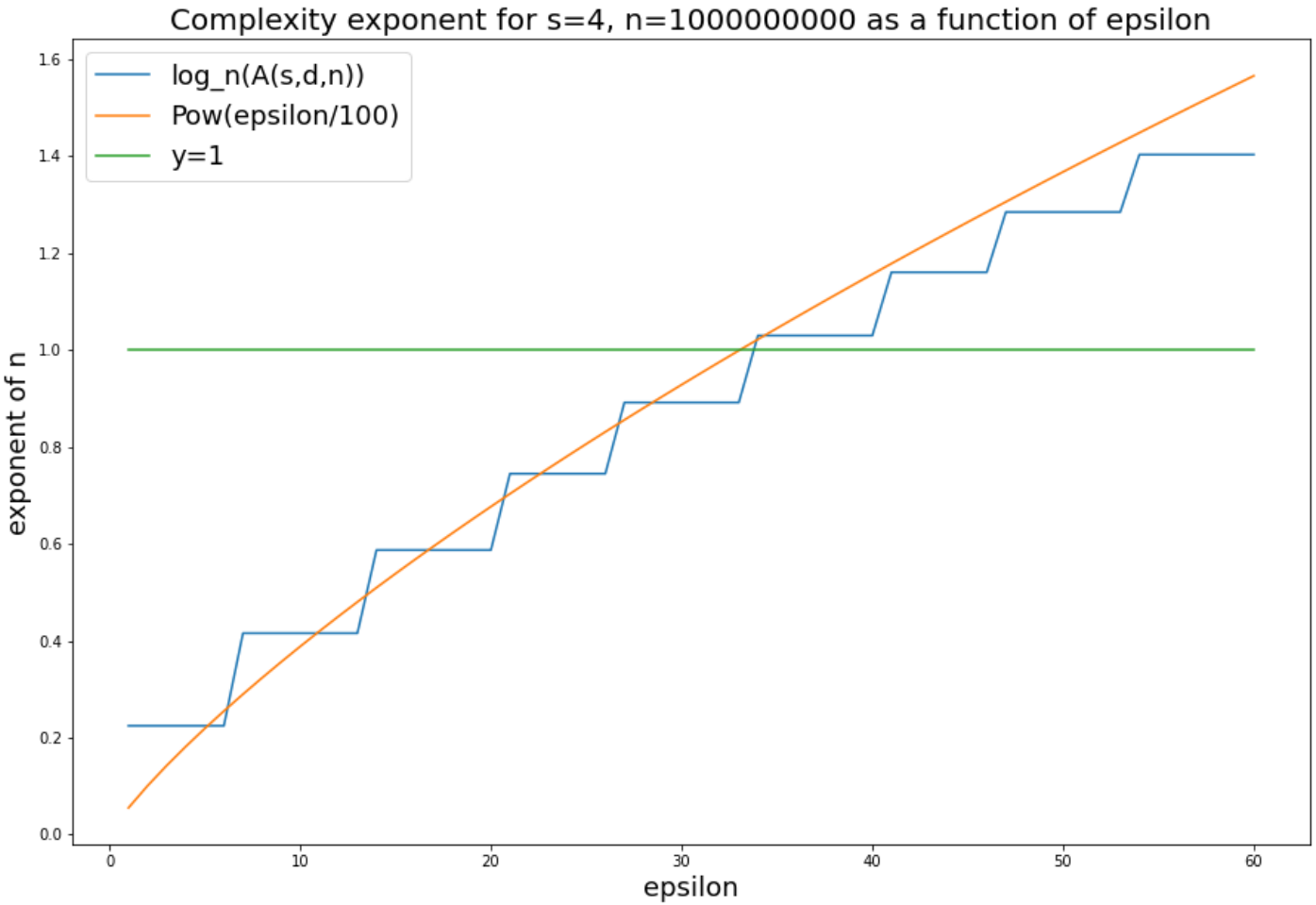


Figure: Illustration of the behaviour of $pow(\epsilon)$ and $\log_n(A(s,k,d))$ for $n = 10^9$ as a function of ϵ, with $s = 4$, $k = \lceil \log_s(n) \rceil$ and $d = \lceil k\epsilon \rceil$.

Methods overview

- Recurrences for counting/generating edit scripts [Myers, 2013], i.e. redundant recurrences for counting/generating condensed neighbourhoods.

Methods overview

- Recurrences for counting/generating edit scripts [Myers, 2013], i.e. redundant recurrences for counting/generating condensed neighbourhoods.
- Translations into ordinary generating functions.

Methods overview

- Recurrences for counting/generating edit scripts [Myers, 2013], i.e. redundant recurrences for counting/generating condensed neighbourhoods.
- Translations into ordinary generating functions.
- Asymptotics analysis of these generating functions: conjectured upper bound for $CN(s, k, d)$.

Methods overview

- Recurrences for counting/generating edit scripts [Myers, 2013], i.e. redundant recurrences for counting/generating condensed neighbourhoods.
- Translations into ordinary generating functions.
- Asymptotics analysis of these generating functions: conjectured upper bound for $CN(s, k, d)$.
- Experimental evaluation: confirmed upper bound in a range of realistic settings.

Recurrences for edit scripts

Lemma 1 [Myers, 2013]

Let $S(s,k,d)$ be defined by the following trivariate recurrence.
If $k \leq d$ or $d = 0$ then $S(s,k,d) := 1$, otherwise

$$S(s,k,d) := \begin{cases} S(s,k-1,d) + (s-1)S(s,k-1,d-1) \\ +(s-1)\sum_{j=0}^{d-1} s^j S(s,k-2,d-1-j) \\ +(s-1)^2 \sum_{j=0}^{d-2} s^j S(s,k-2,d-2-j) \\ +\sum_{j=0}^{d-1} S(s,k-2-j,d-1-j) \end{cases}$$

Let $T(s,k,d) := S(s,k,d) + \sum_{j=1}^{d} s^j S(s,k-1,d-j)$.

Then $CN(s,k,d) \leq T(s,k,d)$.

Generating functions / formal power series

Let the ordinary generating functions of S ad T be

$$\mathsf{S}_{s,d}(z) := \sum_{k=1}^{\infty} S(s,k,d)z^k, \ \mathsf{T}_{s,d}(z) := \sum_{k=1}^{\infty} T(s,k,d)z^k.$$

From the recurrences for S we get

Lemma 2

$$\mathsf{S}_{s,d}(z) = \frac{P_{s,d}(z)}{(1-z)^{d+1}}$$

where $P_{s,d}(z)$ is a polynomial that satisfies $P_{s,d}(1) = (2s-1)^d$.

Lemma 3

$$\mathsf{T}_{s,d}(z) = \mathsf{S}_{s,d}(z) + z\left(\sum_{j=1}^{d-1} s^j \left(\mathsf{S}_{s,d-j}(z) - 1\right)\right) + \frac{s^d}{1-z}.$$

Asymptotics

Using techniques of analytic combinatorics [FO, 1990]:

Lemma 4

Let d be a strictly positive integer. Suppose $P(z)$ is a polynomial such that $P(1) \neq 0$. Then asymptotically, when k becomes large,

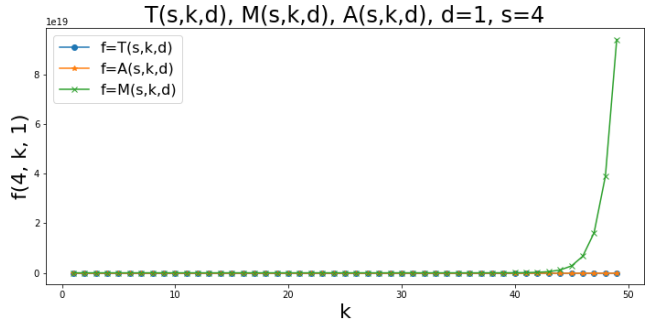
$$[z^k]\frac{P(z)}{(1-z)^{d+1}} \sim \frac{P(1)k^d}{d!}.$$

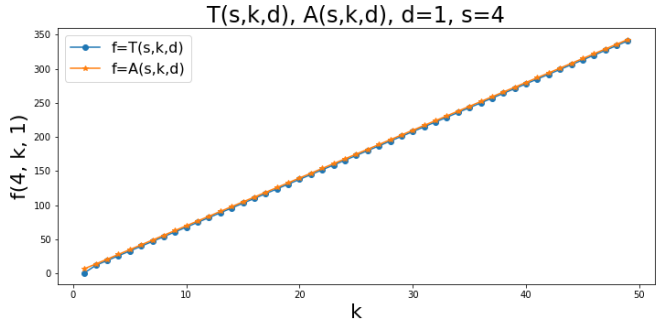
Combined with Lemma 2 and Lemma 3, we can show that

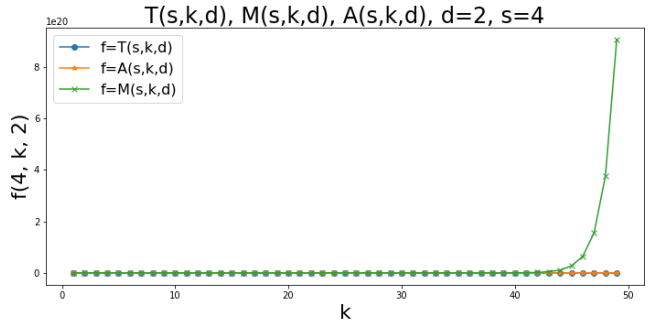
$$\lim_{k\to\infty} \frac{[z^k]\mathrm{T}_{s,d}(z)}{A(s,k,d)} = 1$$

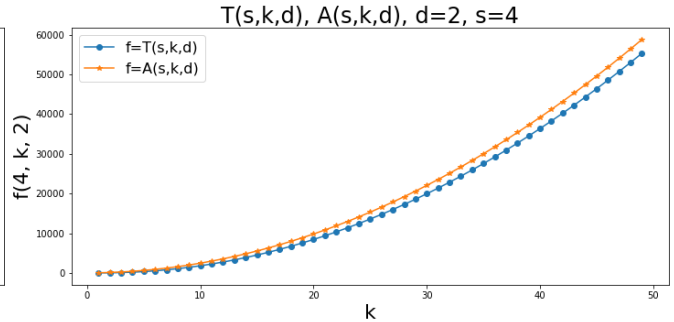
i.e. that asymptotically $T(s,k,d)$ is equivalent to $A(s,k,d)$, our conjectured upped bound for $CN(s,k,d)$.

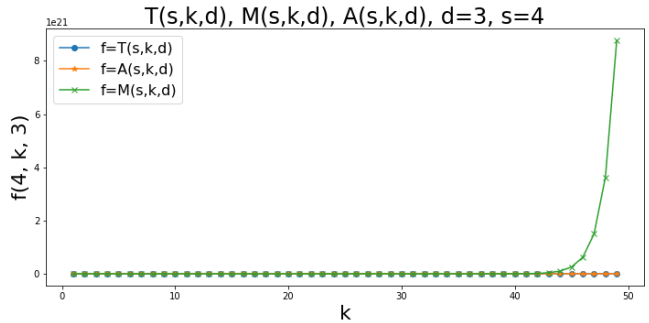
Experimental evaluation

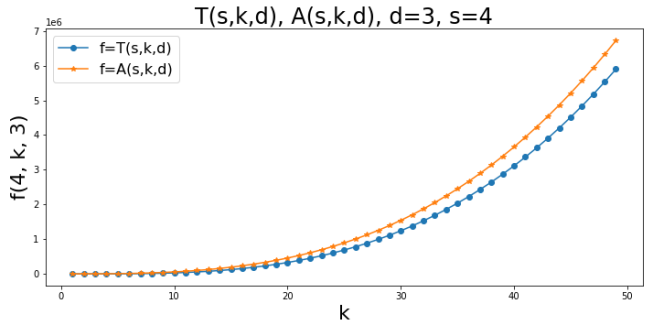


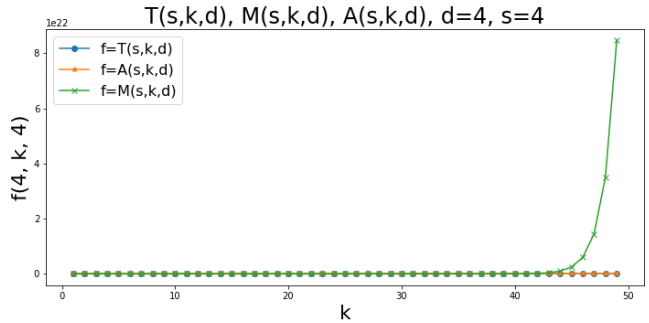


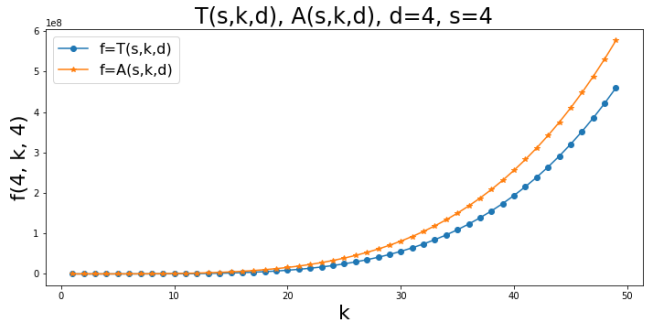












Conclusion

- Motivation: approximate pattern matching (expected) time complexity.

Conclusion

- Motivation: approximate pattern matching (expected) time complexity.
- Open problem (Myers): tighter upper bounds for the maximum size of condensed neighbourhoods.

Conclusion

- Motivation: approximate pattern matching (expected) time complexity.
- Open problem (Myers): tighter upper bounds for the maximum size of condensed neighbourhoods.
- Results: a conjectured tighter upper bound, verified experimentally in some settings relevant for computational biology.

Conclusion

- Motivation: approximate pattern matching (expected) time complexity.
- Open problem (Myers): tighter upper bounds for the maximum size of condensed neighbourhoods.
- Results: a conjectured tighter upper bound, verified experimentally in some settings relevant for computational biology.
- Approximate pattern matching complexity: minor improvement on the sub-linearity window.

Conclusion

- Motivation: approximate pattern matching (expected) time complexity.
- Open problem (Myers): tighter upper bounds for the maximum size of condensed neighbourhoods.
- Results: a conjectured tighter upper bound, verified experimentally in some settings relevant for computational biology.
- Approximate pattern matching complexity: minor improvement on the sub-linearity window.
- Open problem: Improving edit scripts recurrences to reduce redundancy.

References

P. Flajolet and A. M. Odlyzk (1990)

Singularity analysis of generating functions.

SIAM J. Discret. Math., 3(2) 1990, 216 – 240.

E. W. Myers (1994)

A sublinear algorithm for approximate keyword searching.

Algorithmica 12(4/5), 345 – 374.

G. Myers (2013)

What's behing BLAST.

Models and Algorithms for Genome Evolution, Springer, 3 – 15.

F. Paquet-Nadeau (2017)

On the maximum size of condensed sequences neighbourhoods under the Levenshtein distance.

MSc Project Report, Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada.

`https://github.com/cchauve/CondensedNeighbourhoods`