Why Are Learned Indexes So Effective?

Paolo Ferragina [*1] **Fabrizio Lillo** [*2] **Giorgio Vinciguerra** [*1]

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 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 open problem. We obtain this result by discovering and exploiting a link between the original problem and a mean exit time problem over a proper stochastic process which, we show, is related to the space and time occupancy of those learned indexes. Our general result is then specialised to five well-known distributions: Uniform, Lognormal, Pareto, Exponential, and Gamma; and it is corroborated in precision and robustness by a large set of experiments.

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 (Ferragina & Vinciguerra, 2020a).

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:

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

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 (Ao et al., 2011; Kraska et al., 2018; Galakatos et al., 2019; Ding et al., 2020; Ferragina & Vinciguerra, 2020b) and others surveyed in (Ferragina & Vinciguerra, 2020a). 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 $\mathit{rank}(x)$ 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, $\mathit{member}(x)$ can be implemented by checking whether $A[\mathit{rank}(x)] = x$; $\mathit{predecessor}(x)$ consists of returning $A[\mathit{rank}(x) - 1]$; and $\mathit{range}(x, y)$ consists of scanning the array A from position $\mathit{rank}(x)$ 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 $\{(x, \mathit{rank}(x))\}_{x \in S}$ in which we look for a model $f\colon \mathcal{U} \to \{0, \dots, n-1\}$ mapping keys to their position in A that minimises the error $|f(x) - \mathit{rank}(x)|$ 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 $\mathit{rank}(x)$, and thus answer correctly the query on x. As an example, we can use a binary search in A in a neighbourhood of size $\mathit{err} = \max_{x \in \mathcal{U}} |f(x) - \mathit{rank}(x)|$ around the approximate position $f(x)$. An illustrative example is given in Figure 1.

We observe that this has been a significant breakthrough in the design of indexes, because the resulting learned data structure answers queries in $O(\log \mathit{err})$ time plus the cost

[*] Equal contribution [1] Department of Computer Science, University of Pisa, Italy [2] Department of Mathematics, University of Bologna, Italy. Correspondence to: Giorgio Vinciguerra <giorgio.vinciguerra@phd.unipi.it>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

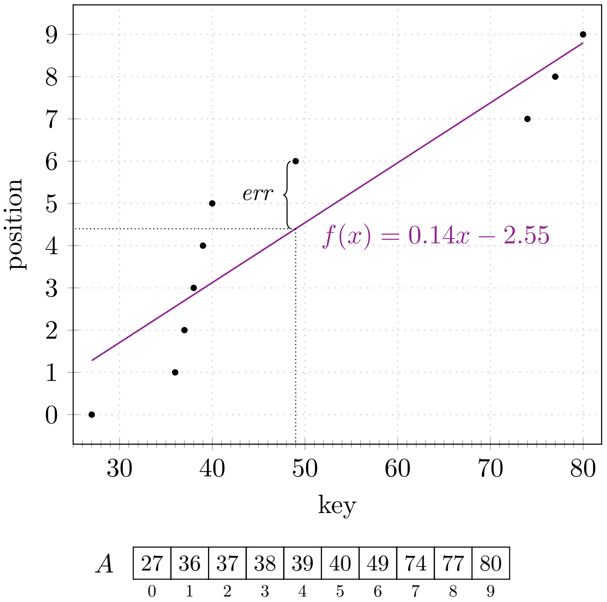

Figure 1. A set S of ten keys stored in a sorted array A and the corresponding set of points $D = \{(x, rank(x))\}_{x \in S}$ in the Cartesian plane. The linear model f, computed using ordinary least squares on D, estimates that $x = 49$ is in position $r = \lfloor f(x) \rfloor = 4$, but the true rank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

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 (Vitter, 2001), are better or worse than a learned index.

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. (2011) used simple least-squares linear regression. Kraska et al. (2018) 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. (2019) orchestrated the segments composing the PLA with a classic B$^+$-tree, while Ferragina & Vinciguerra (2020b) introduced sophisticated and 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 (Kraska et al., 2018; Galakatos et al., 2019; Kipf et al., 2019; Ding et al., 2020; Kipf et al., 2020; Ferragina & Vinciguerra, 2020b). However, it is not yet known whether these learned indexes are *provably better* than classic data structures. 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 Ferragina & Vinciguerra, 2020b). This shows that a learned index is never worse than a B-tree with fan-out B (just take $\varepsilon = \Theta(B)$), but it does not theoretically ensure that it is provably more succinct than it.

As a consequence, learned indexes can be fully recognised as more efficient substitutes of classic data structures only if research corroborates with solid mathematical grounds also their excellent performance in space, currently experimented only on some specific datasets. This amounts to explain from a theoretical perspective their "several orders of magnitude smaller" space occupancy, 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 input S, taken in sorted order, 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 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 & Vinciguerra (2020b) (and due to O'Rourke, 1981) 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 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 variance, the average segment length scales at least *quadratically* with ε which, in turns, 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 almost surely (Theorem 4). Additionally, we specialise Theorem 1 to five well-known distributions (Corollary 1). Finally, we perform a thorough set of experiments corroborating that 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 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 & Vinciguerra (2020b) answers queries as fast as a B$^+$-tree while improving its space to $O(n/B^2)$ (see Corollary 2).

In the concluding section, we will comment on some challenging issues that our analysis raises and that deserve further study and experimentation.

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:

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

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 Figure 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.

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 for the 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 $\alpha > 2$.*

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

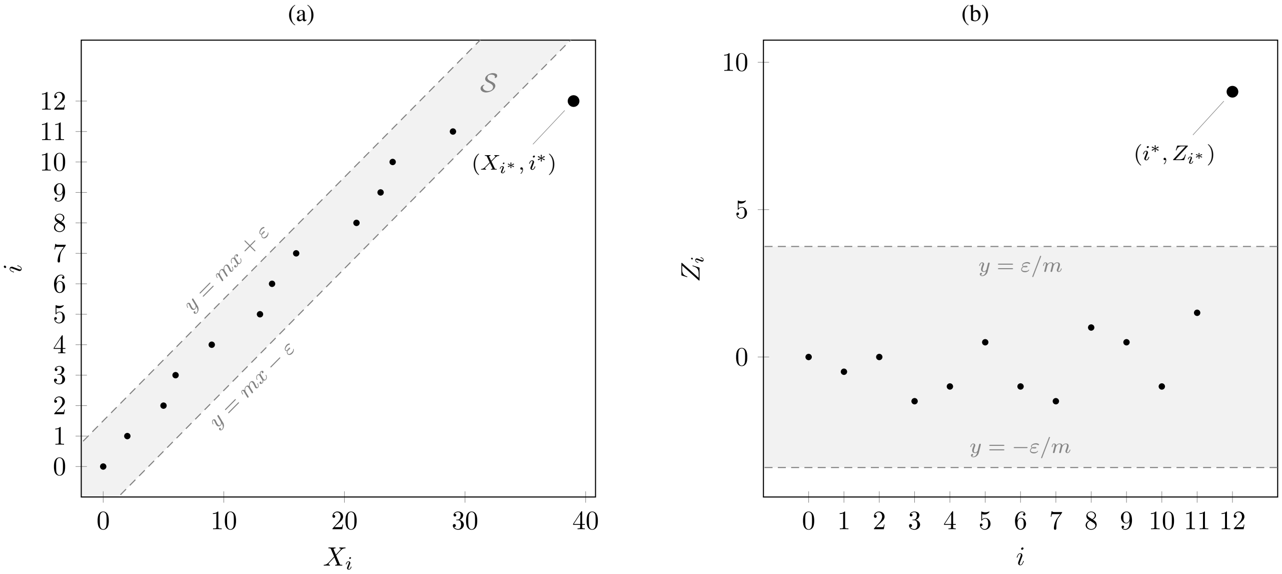

Figure 2. An example of random walk (a) and the corresponding transformed random walk (b).

- $\varepsilon^2/(e^{\sigma^2} - 1)$ *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 $\theta > 0$.*

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” (Motwani & Raghavan, 1995).

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. The proofs of our theorems rely on Central Limit Theorem (CLT), whose domain of validity includes also dependent random variables. For example, if the time series of gaps is weakly stationary, CLT holds (Hamilton, 1994) and our theorems can be extended accordingly. More generally, one can expand even further the class of time series where CLT holds by using the concept of strong mixing (or α-mixing, see Billingsley, 1995) which, broadly speaking, means that gaps temporally distant from one another are nearly independent.

In summary, when autocorrelation of gaps are not extreme, we expect our theorems to continue to hold. Further details and a version of the theorems with extended hypotheses will be given in the journal version of the paper.

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

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

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 $\text{Var}[W_j] = \text{Var}[G_j] = \sigma^2$ and pdf $f_W(w) = f_G(w + 1/m)$.

An example of this transformed random walk is depicted in Figure 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 (Masoliver et al., 2005; Redner, 2001) 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

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

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 (Gardiner, 1985).[2]

Now, it is well known (Gardiner, 1985) that for a driftless Wiener process the MET out of an interval $[-\delta/2, \delta/2]$ is

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

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, we notice that, 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

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

Clearly, by taking the limit $d \to 0$ (i.e. $\mu \to 1/m$), 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 Theorem 3

Following Gardiner (1985, 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

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

3.4. 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 $\text{Var}[i_j^*] = \varsigma^2$. It is well known

[2] A mathematical more precise but equivalent statement can be done using the Donsker's theorem (Billingsley, 1999).

(see Embrechts et al., 1997, §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. 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. O'Rourke, 1981; Keogh et al., 2001; Elmeleegy et al., 2009; Chen & Wang, 2013; Xie et al., 2014, and refs therein). Theorem 4 can eventually give an estimate of the number of segments computed by those 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 (1981), 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 & Vinciguerra (2020b) 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 2. *Let S and n be as in Theorem 1. There exists a data structure on S, the PGM-index, that 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 (1981), 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 Ferragina & Vinciguerra (2020b) and setting $\varepsilon = \Theta(B)$ the claim follows. □

5. 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 most effective in practice for the learned index of Ferragina & Vinciguerra (2020b). Then, we ran the following PLA-algorithms with arguments ε and S:

MET. 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 size 2ε, see Equation 1. This corresponds to the random process we used to prove Theorem 1.

OPT. The algorithm that constructs the optimal PLA-model (O'Rourke, 1981) used in the PGM-index of Ferragina & Vinciguerra (2020b). This algorithm computes the segment (of any slope and intercept) that ε-approximate the longest prefix of S.

Our first experiment analysed the length of the segments computed by each of the previous two 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 Figure 3 (next page) the mean and the standard deviation of these segment lengths. The figure shows that the theoretical mean segment length 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. 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 adapts optimally to each sequence of points given in input. Thus it is more robust to outliers and hence can find longer segments.

Overall this first experiment entails that learned indexes (and, in particular, the learned index based on an optimal use of linear models, see Ferragina & Vinciguerra, 2020b), 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 2).

[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.

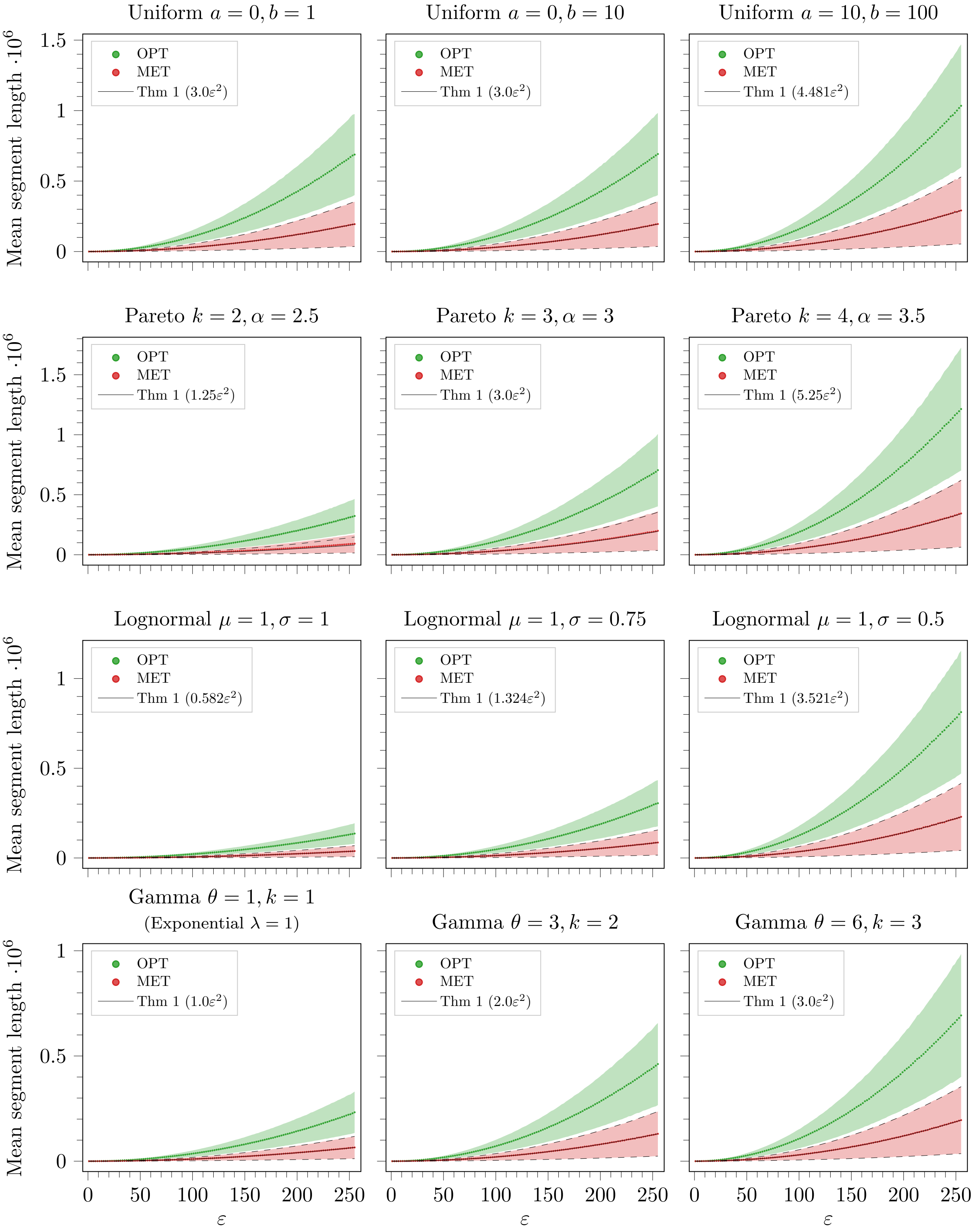

Figure 3. We consider four gap distributions — Uniform, Pareto, Lognormal, and Gamma — with various parameter settings. We plot the formula $(\mu^2/\sigma^2)\,\varepsilon^2$ given in Theorem 1 with a solid black line and the Mean Exit Time (MET) of the experimented random walk with red points. The figure shows that they overlap, thus the formula stated in Theorem 1 is an accurate prediction of 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 and thus it shows that OPT is more robust to outliers.

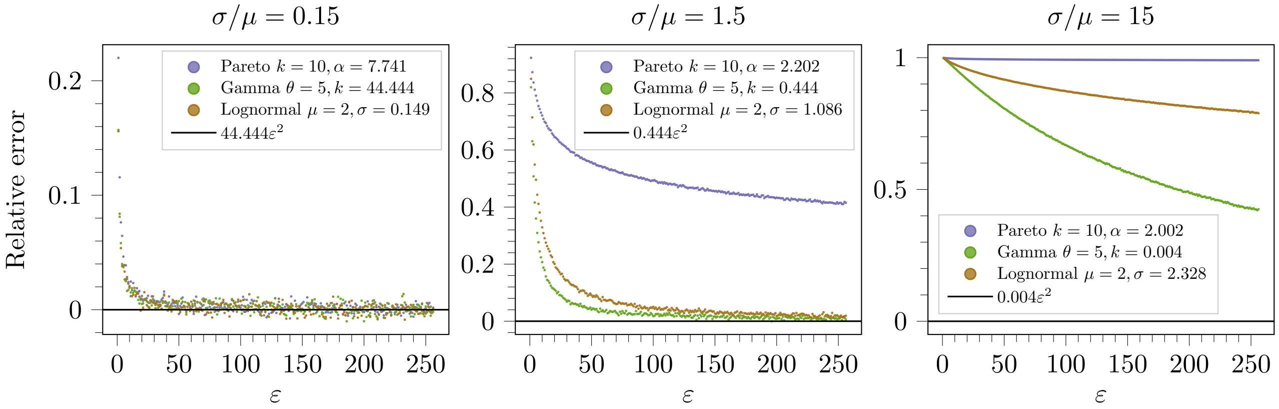

Figure 4. Three plots for three different settings of the ratio σ/μ for the distributions: Pareto, Gamma and Lognormal. We plot the relative error between the formula $(\mu^2/\sigma^2)\,\varepsilon^2$ of Theorem 1 and the experimented MET. Notice how the fat-tail of the distributions affects the accuracy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

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	$[0.056, 0.056]$

Table 1. We show the range of slopes determined by algorithm OPT in the experiments of Figure 3. We notice that those ranges are centred and close to $1/\mu$, which is the theoretical slope that maximises the MET of the random walk depicted in Figure 2a.

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 Figure 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 (see e.g. Feller, 1970) 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, Figure 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.

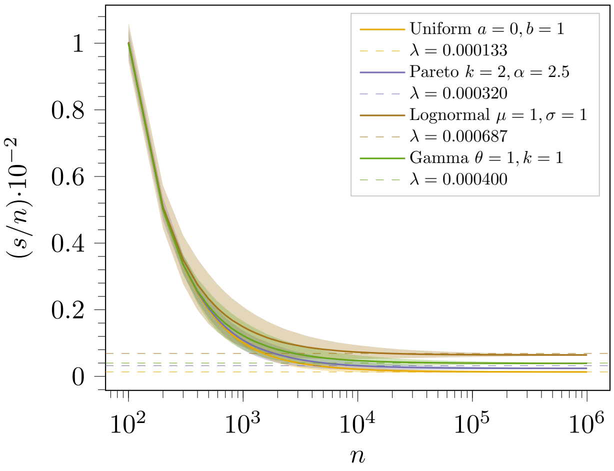

Figure 5. The solid line is the average and the shaded region is the standard deviation of s/n over 10^4 streams for four distributions, where s is the number of segments computed by MET for a stream of length n. The dashed line depicts the limit stated in Theorem 4 to which the experimental values clearly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

The fourth experiment, reported in Figure 5, considered streams of increasing length n (up to 10^6) that follow the gap distributions of the first column of Figure 3. For each part of a stream, we computed with the MET algorithm the

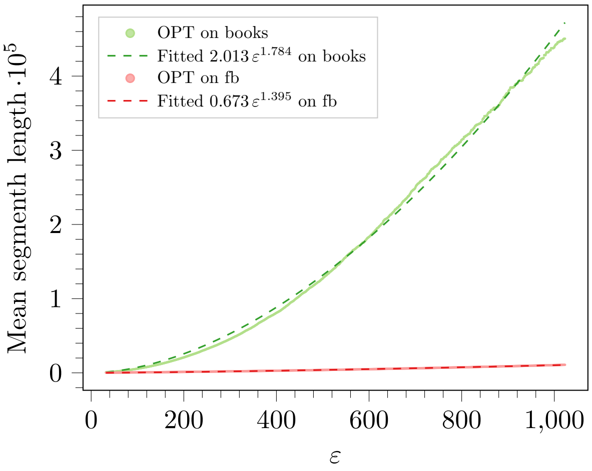

Figure 6. The average length of a segment computed by OPT on two real datasets exhibit a superlinear growth in ε.

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. Figure 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.

Figure 6 shows the results of our final experiment, which measured the average segment length of OPT on real-world datasets of 200 million elements from Kipf et al. (2019). 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. However, as stated at the end of Section 3, future research is needed to shed more light on this issue.

6. Conclusions

In this paper, we have provided the first theoretical analysis of learned indexes, thus offering mathematical grounds to their known excellent practical performance. 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, some 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 with Brownian motion 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 (1981). We know that they are longer than MET and thus grow on average as $\Omega((\mu\varepsilon/\sigma)^2)$, but how much are they longer than what it is stated asymptotically in this Ω-bound?

As a final question, and in the light of the plots in Figure 6, we ask what changes in Theorems 1 and 2 if we relax the iid assumption on the distribution of the gaps. We argue that the bounds should still hold in their superlinear growth, in the same vein as it happens with *weakly* correlated variables in the Central Limit Theorem (see e.g. Feller, 1970).

Acknowledgements. We thank Erik Demaine for a preliminary and inspiring discussion on looking at key gaps when keys are taken in their sorted order. We also thank the anonymous reviewers for their helpful comments and suggestions. 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 European Integrated Infrastructure for Social Mining and Big Data Analytics (SoBigData++, Grant Agreement #871042), and by PRA UniPI 2018 "Emerging Trends in Data Science".

References

Ao, N., Zhang, F., Wu, D., Stones, D. S., Wang, G., Liu, X., Liu, J., and Lin, S. Efficient parallel lists intersection and index compression algorithms using graphics processing units. *PVLDB*, 4(8):470–481, 2011.

Billingsley, P. *Probability and Measure*. Wiley, 3rd edition, 1995.

Billingsley, P. *Convergence of Probability Measures*. Wiley, 2nd edition, 1999.

Chen, D. Z. and Wang, H. Approximating points by a piecewise linear function. *Algorithmica*, 66(3):682–713, 2013.

Ding, J., Minhas, U. F., Yu, J., Wang, C., Do, J., Li, Y., Zhang, H., Chandramouli, B., Gehrke, J., Kossmann, D., Lomet, D., and Kraska, T. ALEX: an updatable adaptive learned index. In *Proc. International Conference on Management of Data*, pp. 969–984, 2020.

Elmeleegy, H., Elmagarmid, A. K., Cecchet, E., Aref, W. G., and Zwaenepoel, W. Online piece-wise linear approximation of numerical streams with precision guarantees. *Proc. VLDB Endowment*, 2(1):145–156, 2009.

Embrechts, P., Mikosch, T., and Klüppelberg, C. *Modelling Extremal Events: For Insurance and Finance*. Springer-Verlag, 1997.

Feller, W. *An introduction to probability theory and its applications*, volume 2. John Wiley & Sons, 1970.

Ferragina, P. and Vinciguerra, G. Learned data structures. In Oneto, L., Navarin, N., Sperduti, A., and Anguita, D. (eds.), *Recent Trends in Learning From Data*, pp. 5–41. Springer, 2020a.

Ferragina, P. and Vinciguerra, G. The PGM-index: a fully-dynamic compressed learned index with provable worst-case bounds. *PVLDB*, 13(8):1162–1175, 2020b. URL `https://pgm.di.unipi.it`.

Galakatos, A., Markovitch, M., Binnig, C., Fonseca, R., and Kraska, T. FITing-Tree: a data-aware index structure. In *Proc. International Conference on Management of Data*, pp. 1189–1206, 2019.

Gardiner, C. W. *Handbook of stochastic methods for physics, chemistry and the natural sciences*. Springer-Verlag, 2nd edition, 1985.

Hamilton, J. D. *Time Series Analysis*. Princeton University Press, 1994.

Keogh, E. J., Chu, S., Hart, D. M., and Pazzani, M. J. An online algorithm for segmenting time series. In *Proc. IEEE International Conference on Data Mining*, pp. 289–296, 2001.

Kipf, A., Marcus, R., van Renen, A., Stoian, M., Kemper, A., Kraska, T., and Neumann, T. SOSD: a benchmark for learned indexes. In *Workshop on ML for Systems at NeurIPS*, 2019.

Kipf, A., Marcus, R., van Renen, A., Stoian, M., Kemper, A., Kraska, T., and Neumann, T. RadixSpline: a single-pass learned index. In *Proc. International Workshop on Exploiting Artificial Intelligence Techniques for Data Management at SIGMOD*, 2020.

Kraska, T., Beutel, A., Chi, E. H., Dean, J., and Polyzotis, N. The case for learned index structures. In *Proc. International Conference on Management of Data*, pp. 489–504, 2018.

Masoliver, J., Montero, M., and Perelló, J. Extreme times in financial markets. *Physical Review E*, 71:056130, 2005.

Motwani, R. and Raghavan, P. *Randomized algorithms*. Cambridge University Press, 1995.

O'Rourke, J. An on-line algorithm for fitting straight lines between data ranges. *Communications of the ACM*, 24 (9):574–578, 1981.

Redner, S. *A guide to first-passage processes*. Cambridge University Press, 2001.

Vitter, J. S. External memory algorithms and data structures: Dealing with massive data. *ACM Computing Surveys*, 33(2):209–271, 2001.

Xie, Q., Pang, C., Zhou, X., Zhang, X., and Deng, K. Maximum error-bounded piecewise linear representation for online stream approximation. *The VLDB Journal*, 23(6): 915–937, 2014.