Theoretical Computer Science 871 (2021) 107–120

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the performance of learned data structures ☆

Paolo Ferragina [a], Fabrizio Lillo [b,c], Giorgio Vinciguerra [a,*]

[a] Department of Computer Science, University of Pisa, Italy
[b] Department of Mathematics, University of Bologna, Italy
[c] Scuola Normale Superiore, Pisa, Italy

ARTICLE INFO

Article history:
Received 11 November 2020
Accepted 22 April 2021
Available online 28 April 2021
Communicated by G.F. Italiano

Keywords:
Learned indexes
Data structures
B-trees
Predecessor search

ABSTRACT

A recent trend in algorithm design consists of augmenting classic data structures with machine learning models, which are better suited to reveal and exploit patterns and trends in the input data so to achieve outstanding practical improvements in space occupancy and time efficiency. This is especially known in the context of indexing data structures for big data where, despite few attempts in evaluating their asymptotic efficiency, theoretical results are yet missing in showing that learned indexes are *provably better* than classic indexes, such as B-trees and their variants. In this paper, we present the first mathematically-grounded answer to this problem by exploiting a link with a mean exit time problem over a proper stochastic process which, we show, is related to the space and time complexity of these learned indexes. As a corollary of this general analysis, we show that plugging this result in the (learned) PGM-index, we get a learned data structure which is provably better than B-trees.

1. Introduction

Very recently, the unexpected combination of data structures and Machine Learning (ML) has led to the development of a new area of algorithmic research, called *learned data structures*. The key design idea consists of augmenting — and sometimes even replacing — classic building blocks of data structures, such as arrays, trees or hash tables, with ML models, which are better suited to reveal and exploit patterns and trends in the input data. This feature, orchestrated with proper algorithms, has led to outstanding practical improvements in space occupancy and time efficiency over a plethora of problems and applications, such as databases, search engines, operating systems, sorting algorithms [2].

The most successful example of the interplay between data structures and machine learning is the *indexable dictionary problem*, which asks to store a set S of n keys over a universe $\mathcal{U}$ (e.g. reals, integers, etc.) in an *index structure* that efficiently supports the following query operations:

- $member(x) = \text{TRUE}$ if $x \in S$, FALSE otherwise;
- $predecessor(x) = \max\{y \in S \mid y < x\}$;
- $range(x, y) = S \cap [x, y]$.

☆ This article extends the work presented at ICML 2020 [1]. The new contributions are detailed at the end of the Introduction.

* Corresponding author.
E-mail address: giorgio.vinciguerra@phd.unipi.it (G. Vinciguerra).

https://doi.org/10.1016/j.tcs.2021.04.015

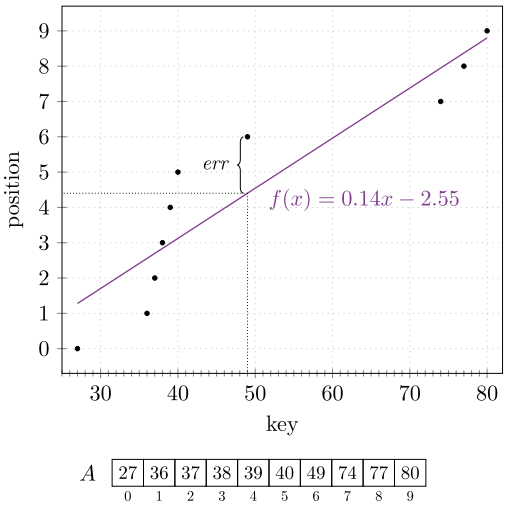

Fig. 1. A set S of ten keys stored in a sorted array A and the corresponding set of points $D = \{(x, rank(x))\}_{x \in S}$ in the Cartesian plane. The linear model f, computed using ordinary least squares on D, estimates that $x = 49$ is in position $r = \lfloor f(x) \rfloor = 4$, but the true rank of x is 6 (hence $err = 2$). We can fix the error incurred by f via a binary search on $A[r - err, r + err]$.

For this problem, many learned data structures (or *learned indexes*, as they are called in this case) have been proposed. Examples include the ones in [3–8] and others surveyed in [2]. The common idea is that *indexes are models* that can be trained to map keys to their location in the sorted S, and this mapping is enough to implement the above queries.

To clarify, let us denote by $rank(x)$ the primitive that returns, for any key $x \in \mathcal{U}$, the number of keys in S which are smaller than x, and let A be the array storing the keys of S in sorted order. Then, $member(x)$ can be implemented by checking whether $A[rank(x)] = x$; $predecessor(x)$ consists of returning $A[rank(x) - 1]$; and $range(x, y)$ consists of scanning the array A from position $rank(x)$ up to the first key larger than y. Given *rank*, we reformulate the indexable dictionary problem as a supervised learning task over a dataset of points $\{(x, rank(x))\}_{x \in S}$ in which we look for a model $f : \mathcal{U} \to \{0, \ldots, n-1\}$ mapping keys to their position in A that minimises the error $|f(x) - rank(x)|$ over all $x \in \mathcal{U}$. The possible presence of an error imposes also the design of proper algorithms that subsequently correct $f(x)$ to get the exact $rank(x)$, and thus answer correctly the query on x. As an example, we can use a binary search in A in a neighbourhood of size $err = \max_{x \in \mathcal{U}} |f(x) - rank(x)|$ around the approximate position $f(x)$. An illustrative example is given in Fig. 1.

We observe that this has been a significant step ahead in the design of indexes because the resulting (learned) data structure answers queries in $O(\log err)$ time plus the cost of computing f, and this might be independent of the number of keys in S. However, we have to notice that although f could be made as much sophisticate as needed to minimise the error, there is a non-negligible side-effect on the overall efficiency of the learned index: the more complex is f, the worse is the query time efficiency and its space occupancy. Consequently, it is not so obvious whether classic index structures, such as B-trees and their variants [9,10], are better or worse than learned indexes.

State-of-the-art learned indexes Starting from the premises above, a significant flow of research has investigated the trade-off among the complexity of the model f, the time to compute and correct the prediction $f(x)$, and the space needed to store f. Ao et al. [3] used simple least-squares linear regression. Kraska et al. [4] proposed a fixed hierarchy of ML models and found that linear regression models were the most effective ones. Other researchers improved these results by proposing dynamic learned indexes based on a Piecewise Linear Approximation (PLA) with a guaranteed maximum error $\varepsilon \geq 1$ (in practice, ε is of the order of hundreds or thousands). In particular, Galakatos et al. [5] orchestrated the segments composing the PLA with a classic B$^+$-tree, while Ferragina and Vinciguerra [7] introduced theoretically more efficient recursive schemes based on *optimal* PLAs, i.e. PLAs with the minimum number of segments.

In practice, learned indexes are fast and occupy a space which is up to several orders of magnitude smaller than classic data structures on several synthetic and real datasets [4–8,11,12]. However, although the authors in [7] showed that querying the PGM-index is as fast as a B-tree with a disk-page size B, it is not yet known whether its space occupancy is *provably better* than the $\Theta(n/B)$ disk pages required by B-trees. In fact, the only known mathematical relation that ties the number n of input keys, the error ε and the size s of the PLA-model (i.e. the number of its segments) is $s \leq n/2\varepsilon$ (see [7]). This shows that the space occupancy of a learned index is never worse than the one taken by a B-tree with disk-page size B (just take $\varepsilon = \Theta(B)$), but it does not theoretically ensure that it is provably more succinct than it.

As a consequence, there is a *methodological gap* in learned index design between what is evident from experiments on several but specific datasets and what research has been able to corroborate with solid mathematical grounds. Bridging this gap amounts to explain from a theoretical perspective the "several orders of magnitude smaller" space occupancy achieved in practice by learned indexes, which in turn consists of showing a dependence in the space complexity between n and s of the form $s = O(n/\varepsilon^c)$, with $c > 1$.

Our contribution We make the first step towards explaining why learned indexes are so effective with respect to traditional indexes.

We obtain this result by considering the gaps between consecutive keys in the sorted input S, and assuming that they are drawn according to a given distribution. This corresponds to the general and realistic scenario of time series data. Then, since the PLA-model at the core of a learned index consists of a sequence of s segments which are at most ε-away (measured along the y-axis) from the points $\{(x, rank(x))\}_{x\in S}$, we turn the problem of determining s into a Mean Exit Time (MET) problem over a stochastic process which estimates how many gaps i have to be drawn from the given distribution until the resulting point (x_i, i) is farther than ε from a segment with a properly defined slope. Now, since this is a fixed slope whereas the algorithm used in Ferragina and Vinciguerra [7] and due to O'Rourke [13] computes the "best" slope, namely the one that induces the longest segment, our result on MET provides a lower bound to the average length of the segments computed by the above (optimal) algorithm, and thus it provides also an upper bound to their number s and to the space taken by the index.

Surprisingly, we show that for any gap distribution with finite mean and variance, the average segment length scales at least *quadratically* with ε which, in turn, means that s decreases as $O(n/\varepsilon^2)$. Specifically, the average segment length is proved to be $\kappa\varepsilon^2$, for a constant $\kappa = \mu^2/\sigma^2$ that depends only on the mean μ and the variance σ^2 of the gap distribution (Theorems 1–3). We then strengthen this result by showing that the upper bound on $s = O(n/\varepsilon^2)$ holds with high probability (Theorem 4). In addition to these key achievements, on the one hand, we specialise Theorem 1 to five well-known distributions (Corollary 1) and, on the other hand, we thoroughly discuss the important case of correlated keys (Section 4). Finally, we perform an extensive set of experiments corroborating that all our theoretical achievements are highly precise.

This leads us to conclude that learned indexes are *provably better than* classic indexing data structures not only in time efficiency but also in space occupancy, and thus they constitute a *robust and effective* indexing choice for modern applications on big data, where space compression and query efficiency are mandatory.

As an illustrative example, let us consider the case of an external-memory setting with disk pages of B keys (typically B is of the order of thousands). Here, a classic B-tree takes $\Theta(n/B)$ space and supports queries in $O(\log_B n)$ I/Os. Given our result, the PGM-index[1] of Ferragina and Vinciguerra [7] answers queries as fast as a B-tree while improving its space to $O(n/B^2)$ with high probability (see Corollary 3).

As a final remark, we note that the preliminary version of this work appeared in [1]. The present contribution includes a new result on repeated keys (Corollary 2), a thorough discussion on the important case of correlated keys (Section 4), an extended Section 6 with new experiments on moving average processes (Fig. 7) and autoregressive processes (Fig. 8) validating the claims of the new Section 4.

2. Preliminaries

We model the sorted input keys $x_0, x_1, \dots$ as a stream generating the gaps $g_1, g_2, \dots$ between consecutive keys so that the ith input key is $x_i = \sum_{j=1}^{i} g_j$ (for convenience, we fix $x_0 = 0$). We assume that the sequence gaps $\{g_i\}_{i\in\mathbb{N}}$ is a realisation of a random process $\{G_i\}_{i\in\mathbb{N}}$, where the G_is are positive, independent and identically distributed (iid) random variables with probability density function (pdf) f_G, mean $\mathbb{E}[G_i] = \mu$ and variance $\mathrm{Var}[G_i] = \sigma^2$. Then, we define the random variables modelling the cumulative sum as $X_i = \sum_{j=1}^{i} G_j$ (for $i = 1, 2, \dots$) and fix $X_0 = 0$.

In this setting, our problem is to find a linear model that approximates the points $(0, 0), (X_1, 1), (X_2, 2), \dots$ in the Cartesian plane within a given maximum error $\varepsilon \geq 1$, measured along the y-axis.

Now, let us consider the two parallel lines $y = mx \pm \varepsilon$, for an m to be chosen later, and the strip $\mathcal{S}$ of height 2ε between them, i.e. $\mathcal{S} = \{(x, y) \mid mx - \varepsilon < y < mx + \varepsilon\}$. As motivated in Section 1, among all the possible choices of the linear model (i.e. values of m), we want the one that maximises $|\mathcal{S}|$. Hence, we are interested in the slope m that maximises the smallest i such that the corresponding point (X_i, i) is outside $\mathcal{S}$. Formally, we are interested in maximising the following random variable:

$$i^* = \min\{i \in \mathbb{N} \mid (X_{i^*}, i^*) \notin \mathcal{S}\}. \tag{1}$$

Since i^* is a random variable, we will find its expected value over different realisations of the sequence $\{X_i\}_{i\in\mathbb{N}}$ as a function of $\varepsilon, m, \mu, \sigma^2$. An example of a realisation is depicted in Fig. 2a.

3. Main results

We recall that the value of i^* depends on the choice of the slope m and the objective of the algorithm is to maximise the expected value of i^*. Our main result is that, in a suitable limit, this maximum is achieved when $m = 1/\mu$, and in this case the number of keys covered scales as $\Theta(\varepsilon^2)$.

More precisely, we can prove the following theorems and corollaries characterising i^* on general or specific distributions of the gaps between consecutive keys in S.

[1] https://pgm.di.unipi.it.

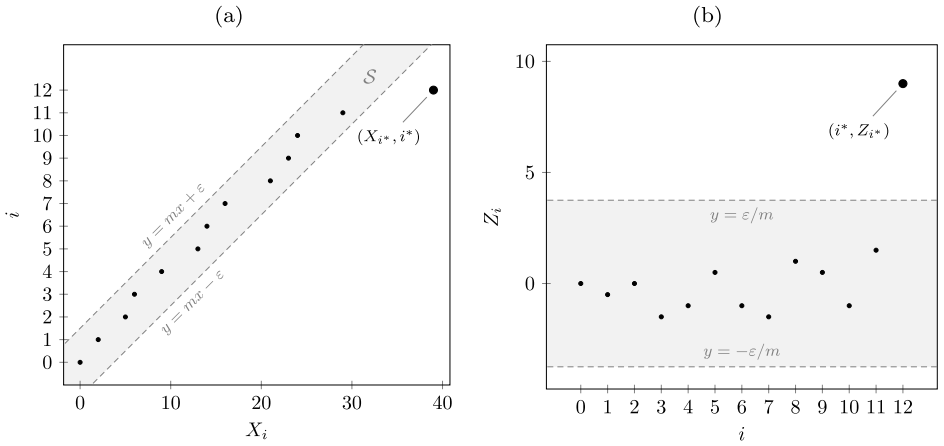

Fig. 2. An example of random walk (a) and the corresponding transformed random walk (b).

Theorem 1. *Given any $\varepsilon \geq 1$ and a sorted set S of n input keys, suppose that the gaps between consecutive keys in S are a realisation of a random process consisting of positive, independent and identically distributed random variables with mean μ and variance σ^2. Then, if ε is sufficiently larger than σ/μ, the expected number of keys covered by a segment with slope $m = 1/\mu$ and maximum error ε is*

$$\frac{\mu^2}{\sigma^2}\varepsilon^2.$$

The following theorem shows that a segment with slope $m = 1/\mu$ is on average the best possible choice in terms of the number of ε-approximated keys.

Theorem 2. *Under the assumptions of Theorem 1, the largest expected number of keys covered by a segment with maximum error ε is achieved with slope $1/\mu$.*

The variance of the length of the segment with slope $m = 1/\mu$ can also be written in closed-form.

Theorem 3. *Under the assumptions of Theorem 1, the variance of the number of keys covered by a segment with slope $1/\mu$ and maximum error ε is*

$$\frac{2}{3}\frac{\mu^4}{\sigma^4}\varepsilon^4.$$

By instantiating some common probability distributions in Theorem 1, it follows the next key corollary.

Corollary 1. *Under the assumptions of Theorem 1, the expected number of keys covered by a segment is:*

- $3\frac{(a+b)^2}{(b-a)^2}\varepsilon^2$ *if the gaps are iid and uniformly distributed with minimum a and maximum b.*
- $\alpha(\alpha-2)\varepsilon^2$ *if the gaps are iid and Pareto (power law) distributed with minimum value $k > 0$ and shape parameter $\alpha > 2$.*
- $\varepsilon^2/(e^{\sigma^2}-1)$ *if the gaps are iid and lognormally distributed with mean μ and variance σ^2.*
- ε^2 *if the gaps are iid and exponentially distributed with rate $\lambda > 0$.*
- $k\varepsilon^2$ *if the gaps are iid and gamma distributed with shape parameter $k > 0$ and scale parameter $\theta > 0$.*

As the next result shows, the number of keys covered by a segment scales as $\Theta(\varepsilon^2)$ even when S contains repeated keys, i.e. when some gaps are equal to zero.

Corollary 2. *Given any $\varepsilon \geq 1$ and a sorted set S of input keys, suppose that the gap between any two consecutive keys in S is zero with probability p, and that, with the remaining probability $(1-p)$, the gap is drawn from a distribution with mean μ and variance σ^2. Define*

$$\kappa^2 = \frac{(1-p)\mu^2}{\sigma^2 + p\mu^2}.$$

If ε is sufficiently larger than $1/\kappa$, the expected number of keys covered by a segment with slope $m = 1/(\mu(1-p))$ and maximum error ε is $\kappa^2\varepsilon^2$.

Finally, we can show that the number of segments s which have slope $m = 1/\mu$ and guarantee a maximum error ε on a stream of length n is very concentrated around $\Theta(n/\varepsilon^2)$.

Theorem 4. *Under the assumptions of Theorem 1, the number of segments s needed to cover a stream of length n with error at most ε converges almost surely to*

$$\frac{\sigma^2}{\mu^2}\frac{n}{\varepsilon^2},$$

and the relative standard deviation of s converges to zero as $1/\sqrt{n}$ when $n \to \infty$.

In the following, given this last result, we will say that the number of segments s is $O(n/\varepsilon^2)$ "with high probability" [14].

The above theorems are based on the assumption that gaps are independent and identically distributed. In applications this condition might not be true, and thus it is important to assess whether our results hold, even in some asymptotic regime, when gaps are autocorrelated. We answer this question affirmatively in Section 4.

3.1. Proof of Theorem 1

Let us consider the Cartesian plane introduced in Section 2. By swapping abscissas and ordinates of the plane, the equation of the two parallel lines becomes $y = (x \pm \varepsilon)/m$ (x and y are the new coordinates), and the sequence of points becomes $\{(i, X_i)\}_{i\in\mathbb{N}}$. This sequence describes a discrete-time random walk with iid increments $G_i = X_i - X_{i-1}$. The main idea of the proof is to determine the Mean Exit Time (MET) of the random walk out of the strip delimited by the two lines above, i.e. the mean of

$$i^* = \min\left\{ i \in \mathbb{N} \;\middle|\; X_i > \frac{i}{m} + \frac{\varepsilon}{m} \vee X_i < \frac{i}{m} - \frac{\varepsilon}{m} \right\}. \tag{2}$$

To simplify the analysis, we consider the following transformed random walk, where we use the equality $X_i = \sum_{j=1}^{i} G_j$ and set $W_j = G_j - 1/m$:

$$Z_i = X_i - \frac{i}{m} = \sum_{j=1}^{i}\left(G_j - \frac{1}{m}\right) = \sum_{j=1}^{i} W_j.$$

The objective (2) can be thus rewritten as

$$i^* = \min\{i \in \mathbb{N} \mid Z_i > \varepsilon/m \vee Z_i < -\varepsilon/m\},$$

which is the exit time of the transformed random walk $\{Z_i\}_{i\in\mathbb{N}}$ whose increments W_j are iid with mean $\mathbb{E}[W_j] = \mathbb{E}[G_j] - 1/m = \mu - 1/m$, variance $\mathrm{Var}[W_j] = \mathrm{Var}[G_j] = \sigma^2$ and pdf $f_W(w) = f_G(w + 1/m)$.

An example of this transformed random walk is depicted in Fig. 2b above.

Let $T(z_0) = \mathbb{E}[i^* \mid Z_0 = z_0$ be the MET if the random walk $\{Z_i\}_{i\in\mathbb{N}}$ starts from z_0. In our case, it starts from $z_0 = y_0 - 0/m = 0$ (since $y_0 = 0$). It is well known [15,16] that $T(z)$ satisfies the Fredholm integral equation of the second kind $T(z_0) = 1 + \int_{-\varepsilon/m}^{\varepsilon/m} f_W(z - z_0)\, T(z)\, dz$, which for our problem can be rewritten as

$$T(z_0) = 1 + \int_{-\varepsilon/m}^{\varepsilon/m} f_G\left(z - z_0 + \frac{1}{m}\right) T(z)\, dz. \tag{3}$$

While solving exactly the integral equation (3) is in general impossible, it is possible to give a general limiting result when ε is sufficiently large. More specifically, when $m = 1/\mu$, the transformed random walk Z_i has increments with zero mean and variance equal to σ^2, and the boundaries of the strip are at $\pm\varepsilon\mu$. When $\sigma \ll \varepsilon\mu$ or equivalently $\varepsilon \gg \sigma/\mu$, the Central Limit Theorem tells us that the distribution of the position of the random walker is Normal because many steps are necessary to reach the boundary. In this case, the transformed random walk converges to a Brownian motion (or Wiener process) in continuous time [17].[2]

[2] A mathematical more precise but equivalent statement can be done using the Donsker's theorem [18].

Now, it is well known [17] that for a driftless Wiener process the MET out of an interval $[-\delta/2, \delta/2]$ is

$$T(x) = \frac{(\delta/2)^2 - x^2}{\sigma^2}, \tag{4}$$

where $x \in [-\delta/2, \delta/2]$ is the value of the process at the initial time. In our case, $x = 0$ and $\delta = 2\varepsilon/m = 2\varepsilon\mu$, thus we finally have the statement of the theorem.

3.2. Proof of Theorem 2

Using an approach similar to the one in Section 3.1, if $m \neq 1/\mu$, the transformed random walk $Z_i = X_i - 1/m = \sum_{j=1}^{i} W_j$ has increments with mean $d \equiv \mathbb{E}[W_j] = \mu - 1/m$ and variance σ^2 (see the previous section). For large ε the process converges to a Brownian motion with drift. The MET out of an interval $[-\delta/2, \delta/2]$ for a Brownian motion with drift coefficient $d \neq 0$ and diffusion rate σ can be proved to be

$$T(0) = \frac{\delta}{2d}\left[\frac{e^{d\delta/\sigma^2} + e^{-d\delta/\sigma^2} - 2}{e^{d\delta/\sigma^2} - e^{-d\delta/\sigma^2}}\right]. \tag{5}$$

To show this, we use the known fact (see [17, §5.2.7]) that the MET $T(x)$ out of an interval $[-\delta/2, \delta/2]$ of a Brownian motion with drift d and diffusion rate σ starting at position x satisfies the differential equation

$$d\,\frac{dT(x)}{dx} + \frac{\sigma^2}{2}\frac{d^2T(x)}{dx^2} = -1,$$

with the boundary conditions

$$T(\delta/2) = T(-\delta/2) = 0.$$

The solution of this Cauchy problem is

$$T(x) = \frac{\delta - 2x}{2d} + \frac{\delta}{d}\left[\frac{e^{-d\delta/\sigma^2} - e^{-2dx/\sigma^2}}{e^{d\delta/\sigma^2} - e^{-d\delta/\sigma^2}}\right].$$

If the random walker starts at $x = 0$, this expression becomes $T(0)$ of Equation (5).

Clearly, by taking the limit $d \to 0$ (i.e. $\mu \to 1/m$) in (5), one obtains Equation (4). As in the proof of Theorem 1, we have $\delta = 2\varepsilon/m$, thus substituting it in the equation above we get

$$T(0) = \frac{\varepsilon}{md}\tanh\left(\frac{\varepsilon d}{m\sigma^2}\right).$$

It is easy to see that the maximum of $T(0)$ is achieved for $d = 0$, i.e. when $m = 1/\mu$, which is exactly the setting considered in Theorem 1.

3.3. Proof of Corollary 2

Under the assumptions of the corollary, the gaps G_j have mean value $\tilde{\mu} = (1-p)\mu$ and variance $\tilde{\sigma}^2 = (1-p)(\sigma^2 + \mu^2) - (1-p)^2\mu^2$, thus the increments $W_j = G_j - 1/m = G_j - \tilde{\mu}$ of the transformed random walk have zero mean and variance $\tilde{\sigma}^2$. Using Theorem 1 and Theorem 2, we conclude that the optimal slope is $m = 1/\tilde{\mu}$ and the expected number of keys is $(\tilde{\mu}^2/\tilde{\sigma}^2)\,\varepsilon^2$, i.e. the thesis.

3.4. Proof of Theorem 3

Following Gardiner [17, Equation 5.2.156], the second moment $T_2(x)$ of the exit time of a Brownian motion with diffusion rate σ starting at x is the solution of the partial differential equation

$$-2T(x) = \frac{\sigma^2}{2}\,\partial_x^2\,T_2(x),$$

where $T(x)$ is the MET out of an interval $[-\delta/2, \delta/2]$ (see Equation (4)), with boundary conditions $T_2(\pm\delta/2) = 0$. Solving for $T_2(x)$, we get

$$T_2(x) = \frac{x^4 - 2\delta^2x^2/3 + 5\delta^4/16}{3\sigma^4}.$$

Setting $x = 0$ and $\delta = 2\varepsilon/m = 2\varepsilon\mu$, we find that the second moment of the exit time starting at $x = 0$ is

$$T_2(0) = \frac{5}{3}\frac{\mu^4}{\sigma^4}\varepsilon^4,$$

thus

$$T_2(0) - [T(0)]^2 = \frac{2}{3}\frac{\mu^4}{\sigma^4}\varepsilon^4.$$

3.5. Proof of Theorem 4

Consider a process that starts a new segment $j+1$ as soon as the current one j cannot cover more than i_j^* keys without exceeding the error ε (see Equation (2)). We define the total number of segments s on a stream of length n as

$$s(n) = \sup\{k \geq 1 \mid S_k \leq n\},$$

where $S_k = i_1^* + i_2^* + \cdots + i_k^*$.

We notice that $\{s(n)\}_{n\geq 0}$ is a renewal counting process of non-negative integer random variables $i_1^*, \ldots, i_k^*$, which are independent due to the lack of memory of the random walk. Let $\mathbb{E}[i_j^*] = 1/\lambda$ and $\mathrm{Var}[i_j^*] = \varsigma^2$. It is well known [19, §2.5.2] that $\mathbb{E}[s(n)] = \lambda n + O(1)$ as $n \to \infty$, $\mathrm{Var}[s(n)] = \varsigma^2\lambda^3 n + o(n)$ as $n \to \infty$, and that $s(n)/n \xrightarrow{\text{a.s.}} \lambda$. In our case (see Theorems 1 and 3), it holds

$$\frac{1}{\lambda} = \frac{\mu^2}{\sigma^2}\varepsilon^2 \quad \text{and} \quad \varsigma^2 = \frac{2}{3}\frac{\mu^4}{\sigma^4}\varepsilon^4,$$

hence $s(n)/n \xrightarrow{\text{a.s.}} \lambda = (\sigma/(\mu\varepsilon))^2$. Finally, the following ratio converges to zero as $n \to \infty$:

$$\frac{\sqrt{\mathrm{Var}[s(n)]}}{\mathbb{E}[s(n)]} \to \sqrt{\frac{\varsigma^2\lambda}{n}} = \sqrt{\frac{2}{3}}\frac{\mu\varepsilon}{\sigma}\frac{1}{\sqrt{n}}.$$

4. A conjecture for the case of correlated keys

In this section, we study the case in which the independence assumption of Section 3 is waived. Specifically, we study a random process $\{G_i\}_{i\in\mathbb{N}}$ generating gaps that consist of positive and identically distributed random variables with mean $\mathbb{E}[G_i] = \mu$, variance $\mathrm{Var}[G_i] = \sigma^2$, and covariances $C(\ell) = \mathrm{Cov}[G_i, G_{i+\ell}] = \mathbb{E}[G_i G_{i+\ell}] - \mu^2$ for any lag $\ell \geq 1$. As usual, we define the random variables modelling the ith input key X_i as the cumulative sum $X_i = \sum_{j=1}^{i} G_j$ (for $i = 1, 2, \ldots$) and fix $X_0 = 0$. It is easy to see that their mean is $\mathbb{E}[X_i] = i\mu$ and their variance is

$$\begin{aligned}
\mathrm{Var}[X_i] &= \sum_{j=1}^{i}\sum_{k=1}^{i}\mathrm{Cov}[G_j, G_k] \\
&= \sum_{j=1}^{i}\mathrm{Var}[G_j] + 2\sum_{j<i}\mathrm{Cov}[G_j, G_i] \\
&= i\sigma^2 + 2\left[(i-1)C(1) + (i-2)C(2) + \cdots + C(i-1)\right] \\
&= i\sigma^2 + 2\sum_{\ell=1}^{i-1}(i-\ell)C(\ell) \\
&= i\sigma^2\left[1 + 2\sum_{\ell=1}^{i-1}\left(1 - \frac{\ell}{i}\right)\rho(\ell)\right],
\end{aligned} \tag{6}$$

where $\rho(\ell) \equiv C(\ell)/\sigma^2$ is the autocorrelation function.

When i is much larger than the time scale ℓ_0 after which the autocorrelation is negligible (the "memory" of the process), the ℓ/i term in round brackets can be neglected.

Hence, as $i \gg \ell_0$ we get the approximation

$$\mathrm{Var}[X_i] \simeq i\left(\sigma^2 + 2\sum_{\ell=1}^{\ell_0}C(\ell)\right) = i\sigma^2\left(1 + 2\sum_{\ell=1}^{\ell_0}\rho(\ell)\right). \tag{7}$$

Thus for large i, the process becomes exactly diffusive (i.e. the $\mathrm{Var}[X_i]$ increases linearly with i) as in a random walk with iid increments and effective diffusion rate $\sigma^2(1 + 2\sum_{\ell=1}^{\ell_0}\rho(\ell))$. We therefore state the following conjecture:

Conjecture 1. *If ε is sufficiently large, the random walk will make a large number i of steps, and thus it will satisfy the condition of Theorem 1 with mean $\mathbb{E}[X_i] = i\mu$ and variance $\mathrm{Var}[X_i]$ given by Equation (7), giving for the expected number of keys covered by a segment with slope $m = 1/\mu$ and maximum error ε the value*

$$\frac{1}{1+2\sum_{\ell=1}^{\ell_0}\rho(\ell)}\frac{\mu^2}{\sigma^2}\varepsilon^2 \approx \frac{1}{1+2\sum_{\ell=1}^{\infty}\rho(\ell)}\frac{\mu^2}{\sigma^2}\varepsilon^2.$$

In the above approximation, we have extended the sum at the denominator from $\ell \le \ell_0$ to $\ell \to +\infty$, since by construction $\rho(\ell)$ is negligible (or zero) when $\ell > \ell_0$. The above formula shows that in a random walk with correlated increments the expected number of keys increases quadratically with ε, as in the random walk with iid increments. The main difference is the prefactor multiplying $\mu^2\varepsilon^2/\sigma^2$: when the increments are positively correlated ($\rho(\ell) > 0$), the prefactor is smaller than one, i.e. the expected number of keys is smaller than for a random walk with iid increments.

In order to dig into the significance of this observation, let us study few specific, yet realistic, examples. In Section 6, we provide numerical evidence that the above conjecture describes accurately the expected number of keys in one of the examples presented below.

Example 1 *(Moving-average process).* Let us consider a process $\{U_i\}_{i\in\mathbb{N}}$ of positive iid variables U_i having mean $\mathbb{E}[U_i] = \mu_U$ and variance $\mathrm{Var}[U_i] = \sigma_U^2$. We then assume that a gap G_i is generated by a convolution of ℓ_0 variables U_i as $G_i = \sum_{k=1}^{\ell_0}\phi_i U_i$, where ϕ_i are positive weights, i.e. that $\{G_i\}_{i\in\mathbb{N}}$ is a *moving-average process of order* ℓ_0.

It is immediate to show that $\mu := \mathbb{E}[G_i] = \mu_U\sum_{k=1}^{\ell_0}\phi_k$ and $\sigma^2 := \mathrm{Var}[G_i] = \sigma_U^2\sum_{k=1}^{\ell_0}\phi_k^2$. Moreover, it holds

$$C(\ell) = \mathrm{Cov}[G_i, G_{i+\ell}] = \begin{cases} \sigma_U^2\sum_{k=1}^{\ell_0-\ell}\phi_k\phi_{k+\ell} & \text{if } \ell < \ell_0 \\ 0 & \text{otherwise.} \end{cases}$$

In the special case of a flat filter, i.e. $\phi_i = 1$ for any i, it is easy to see that $\mu = \ell_0\mu_U$, $\sigma^2 = \ell_0\sigma_U^2$, and that $C(\ell) = \sigma_U^2(\ell_0 - \ell)$ if $\ell < \ell_0$ and $C(\ell) = 0$ otherwise. By plugging the last definitions of σ^2 and $C(\ell)$ into Equation (6), we obtain

$$\begin{aligned}
\mathrm{Var}[X_i] &= i\ell_0\sigma_U^2 + 2i\sum_{\ell=1}^{\ell_0-1}\left(1-\frac{\ell}{i}\right)\sigma_U^2(\ell_0-\ell) \\
&\le i\ell_0\sigma_U^2 + 2i\sigma_U^2\sum_{\ell=1}^{\ell_0-1}(\ell_0-\ell) \\
&= i\left(\ell_0\sigma_U^2 + 2\sigma_U^2\frac{\ell_0(\ell_0-1)}{2}\right) \\
&= i\ell_0^2\sigma_U^2 \\
&= i\ell_0\sigma^2.
\end{aligned}$$

To mimic the statement of Theorem 1 in the special case we just described, we conjecture that if ε is sufficiently larger than $\sigma\sqrt{\ell_0}/\mu$, the expected number of keys covered by a segment with slope $1/\mu$ and maximum error ε is at least

$$\frac{\mu^2}{\sigma^2\ell_0}\varepsilon^2. \tag{8}$$

□

Example 2 *(Autoregressive process).* This second example assumes that the gaps follow an AR(1) process, i.e. $G_i = \varphi G_{i-1} + \eta_i$, where φ is a real parameter, and η_i is a white noise with mean μ_η and variance σ_η^2.

It is well known [20] that when $|\varphi| < 1$

$$\mu := \mathbb{E}[G_i] = \frac{\mu_\eta}{1-\varphi}, \quad \sigma^2 := \mathrm{Var}[G_i] = \frac{\sigma_\eta^2}{1-\varphi^2}, \quad \text{and } C(\ell) = \sigma^2\varphi^\ell.$$

The autocorrelation function decays exponentially to zero, thus in this case $\rho(\ell)$ is never zero, even if the time scale of the process is finite and related to $|\varphi|$. The variance of the random walk is

$$\begin{aligned}\mathrm{Var}[X_i] &= i\sigma^2\left[1+2\sum_{\ell=1}^{i}\left(1-\frac{\ell}{i}\right)\varphi^\ell\right]\\ &= i\sigma^2\left(1+2\frac{\varphi}{1-\varphi}-2\frac{\varphi-\varphi^{i+1}}{(1-\varphi)^2 i}\right).\end{aligned}$$

When i is very large the last (negative) term in brackets becomes negligible, and the variance of the random walk may be approximated by

$$\mathrm{Var}[X_i] \simeq i\sigma^2\frac{1+\varphi}{1-\varphi}.$$

To mimic the statement Theorem 1 in the special case we just described, we conjecture that if ε is sufficiently larger than $\sqrt{\frac{1+\varphi}{1-\varphi}}\frac{\sigma}{\mu}$, the expected number of keys covered by a segment with slope $1/\mu$ and maximum error ε is

$$\frac{1-\varphi}{1+\varphi}\frac{\mu^2}{\sigma^2}\varepsilon^2. \tag{9}$$

□

5. Some implications

We now mention some key implications of Theorems 1 and 4 that go beyond the realm of learned indexes. The computation of a Piecewise Linear Approximation (PLA) has indeed gathered attention in many other fields, such as computational geometry, time series approximation, image processing, database, geographic information systems, machine learning, etc., with a variety of error definitions, constraints, and proposed algorithms (see e.g. [13,21–24] and refs therein). Theorem 4 can eventually give an estimate of the number of segments computed by these algorithms when they are given a dataset satisfying the assumptions of Theorem 1. In particular, taking as a reference the linear time algorithm for computing the optimal (i.e. minimum-sized) PLA P with maximum error ε, that we could trace back to O'Rourke [13], we have that the number of segments composing P is bounded above by $O(n\sigma^2/(\mu\varepsilon)^2)$ with high probability (by Theorem 4).

In light of our new results, we can strengthen the solution of Ferragina and Vinciguerra [7] to the indexable dictionary problem by showing that their PGM-index achieves the same query time complexity of a B-tree, but within an improved space occupancy of $O(n/B^2)$ (versus the $\Theta(n/B)$ space of a B-tree).

Corollary 3. *Let S and n be as in Theorem 1. The PGM-index data structure built on S uses $O(n/B^2)$ space with high probability, and answers rank, membership and predecessor queries in optimal $O(\log_B n)$ I/Os, where B is block size of the external-memory model. Range queries are answered in extra (optimal) $O(K)$ time and $O(K/B)$ I/Os, where K is the number of keys satisfying the range query.*

Proof. Since the PGM-index is built on the s segments computed by the optimal algorithm of O'Rourke [13], then the minimality of s and Theorem 4 imply that $s = O(n/\varepsilon^2)$ with high probability (by Theorem 4). Substituting this bound into Theorem 1 of [7] and setting $\varepsilon = \Theta(B)$ the claim follows. □

6. Experiments

We start with an experiment aimed at validating our main result (Theorem 1).[3] We generated 10^7 random streams of gaps for each of the following distributions: Uniform, Pareto, Lognormal, Exponential/Gamma. For each generated stream S, we picked an integer ε in the range $[1, 2^8]$, which contains the values that were shown to be the most effective in practice for the learned index of [7]. Then, we ran the following PLA-algorithms with input parameters ε and S:

MET. This is the algorithm that fixes the slope of a segment to $1/\mu$ and stops when the next point of S is outside the strip of width 2ε, see Equation (1). This corresponds to the random process we used to prove Theorem 1.

OPT. This is the algorithm that constructs the optimal PLA-model [13] used in the PGM-index of [7]. This algorithm computes the segment (of any slope and intercept) that ε-approximate the longest prefix of S.

We analysed the length of the segments computed by the two previous algorithms, that is, the index of the first key that causes the algorithm to stop because the (vertical) distance of the point from the segment is larger than ε. We plot in Fig. 3 the mean and the standard deviation of these segment lengths. The figure shows that the theoretical mean segment length

[3] The code to reproduce the experiments is available at https://github.com/gvinciguerra/Learned-indexes-effectiveness. The experiments were run on an Intel Xeon Gold 6132 CPU.

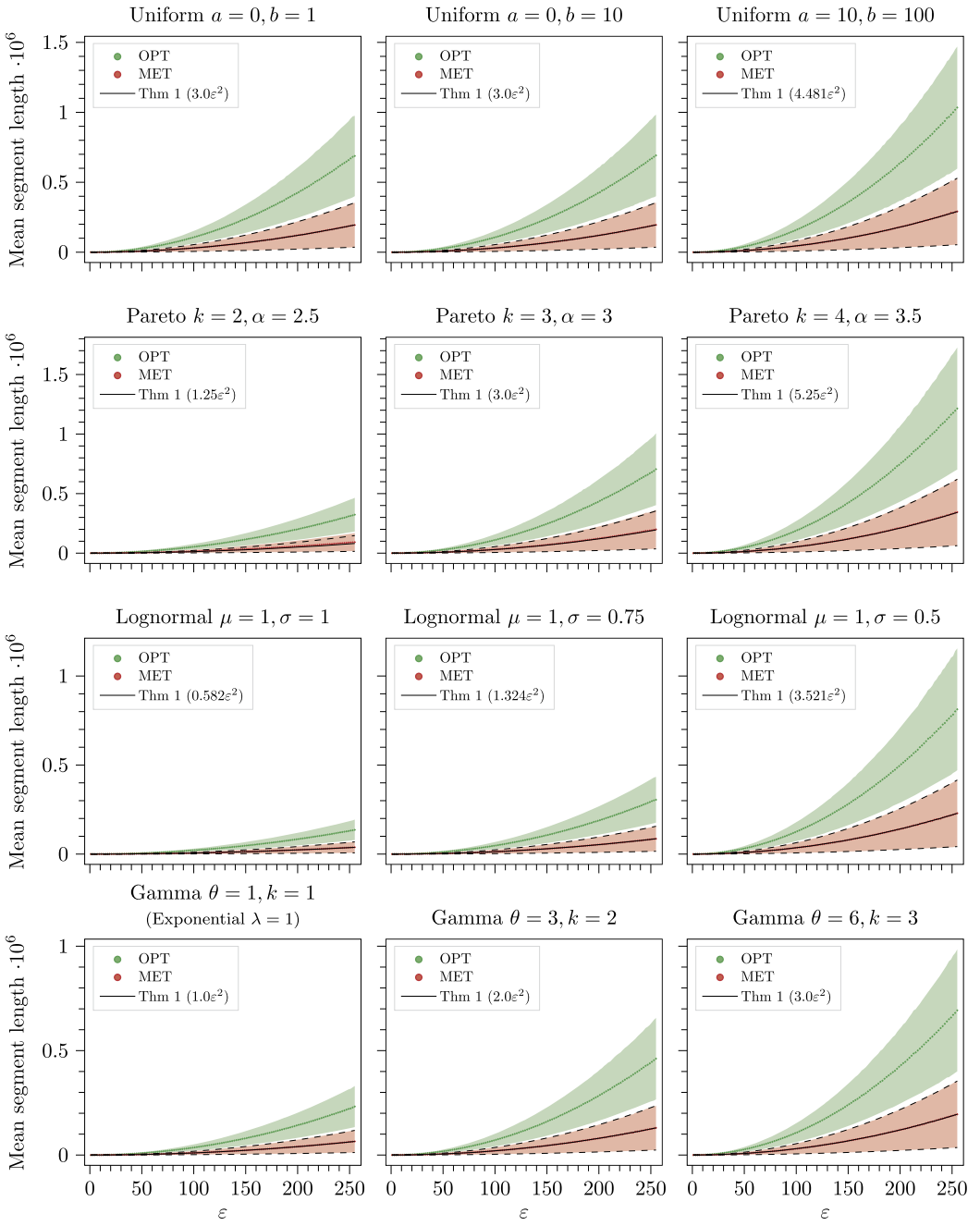

Fig. 3. We consider four gap distributions – Uniform, Pareto, Lognormal, and Gamma – with various parameter settings. We plot the formula $(\mu^2/\sigma^2)\,\varepsilon^2$ given in Theorem 1 with a solid black line and the Mean Exit Time (MET) of the experimented random walk with red points. The figure shows that they overlap, thus the formula stated in Theorem 1 accurately predicts the experimented MET. The figure also shows the performance of the algorithm OPT with green points. The shaded regions represent the standard deviation. The improvement of OPT with respect to MET is evident, indicating that OPT is more robust to outliers. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

computed according to Corollary 1 (hence the formula $(\mu^2/\sigma^2)\,\varepsilon^2$), depicted as a solid black line, accurately describes the experimented algorithm MET, depicted as red points, over all tested distributions (just observe that the solid black line overlaps the red points). Moreover, the standard deviation of the exit time, depicted as a shaded red region, follows the corresponding bound proved in Theorem 3 and depicted as two dashed black lines in each plot. So our theoretical analysis of Theorem 1 is tight.

Not surprisingly, the plots show also that OPT performs better than MET. This is because MET fixes the slope of a segment to $1/\mu$, while OPT optimally adapts to each sequence of points given in input by choosing the slope that maximises the number of points covered by a segment. Thus it is more robust to outliers and hence can find longer segments.

Table 1
The range of slopes found by algorithm OPT in the experiments of Fig. 3. Notice that these ranges are centred and close to $1/\mu$, which is the theoretical slope that maximises the MET of the random walk depicted in Fig. 2a.

Distribution	Parameters	$1/\mu$	Avg. slope range
Uniform	$a=0, b=1$	2	[2.000, 2.002]
Uniform	$a=0, b=10$	0.2	[0.200, 0.200]
Uniform	$a=10, b=100$	0.018	[0.018, 0.018]
Pareto	$k=2, \alpha=2.5$	0.3	[0.300, 0.301]
Pareto	$k=3, \alpha=3$	0.222	[0.222, 0.222]
Pareto	$k=4, \alpha=3.5$	0.179	[0.179, 0.179]
Lognormal	$\mu=1, \sigma=0.5$	0.325	[0.325, 0.325]
Lognormal	$\mu=1, \sigma=0.75$	0.278	[0.278, 0.278]
Lognormal	$\mu=1, \sigma=1$	0.223	[0.223, 0.224]
Exponential	$\lambda=1$	1	[1.000, 1.003]
Gamma	$\theta=3, k=2$	0.167	[0.167, 0.167]
Gamma	$\theta=6, k=3$	0.056	[0.056, 0.056]

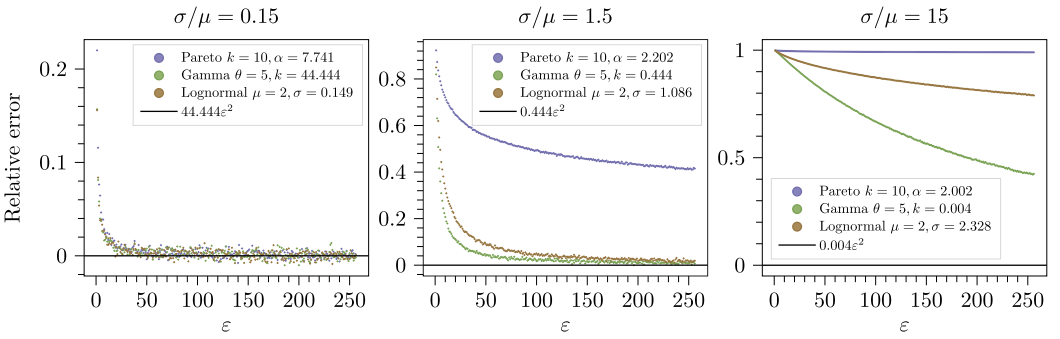

Fig. 4. Three plots for three different settings of the ratio σ/μ for the distributions: Pareto, Gamma and Lognormal. We plot the relative error between the formula $(\mu^2/\sigma^2)\,\varepsilon^2$ of Theorem 1 and the experimented MET. Notice how the fat-tail of the distributions affects the accuracy of the formula with respect to MET, as commented in the text.

Overall this first experiment entails that learned indexes (and, in particular, the learned index based on an optimal use of linear models, see [7]) use a space that decreases as fast as $O(n/\varepsilon^2)$, where n is the number of keys in the dataset and ε is the maximum error admitted by the learned index (Corollary 3).

The second experiment analysed the slopes found by OPT over the sequence of points generated according to the previous experiment, and averaged over ε. We compared them to the fixed slope $1/\mu$ of MET. Table 1 clearly shows that these slopes are centred around $1/\mu$, thus confirming the result of Theorem 2 that $1/\mu$ is the best slope on average.

The third experiment was devoted to studying the *accuracy* of the approximation to the mean exit time provided by the formula $(\mu^2/\sigma^2)\,\varepsilon^2$ under the assumption "with ε sufficiently larger than σ/μ" present in the statement of Theorem 1. To this end, we properly set the distribution parameters to obtain a ratio σ/μ in $\{0.15, 1.5, 15\}$. We plot in Fig. 4 the relative error between the experimented MET (i.e. the empirical mean segment length) and the formula above, as ε grows from 1 to 2^8. For the left plot, we notice that for all the distributions the relative error converges soon to 0 (here, the ratio σ/μ is very small compared to ε). In the middle plot, the convergence is fast for Gamma and Lognormal distributions, but it is slower for Pareto because $\alpha = 2.202$ generates a very fat tail that slows down extremely the convergence of the Central Limit Theorem. This is a well-known fact [20] since the third moment diverges and the region where the Gaussian approximation holds grows extremely slowly with the number of steps of the walk. This effect is even more evident in the rightmost plot where all the three distributions have very fat tails. Overall, Fig. 4 confirms that ε does not need to be "too much larger" than σ/μ to get convergence to the predicted mean exit time, as stated in Theorem 1.

The fourth experiment considered streams of increasing length n (up to 10^6) that follow the gap distributions of the first column of Fig. 3. For each part of a stream, we computed with the MET algorithm the s segments that approximate that stream with error $\varepsilon = 50$. By repeating the experiment 10^4 times, we computed the average and the standard deviation of s/n. Fig. 5 shows that for a large n the distribution of s/n concentrates on $\lambda = (\sigma/(\mu\,\varepsilon))^2$, with a speed that is faster for smaller $\mu\varepsilon/\sigma$, as predicted by Theorem 4.

The fifth experiment, reported in Fig. 6, shows the average segment length of OPT on real-world datasets of 200 million elements from [11]. The books dataset represents book sale popularity from Amazon, while fb contains Facebook user IDs. Even though these datasets do not satisfy the assumption of Theorem 1, the fitted curves show a superlinear growth in ε. This suggests that the $\varepsilon^{1+o(1)}$ growth established in our analysis may also be valid on datasets that do not strictly follow the assumption on iid gaps.

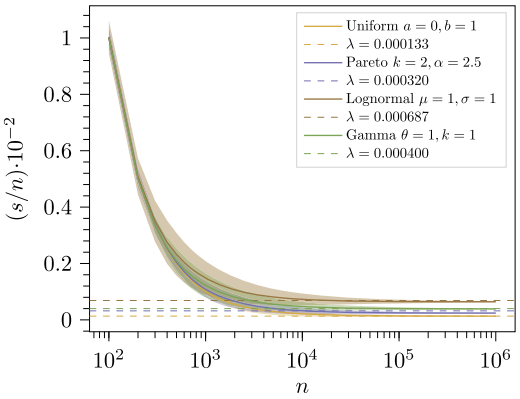

Fig. 5. The solid line is the average and the shaded region is the standard deviation of s/n over 10^4 streams for four distributions, where s is the number of segments computed by MET for a stream of length n. The dashed line depicts the limit stated in Theorem 4 to which the experimental values clearly converge to (quickly, at moderately small values of n).

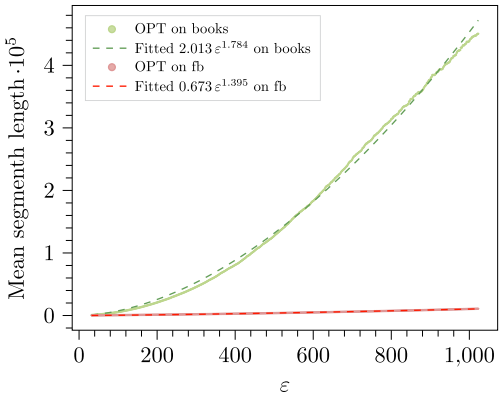

Fig. 6. The average length of a segment computed by OPT on two real datasets exhibit a superlinear growth in ε.

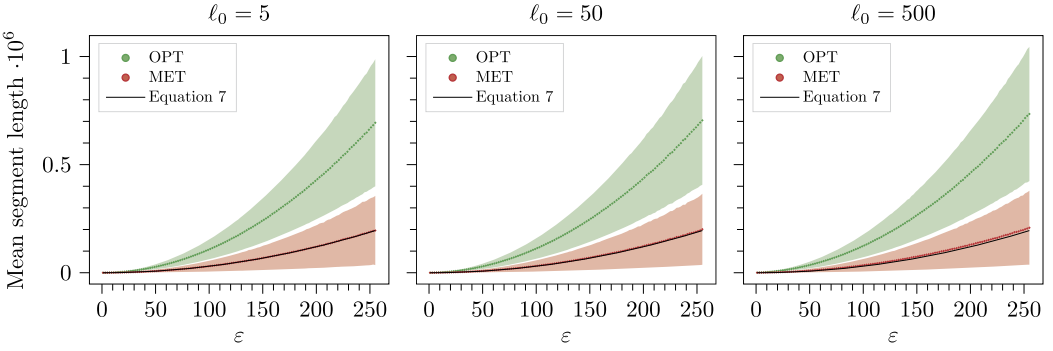

Fig. 7. The mean segment length computed by OPT and MET on keys generated by three moving-average processes of order $\ell_0 = 5, 50$ and 500, respectively. The solid black line overlaps the red dots of MET and thus it shows that Equation (8) provides a good approximation for the case of correlated keys.

The sixth experiment considered the random process described in Example 1 of Section 4, i.e. streams of gaps generated by moving-average processes of order ℓ_0. Specifically, we computed the moving average of reals drawn from a uniform distribution (with parameters $a = 0$, $b = 1$) by using unit weights ϕ_i and by varying ℓ_0 in $\{5, 50, 500\}$. For each value of ℓ_0, we repeated the experiment 10^7 times, each time picking an integer ε in the range $[1, 2^8]$ and running OPT and MET with argument ε. Fig. 7 shows that the mean segment length of the two algorithms scales quadratically in ε and that the conjectured correction of the prefactor related to autocorrelation is in a very good agreement with simulations. This entails that even in the case of keys correlated at large lags (e.g. $\ell_0 = 500$) the result of Theorem 1 might still hold, as discussed thoroughly in Section 4.

The seventh and final experiment considered the random process described in Example 2 of Section 4, i.e. streams of gaps generated by autoregressive processes with parameter φ. We sampled the white noise terms from a uniform distribution

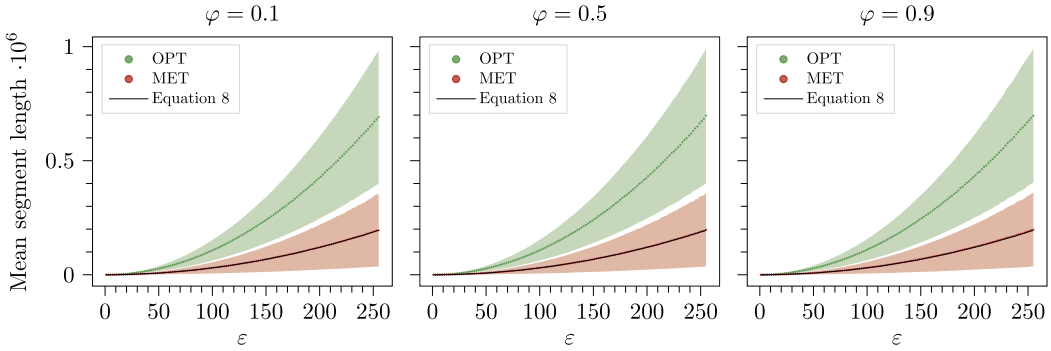

Fig. 8. The mean segment length computed by OPT and MET on keys generated by three autoregressive processes with parameter $\varphi = 0.1, 0.5$ and 0.9, respectively. The solid black line overlaps the red dots of MET and thus it shows that Equation (9) provides a good approximation for the case of correlated keys.

(with parameters $a = 0$, $b = 1$) and varied φ in $\{0.1, 0.5, 0.9\}$. For each value of φ, we repeated the experiment 10^7 times, each time picking an integer ε in the range $[1, 2^8]$ and running OPT and MET with argument ε. Fig. 8 shows that the mean segment length of the two algorithms scales quadratically in ε and that the conjectured correction of the prefactor related to autocorrelation is in very good agreement with simulations.

7. Conclusions

In this paper, we have provided the first theoretical analysis of learned indexes, thus offering mathematical grounds to their known excellent practical performance in terms of space occupancy. Our theoretical results have been corroborated in precision and robustness by a large set of experiments. Our paper leaves open a series of interesting theoretical questions, two of them are sketched here.

The first one concerns the main result stated in Theorem 1. It holds under the condition that "ε is sufficiently larger than σ/μ", therefore it is natural to ask whether this condition can be waived, thus making the theorem stronger, and whether/how we can bound the error made by the approximation for finite and not too large values for $\varepsilon\mu/\sigma$.

A second question asks to provide a formal analysis of the distribution of the segment lengths found by the optimal algorithm (OPT) proposed by O'Rourke [13]. We know that they are longer than the ones found by MET and thus their number grows on average as $\Omega((\mu\varepsilon/\sigma)^2)$, but how much are they longer than what it is stated asymptotically in this Ω-bound?

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

Acknowledgements

We thank Erik Demaine for a preliminary and inspiring discussion. We thank the staff of the Green Data Centre at the University of Pisa for providing us with machines and technical support to execute the experiments that have been presented in this paper.

Part of this work has been supported by the Italian MIUR PRIN project "Multicriteria data structures and algorithms: from compressed to learned indexes, and beyond" (Prot. 2017WR7SHH), by Regione Toscana (under POR FSE 2014/2020), by the EU H2020 programmes "SoBigData++: European Integrated Infrastructure for Social Mining and Big Data Analytics" (INFRAIA-01-2018-2019, grant #871042), and by "Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us" (grant #820437).

References

[1] P. Ferragina, F. Lillo, G. Vinciguerra, Why are learned indexes so effective?, in: Proc. 37th International Conference on Machine Learning, vol. 119, ICML, 2020.

[2] P. Ferragina, G. Vinciguerra, Learned data structures, in: L. Oneto, N. Navarin, A. Sperduti, D. Anguita (Eds.), Recent Trends in Learning From Data, Springer, 2020, pp. 5–41.

[3] N. Ao, F. Zhang, D. Wu, D.S. Stones, G. Wang, X. Liu, J. Liu, S. Lin, Efficient parallel lists intersection and index compression algorithms using graphics processing units, Proc. VLDB Endow. 4 (2011) 470–481.

[4] T. Kraska, A. Beutel, E.H. Chi, J. Dean, N. Polyzotis, The case for learned index structures, in: Proc. International Conference on Management of Data, 2018, pp. 489–504.

[5] A. Galakatos, M. Markovitch, C. Binnig, R. Fonseca, T. Kraska, FITing-Tree: a data-aware index structure, in: Proc. International Conference on Management of Data, 2019, pp. 1189–1206.
[6] J. Ding, U.F. Minhas, J. Yu, C. Wang, J. Do, Y. Li, H. Zhang, B. Chandramouli, J. Gehrke, D. Kossmann, D. Lomet, T. Kraska, ALEX: an updatable adaptive learned index, in: Proc. International Conference on Management of Data, 2020, pp. 969–984.
[7] P. Ferragina, G. Vinciguerra, The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds, Proc. VLDB Endow. 13 (2020) 1162–1175, https://pgm.di.unipi.it.
[8] A. Boffa, P. Ferragina, G. Vinciguerra, A "learned" approach to quicken and compress rank/select dictionaries, in: Proc. SIAM Symposium on Algorithm Engineering and Experiments, ALENEX, 2021.
[9] J.S. Vitter, External memory algorithms and data structures: dealing with massive data, ACM Comput. Surv. 33 (2001) 209–271.
[10] G. Navarro, J. Rojas-Ledesma, Predecessor search, ACM Comput. Surv. 53 (2020).
[11] A. Kipf, R. Marcus, A. van Renen, M. Stoian, A. Kemper, T. Kraska, T. Neumann, SOSD: a benchmark for learned indexes, in: Workshop on ML for Systems at NeurIPS, 2019.
[12] A. Kipf, R. Marcus, A. van Renen, M. Stoian, A. Kemper, T. Kraska, T. Neumann, RadixSpline: a single-pass learned index, in: Proc. International Workshop on Exploiting Artificial Intelligence Techniques for Data Management at SIGMOD, 2020.
[13] J. O'Rourke, An on-line algorithm for fitting straight lines between data ranges, Commun. ACM 24 (1981) 574–578.
[14] R. Motwani, P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
[15] J. Masoliver, M. Montero, J. Perelló, Extreme times in financial markets, Phys. Rev. E 71 (2005) 056130.
[16] S. Redner, A Guide to First-Passage Processes, Cambridge University Press, 2001.
[17] C.W. Gardiner, Handbook of Stochastic Methods for Physics, Chemistry and the Natural Sciences, 2nd ed., Springer-Verlag, 1985.
[18] P. Billingsley, Convergence of Probability Measures, 2nd ed., Wiley, 1999.
[19] P. Embrechts, T. Mikosch, C. Klüppelberg, Modelling Extremal Events: for Insurance and Finance, Springer-Verlag, 1997.
[20] W. Feller, An Introduction to Probability Theory and its Applications, vol. 2, John Wiley & Sons, 1970.
[21] E.J. Keogh, S. Chu, D.M. Hart, M.J. Pazzani, An online algorithm for segmenting time series, in: Proc. IEEE International Conference on Data Mining, 2001, pp. 289–296.
[22] H. Elmeleegy, A.K. Elmagarmid, E. Cecchet, W.G. Aref, W. Zwaenepoel, Online piece-wise linear approximation of numerical streams with precision guarantees, Proc. VLDB Endow. 2 (2009) 145–156.
[23] D.Z. Chen, H. Wang, Approximating points by a piecewise linear function, Algorithmica 66 (2013) 682–713.
[24] Q. Xie, C. Pang, X. Zhou, X. Zhang, K. Deng, Maximum error-bounded piecewise linear representation for online stream approximation, VLDB J. 23 (2014) 915–937.