

Compressive Mining: Fast and Optimal Data Mining in the Compressed Domain

Michail Vlachos · Nikolaos M. Freris · Anastasios Kyrillidis

Received: date / Accepted: date

Abstract Real-world data typically contain repeated and periodic patterns. This suggests that they can be effectively represented and compressed using only a few coefficients of an appropriate basis (e.g., Fourier, Wavelets, etc.). However, distance estimation when the data are represented using different sets of coefficients is still a largely unexplored area. This work studies the optimization problems related to obtaining the *tightest* lower/upper bound on Euclidean distances when each data object is potentially compressed using a different set of orthonormal coefficients. Our technique leads to better distance estimates, which translates into more accurate search, learning and mining operations *directly* in the compressed domain.

We establish the properties of optimal solutions, and leverage the theoretical analysis to develop a fast algorithm to obtain an *exact* solution to the problem. The suggested solution provides the tightest provable estimation of the L_2 -norm or the correlation. We show that typical data-analysis operations, such as k-NN search or k-Means clustering, can operate more accurately using the proposed compression and distance reconstruction technique. We compare it with many other prevalent compression and reconstruction techniques, including random projections and PCA-based techniques. We highlight a surprising result, namely that when the data are highly sparse in some basis, our

technique may even outperform PCA-based compression.

The contributions of this work are generic as our methodology is applicable to any sequential or high-dimensional data as well as to any orthogonal data transformation used for the underlying data compression scheme.

Keywords: Data Compression, Compressive Sensing, Fourier, Wavelets, Water-filling algorithm, Convex Optimization

1 Introduction

Increasing data sizes are a perennial problem for data analysis. This dictates the need not only for more efficient data-compression schemes, but also for analytic operations that work directly in the compressed domain. Compression schemes exploit inherent patterns and structures in the data. In fact, many natural and industrial processes exhibit patterns and periodicities. Periodic behavior is omnipresent; be it in environmental and natural processes [1, 2], in medical and physiological measurements (e.g., ECG data [3]), weblog data [4, 5], or network measurements [6]. The aforementioned are only a few of the numerous scientific and industrial fields that exhibit repetitions. Examples from some of these areas are shown in Fig. 1.

When data contain an inherent structure, more efficient compression can be performed with minimal loss in data quality (see Fig. 3 for an example). The bulk of related work on compression and distance estimation used the same sets of coefficients for all objects [7–10]. This simplified the distance estimation in the compressed domain. However, by encoding the data

M. Vlachos and A. Kyrillidis
IBM-Research Zürich, Säumerstrasse 4, CH-8803, Rüschlikon,
Switzerland
E-mail: {mvl,nas}@zurich.ibm.com

N. Freris
School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne (EPFL), CH-1015 Lausanne,
Switzerland
E-mail: nikolaos.freris@epfl.ch

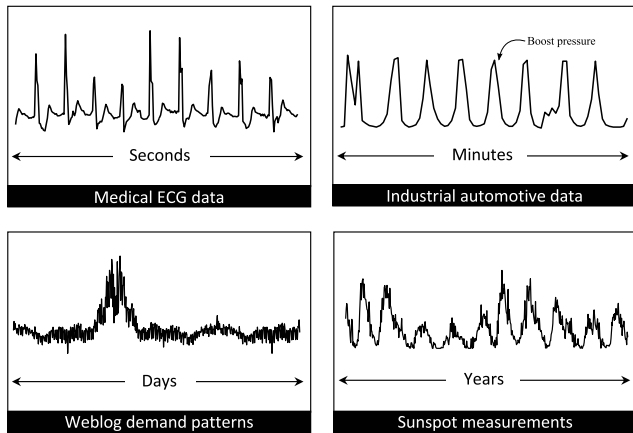


Fig. 1: Many scientific fields entail periodic data. Examples from medical, industrial, web and astronomical measurements.

using only a few and potentially disjoint sets of high-energy coefficients in an orthonormal basis, one can achieve better reconstruction performance. Nonetheless, it was not known how to compute tight distance estimates using such a representation. Our work exactly addresses this issue: given data that are compressed using disjoint coefficient sets of an orthonormal basis (for reasons of higher fidelity), *how can distances among the compressed objects be estimated with the highest fidelity?*

Here, we provide the tightest possible upper and lower bounds on the original distances, based only on the compressed objects. By *tightest*, we mean that, given the information available, no better estimate can be derived. Distance estimation is fundamental for data mining: the majority of mining and learning tasks are distance-based, including clustering (e.g. k-Means or hierarchical), k-NN classification, outlier detection, pattern matching, etc. This work focuses on the case where the distance is the widely used Euclidean distance (L_2 -norm), but makes no assertions on the underlying transform used to compress the data: As long as the transform is orthonormal, our methodology is applicable. In the experimental section, we use both Fourier and Wavelets Decomposition as a data compression technique. Our main **contributions** are summarized below:

- We formulate the problem of tight distance estimation in the compressed domain as two optimization problems for obtaining lower/upper bounds. We show that both problems can be solved simultaneously by solving a single convex optimization program.

- We derive the necessary and sufficient Karush-Kuhn-Tucker (KKT) conditions and study the properties of optimal solutions. We use the analysis to devise

exact closed-form solution algorithms for the optimal distance bounds.

- We evaluate our analytical findings experimentally; we compare the proposed algorithms with prevalent distance estimation schemes, and demonstrate significant improvements in terms of estimation accuracy. We further compare the performance of our optimal algorithm with that of a numerical scheme based on convex optimization, and show that our scheme is at least two orders of magnitude faster, while also providing more accurate results.

- We also provide extensive evaluations with mining tasks in the compressed domain using our approach and many other prevalent compression and distance reconstruction schemes used in the literature (random projections, SVD, etc).

2 Related Work

We briefly position our work in the context of other similar approaches in the area. The majority of data-compression techniques for sequential data use the *same* set of low-energy coefficients whether using Fourier [7,8], Wavelets [9,10] or Chebyshev polynomials [13] as the orthogonal basis for representation and compression. Using the same set of orthogonal coefficients has several advantages: a) it is straightforward to compare the respective coefficients; b) space partitioning and indexing structures (such as R-trees) can be directly used on the compressed data; c) there is no need to store also the indices (position) of the basis functions to which the stored coefficients correspond. The disadvantage is that both object reconstruction and distance estimation may be far from optimal for a given fixed compression ratio.

One can also record side information, such as the energy of the discarded coefficients, to better approximate the distance between compressed sequences by leveraging the Cauchy-Schwartz inequality [14]. This is shown in Figure 2a). In [12,11], the authors advocated the use of high-energy coefficients and side information on the discarded coefficients for weblog sequence repositories; in that setting one of the sequences was compressed, whereas the query was uncompressed, i.e., all coefficients were available as illustrated in Figure 2b). This work examines the most general and challenging case when both objects are compressed. In such case, we record a (generally) different set of high-energy coefficients and also store aggregate side information, such as the energy of the omitted data; this is depicted in Figure 2c). We are not aware of any previous art addressing this problem to derive

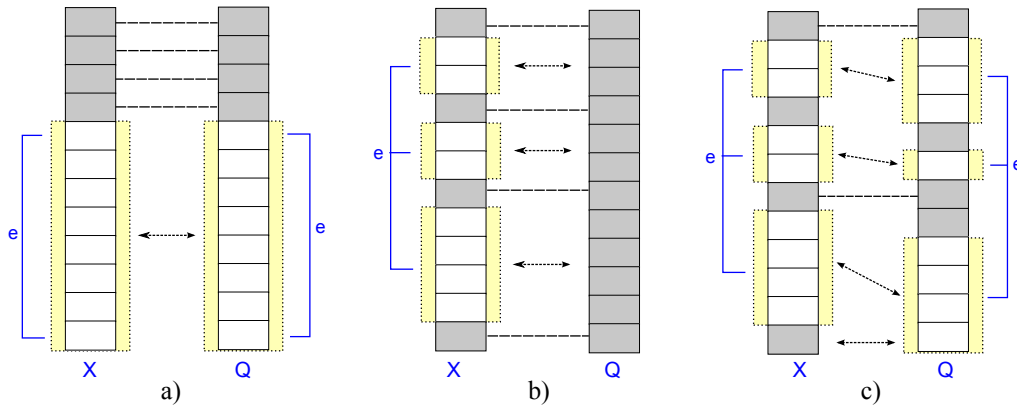


Fig. 2: Comparison with previous work. Distance estimation between a compressed sequence (X) and a query (Q) represented in *any* complete orthonormal basis. A compressed sequence is represented by a set of stored coefficients (gray) as well as the error e incurred because of compression (yellow). a) Both X, Q are compressed by storing the first coefficients. b) The highest-energy coefficients are used for X , whereas Q is uncompressed as in [11, 12]. c) The problem we address: both sequences are compressed using the highest-energy coefficients; note that in general for each object a different set of coefficients is used.

either optimal or suboptimal bounds on distance estimation.

The above approaches consider determining distance estimation in the compressed domain. There is also a big body of work that considers probabilistic distance estimation via low-dimensional embeddings. Several projection techniques for dimensionality reduction can preserve the geometry of the points [15, 16]. These results heavily depend on the work of Johnson and Lindenstrauss [17], according to which any set of points can be projected onto a logarithmic (in the cardinality of the data points) dimensional subspace, while still retaining the relative distances between the points, thus preserving an approximation of their nearest neighbors [18, 19] or clustering [20, 21]. Both random [22] and deterministic [23] constructions have been proposed in the literature.

This paper extends and expands the work of [24]. Here we include additional experiments that show the performance of our methodology for k -NN-search, and k -Means clustering directly in the compressed domain. We also compare our approach with the performance of Principal Components and Random Projection techniques (both in the traditional and in the compressive sensing setting). Finally, we also conduct experiments using other orthonormal bases (namely, wavelets) to demonstrate the generality of our technique. In the experimental section of this work, we compare our methodology to both deterministic and probabilistic techniques.

3 Searching Data Using Distance Estimates

We consider a database \mathcal{DB} that stores sequences as N -dimensional complex vectors $\mathbf{x}^{(i)} \in \mathbb{C}^N, i = 1, \dots, V$. A search problem that we examine is abstracted as follows: a user is interested in finding the k most ‘similar’ sequences to a given query sequence $\mathbf{q} \in \mathcal{DB}$, under a certain distance metric $d(\cdot, \cdot) : \mathbb{C}^{N \times N} \rightarrow \mathbb{R}_+$. This is an elementary, yet fundamental operation known as k -Nearest-Neighbor (k -NN)-search. It is a core function in database-querying, data-mining and machine-learning algorithms including classification (NN classifier), clustering, etc.

In this paper, we focus on the case where $d(\cdot, \cdot)$ is the Euclidean distance. We note that other measures, e.g., time-invariant matching, can be formulated as Euclidean distance on the periodogram [25]. Correlation can also be expressed as an instance of Euclidean distance on properly normalized sequences [26]. Therefore, our approach is applicable on a wide range of distance measures with little or no modification. However, for ease of exposition, we focus on the Euclidean distance as the most used measure in the literature [27].

Search operations can be quite costly, especially for cases when the dimensionality N of data is high: sequences need to be retrieved from the disk for comparison against the query \mathbf{q} . An effective way to mitigate this is to retain a compressed representation of the sequences to be used as an initial pre-filtering step. The set of compressed sequences could be small enough to keep in-memory, hence enabling a significant performance speedup. In essence, this is a *mul-*

tilevel filtering mechanism. With only the compressed sequences available, we obviously cannot infer the exact distance between the query \mathbf{q} and a sequence $\mathbf{x}^{(i)}$ in the database. However, it is still plausible to obtain *lower* and *upper bounds* of the distance. Using these bounds, one might request a superset of the k -NN answers, which will be then verified using the uncompressed sequences that will need to be fetched and compared with the query, so that the exact distances can be computed. Such filtering ideas are used in the majority of the data-mining literature for speeding up search operations [7, 8, 28].

4 Notation

Consider an N -dimensional sequence $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_N]^T \in \mathbb{R}^N$. For compression purposes, \mathbf{x} is first transformed using a sparsity-inducing (i.e., compressible) basis $\mathcal{F}(\cdot)$ in \mathbb{R}^N or \mathbb{C}^N , such that $\mathbf{X} = \mathcal{F}(\mathbf{x})$. We denote the *forward* linear mapping $\mathbf{x} \rightarrow \mathbf{X}$ by \mathcal{F} , whereas the inverse linear map $\mathbf{X} \rightarrow \mathbf{x}$ is denoted by \mathcal{F}^{-1} , i.e., we say $\mathbf{X} = \mathcal{F}(\mathbf{x})$ and $\mathbf{x} = \mathcal{F}^{-1}(\mathbf{X})$. A nonexhaustive list of invertible linear transformations includes Discrete Fourier Transform (DFT), Discrete Cosine Transform, Discrete Wavelet Transform, etc.

As a running example for this paper, we assume that a sequence is compressed using DFT. In this case, the basis represent sinusoids of different frequencies, and the pair (\mathbf{x}, \mathbf{X}) satisfies

$$X_l = \frac{1}{\sqrt{N}} \sum_{k=1}^N x_k e^{i2\pi(k-1)(l-1)/N}, \quad l = 1, \dots, N$$

$$x_k = \frac{1}{\sqrt{N}} \sum_{l=1}^N X_l e^{i2\pi(k-1)(l-1)/N}, \quad k = 1, \dots, N$$

where i is the imaginary unit $i^2 = -1$.

Given the above, we assume the L_2 -norm as the distance between two sequences \mathbf{x}, \mathbf{q} , which can easily be translated into distance in the frequency domain because of Parseval's theorem [29]:

$$d(\mathbf{x}, \mathbf{q}) := \|\mathbf{x} - \mathbf{q}\|_2 = \|\mathbf{X} - \mathbf{Q}\|_2$$

In the experimental section, we also show applications of our methodology when wavelets are used as the signal decomposition transform.

5 Motivation

The choice of which coefficients to use has a direct impact on the data approximation quality. Although it

has long been recognized that sequence approximation is indeed superior when using high-energy (i.e., best) coefficients [30, 12] - see also Figure 3 for an illustrative example - a barrier still has to be overcome: the efficiency of solution for distance estimation.

Consider a sequence represented using its high-energy coefficients. Then, the compressed sequence will be described by a set of C_x coefficients that hold the largest energy. We denote the vector describing the positions of those coefficients in \mathbf{X} as p_x^+ , and the positions of the remaining ones as p_x^- (that is, $p_x^+ \cup p_x^- = \{1, \dots, N\}$). For any sequence \mathbf{X} , we store the vector $\mathbf{X}(p_x^+)$ in the database, which we denote simply by $\mathbf{X}^+ := \{X_i\}_{i \in p_x^+}$. We denote the vector of discarded coefficients by $\mathbf{X}^- := \{X_i\}_{i \in p_x^-}$. In addition to the best coefficients of a sequence, we can also record one additional value for the energy of the compression error, $e_x = \|\mathbf{X}^-\|_2^2$, i.e., the sum of squared magnitudes of the omitted coefficients.

Then, one needs to solve the following minimization (maximization) problem for calculating the lower (upper) bounds on the distance between two sequences based on their compressed versions:

$$\begin{aligned} \min(\max)_{\mathbf{X}^- \in \mathbb{C}^{|p_x^-|}, \mathbf{Q}^- \in \mathbb{C}^{|p_q^-|}} \quad & \|\mathbf{X} - \mathbf{Q}\|_2 \\ \text{s.t.} \quad & |X_l^-| \leq \min_{j \in p_x^+} |X_j|, \quad \forall l \in p_x^- \\ & |Q_l^-| \leq \min_{j \in p_q^+} |Q_j|, \quad \forall l \in p_q^- \\ & \sum_{l \in p_x^-} |X_l^-|^2 = e_x, \quad \sum_{l \in p_q^-} |Q_l^-|^2 = e_q \end{aligned} \quad (1)$$

The inequality constraints are due to the fact that we use the high-energy components for the compression. Hence, any of the omitted components must have an energy lower than the minimum energy of any kept component.

The optimization problem presented is a complex-valued program: we show a single real-valued *convex program* that is *equivalent* to both the minimization and maximization problems. This program can be solved efficiently with numerical methods [31], cf. Sec. 8.1. However, as we show in the experimental section, evaluating an instance of this problem is not efficient in practice, even for a single pair of sequences. Therefore, although a solution can be found *numerically*, it is generally costly and not suitable for large mining tasks, where one would like to evaluate thousands or millions of lower/upper bounds on compressed sequences.

In this paper, we show how solve this problem **analytically** by exploiting the derived optimality conditions. In this manner we can solve the problem in

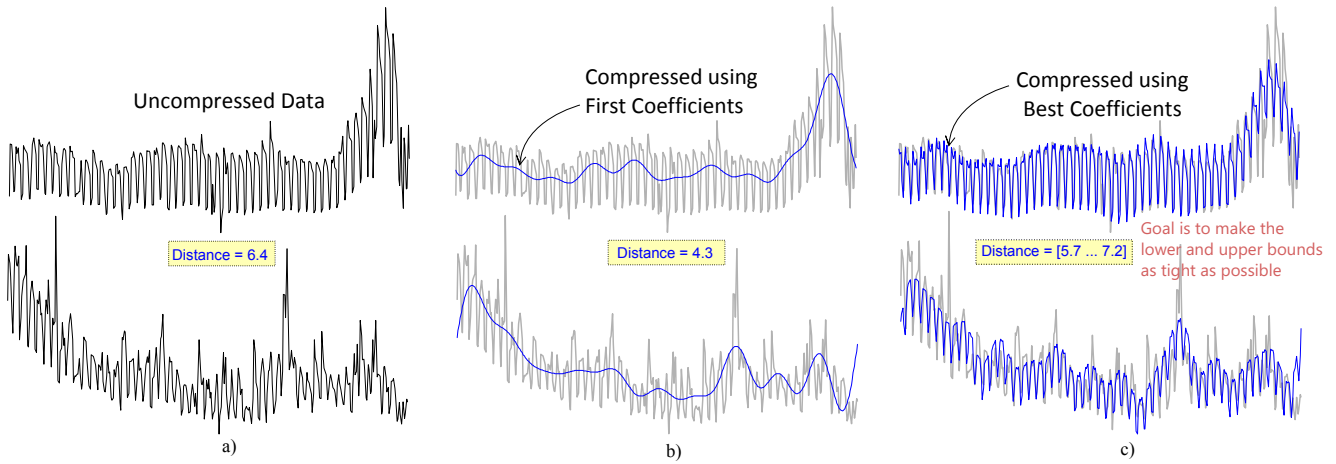


Fig. 3: Motivation for using the high-energy (best) coefficients for compression. Using the best 10 coefficients (c) results in significantly better sequence approximation than when using the first coefficients (b).

a fraction of the time required by numerical methods. We solve this problem as a ‘double-waterfilling’ instance. Vlachos et al. have shown how the optimal lower and upper distance bounds between a compressed and an uncompressed sequence can be relegated to a single waterfilling problem [12]. We revisit this approach as it will be used as a building block for our solution. In addition, we later derive optimality properties for our solution.

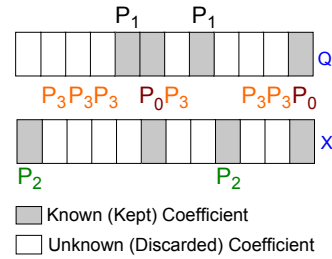


Fig. 4: Visual illustration of sets P_0, P_1, P_2, P_3 between two compressed objects.

6 An Equivalent Convex Optimization Problem

For ease of notation, we consider the partition $\mathcal{P} = \{P_0, P_1, P_2, P_3\}$ of $\{1, \dots, N\}$ (see Fig. 4), where we set the following:

- $P_0 = p_x^+ \cap p_q^+$ are the common known components in two compressed sequences \mathbf{X}, \mathbf{Q} .
- $P_1 = p_x^- \cap p_q^+$ are the components unknown for \mathbf{X} but known for \mathbf{Q} .
- $P_2 = p_x^+ \cap p_q^-$ are the components known for \mathbf{X} but unknown for \mathbf{Q} .
- $P_3 = p_x^- \cap p_q^-$ are the components unknown for both sequences.

Using the standard notation \mathbf{x}^* for the *conjugate transpose* of a complex vector \mathbf{x} , $\Re\{\cdot\}$ to denote the real part of a complex number, and considering all vectors as column vectors, we have that the squared Euclidean

distance is given by:

$$\begin{aligned}
 \|\mathbf{x} - \mathbf{q}\|_2^2 &= \|\mathbf{X} - \mathbf{Q}\|_2^2 = (\mathbf{X} - \mathbf{Q})^* (\mathbf{X} - \mathbf{Q}) \\
 &= \|\mathbf{X}\|_2^2 + \|\mathbf{Q}\|_2^2 - 2\mathbf{X}^* \mathbf{Q} \\
 &= \|\mathbf{X}\|_2^2 + \|\mathbf{Q}\|_2^2 - 4 \sum_{i=1}^N \Re\{X_i Q_i\} \\
 &= \|\mathbf{X}\|_2^2 + \|\mathbf{Q}\|_2^2 - 4 \left(\sum_{l \in P_0} \Re\{X_l Q_l\} \right. \\
 &\quad \left. + \sum_{l \in P_1} \Re\{X_l Q_l\} + \sum_{l \in P_2} \Re\{X_l Q_l\} \right. \\
 &\quad \left. + \sum_{l \in P_3} \Re\{X_l Q_l\} \right).
 \end{aligned}$$

Note that $\|\mathbf{X}\|_2, \|\mathbf{Q}\|_2$ can be inferred by summing the squared magnitudes of the known coefficients with the energy of the compression error. Also, the term $\sum_{l \in P_0} \Re\{X_l Q_l\}$ is known, whereas the last three sums are unknown. Considering the polar form, i.e., absolute value $|\cdot|$ and argument $\arg(\cdot)$

$$X_l = |X_l| e^{j \arg(X_l)}, \quad Q_l = |Q_l| e^{j \arg(Q_l)},$$

we have that the decision variables are vectors $|X_l|, \arg(X_l), l \in p_x^-$ as well as $|Q_l|, \arg(Q_l), l \in p_q^-$. Observe that for $x, y \in \mathbb{C}$ with $|x|, |y|$ known, we have that

$-|x||y| \leq \Re\{xy\} \leq |x||y|$, where the upper bound is attained when $\arg(x) + \arg(y) = 0$ and the lower bound when $\arg(x) + \arg(y) = \pi$.

Therefore, both problems (1) boil down to the real-valued optimization problem

$$\begin{aligned} \min \quad & -\sum_{l \in P_1} a_l b_l - \sum_{l \in P_2} a_l b_l - \sum_{l \in P_3} a_l b_l \quad (2) \\ \text{s.t.} \quad & 0 \leq a_l \leq A, \quad \forall l \in p_x^- \\ & 0 \leq b_l \leq B, \quad \forall l \in p_q^- \\ & \sum_{l \in p_x^-} a_l^2 \leq e_x \\ & \sum_{l \in p_q^-} b_l^2 \leq e_q, \end{aligned}$$

where a_l, b_l represent $|X_l|, |Q_l|$, respectively, and $A := \min_{j \in p_x^+} |X_j|, B := \min_{j \in p_q^+} |Q_j|$. Note also that we have relaxed the equality constraints to inequality constraints as the objective function of (2) is decreasing in all a_i, b_i , so the optimum of (2) has to satisfy the relaxed inequality constraints with equality, because of the elementary property that $|p_x^-|A^2 \geq e_x, |p_q^-|B^2 \geq e_q$. Recall that in the first sum only $\{a_i\}$ are known and in the second only $\{b_i\}$, whereas in the third all variables are unknown.

We have reduced the original problem to a single optimization program, which, however, is not convex unless $p_x^- \cap p_q^- = \emptyset$. It is easy to check that the constraint set is convex and compact; however, the bilinear function $f(x, y) := xy$ is convex in each argument alone, but *not* jointly. We consider the re-parametrization of the decision variables $z_i = a_i^2$ for $i \in p_x^-$, and $y_i = b_i^2$ for $i \in p_q^-$, we set $Z := A^2, Y := B^2$ and get the equivalent problem:

$$\begin{aligned} \min \quad & -\sum_{i \in P_1} b_i \sqrt{z_i} - \sum_{i \in P_2} a_i \sqrt{y_i} - \sum_{i \in P_3} \sqrt{z_i} \sqrt{y_i} \quad (3) \\ \text{s.t.} \quad & 0 \leq z_i \leq Z, \quad \forall i \in p_x^- \\ & 0 \leq y_i \leq Y, \quad \forall i \in p_q^- \\ & \sum_{i \in p_x^-} z_i \leq e_x \\ & \sum_{i \in p_q^-} y_i \leq e_q. \end{aligned}$$

The necessary and sufficient conditions on optimality are presented in appendix 12.1.

Optimal lower/upper bounds: Let us denote the optimal value of (3) by $v_{\text{opt}} \leq 0$. Then the optimal lower bound (LB) and upper bound (UB) for the distance es-

timization problem under consideration are given by

$$LB = \sqrt{\hat{D} + 4v_{\text{opt}}} \quad (4)$$

$$UB = \sqrt{\hat{D} - 4v_{\text{opt}}} \quad (5)$$

$$\hat{D} := \|X\|_2^2 + \|Q\|_2^2 - 4 \sum_{l \in P_0} \Re\{X_l Q_l\}.$$

Remark 1 Interestingly, the widely used convex solver `cvx` [32] cannot directly address (3)—the issue is that it fails to recognize convexity of the objective functions. For a numerical solution, we consider solving a relaxed version of the minimization problem (1), where equality constraints are replaced by \leq inequalities. We note that this problem is not equivalent to (1), but still provides a viable lower bound. An upper bound can be obtained by (cf. (4), (5)):

$$UB = \sqrt{2\hat{D} - LB^2}.$$

We test the tightness of such approach in the experimental section 10.

7 Exact Solutions

In this section, we study algorithms for obtaining exact solutions for the optimization problem (3). By *exact*, we mean that the optimal value is obtained in a finite number of computations as opposed to when using a numerical scheme for convex optimization. In the latter case, an approximate solution is obtained by means of an iterative scheme which converges with finite precision. Before addressing the general problem, we briefly recap a special case that was dealt with in [12], where the sequence \mathbf{Q} was assumed to be uncompressed. In this case, an exact solution is provided via the *waterfilling* algorithm, which constitutes a key building block for obtaining exact solutions to the general problem later on. We then proceed to study the properties of optimal solutions; our theoretical analysis gives rise to an *exact* algorithm, cf. Sec. 8.2.

7.1 Waterfilling Algorithm.

The case that \mathbf{Q} is uncompressed is a special instance of our problem with $p_q^- = \emptyset$, whence also $P_2 = P_3 = \emptyset$. The problem is strictly convex, and (A-2d) yields

$$z_i = \left(\frac{b_i}{\lambda + \alpha_i} \right)^2 \Leftrightarrow a_i = \frac{b_i}{\lambda + \alpha_i} \quad (6)$$

In such a case, the strict convexity guarantees the existence of a *unique* solution satisfying the KKT conditions as given by the *waterfilling* algorithm, cf. Fig.

5. The algorithm progressively increases the unknown coefficients a_i until saturation, i.e., until they reach A , in which case they are fixed. The set C is the set of non saturated coefficients at the beginning of each iteration, whereas R denotes the “energy reserve,” i.e., the energy that can be used to increase the non saturated coefficients; v_{opt} denotes the optimal value.

Waterfilling algorithm

Inputs: $\{b_i\}_{i \in p_x^-}, e_x, A$

Outputs: $\{a_i\}_{i \in p_x^-}, \lambda, \{\alpha_i\}_{i \in p_x^-}, v_{opt}, R$

1. Set $R = e_x, C = p_x^-$
2. **while** $R > 0$ and $C \neq \emptyset$ **do**
3. set $\lambda = \sqrt{\frac{\sum_{i \in C} b_i^2}{R}}, a_i = \frac{b_i}{\lambda}, i \in C$
4. **if** for some $i \in C, a_i > A$ **then**
5. $a_i = A, C \leftarrow C - \{i\}$
6. **else break;**
7. **end if**
8. $R = e_x - (|p_x^-| - |C|)A^2$
9. **end while**
10. Set $v_{opt} = -\sum_{i \in p_x^-} a_i b_i$ and

$$\alpha_i = \begin{cases} 0, & \text{if } a_i < A \\ \frac{b_i}{A} - \lambda, & \text{if } a_i = A \end{cases}$$

Fig. 5: Waterfilling algorithm for optimal distance estimation between a compressed and an uncompressed sequence

As a shorthand notation, we write $\mathbf{a} = \text{waterfill}(\mathbf{b}, e_x, A)$. Note that in this case the problem (2) for $P_2 = P_3 = \emptyset$ is convex, so the solution can be obtained via the KKT conditions to (2), which are different from those for the re-parameterized problem (3); this was done in [12]. The analysis and straightforward extensions are summarized in Lemma 1.

Lemma 1 (Exact solutions)

1. If either $p_x^- = \emptyset$ or $p_q^- = \emptyset$ (i.e., when at least one of the sequences is uncompressed) we can obtain an exact solution to the optimization problem (2) via the waterfilling algorithm.
2. If $P_3 = p_x^- \cap p_q^- = \emptyset$, i.e., when the two compressed sequences do not have any common unknown coefficients, the problem is decoupled in \mathbf{a}, \mathbf{b} , and the waterfilling algorithm can be used separately to obtain exact solutions to both unknown vectors.
3. If $P_1 = P_2 = \emptyset$, i.e., when both compressed sequences have the same discarded coefficients, the optimal value is simply equal to $-\sqrt{e_x} \sqrt{e_q}$, but there is no unique solution for \mathbf{a}, \mathbf{b} .

Proof The first two cases are obvious. For the third one, note that it follows immediately from the Cauchy–Schwartz inequality that $-\sum_{l \in P_3} a_l b_l \geq -\sqrt{e_x} \sqrt{e_q}$, and in this case this is also attainable. Just consider for example, $a_l = \sqrt{\frac{e_x}{|P_3|}}, b_l = \sqrt{\frac{e_q}{|P_3|}}$, which is feasible because $|p_x^-|A^2 \geq e_x, |p_q^-|B^2 \geq e_q$, as follows by compression with the high-energy coefficients. ■

We have shown how to obtain exact optimal solutions for special cases. To derive efficient algorithms for the general case, we first study and establish some properties of the optimal solution of (3).

Theorem 1 (Properties of optimal solutions)

Let an augmented optimal solution of (2) be denoted by $(\mathbf{a}^{opt}, \mathbf{b}^{opt})$; where $\mathbf{a}^{opt} := \{a_i^{opt}\}_{i \in p_x^- \cup p_q^-}$ denotes the optimal solution extended to include the known values $|X_l|_{l \in P_2}$, and $\mathbf{b}^{opt} := \{b_i^{opt}\}_{i \in p_x^- \cup p_q^-}$ denotes the optimal solution extended to include the known values $|Q_l|_{l \in P_1}$. Let us further define $e'_x = e_x - \sum_{l \in P_1} a_l^2, e'_q = e_q - \sum_{l \in P_2} b_l^2$. We then have the following:

1. The optimal solution satisfies¹

$$\mathbf{a}^{opt} = \text{waterfill}(\mathbf{b}^{opt}, e_x, A), \quad (7a)$$

$$\mathbf{b}^{opt} = \text{waterfill}(\mathbf{a}^{opt}, e_q, B). \quad (7b)$$

In particular, it follows that $a_i^{opt} > 0$ iff $b_i^{opt} > 0$ and that $\{a_i^{opt}\}, \{b_i^{opt}\}$ have the same ordering. In addition, $\min_{l \in P_1} a_l \geq \max_{l \in P_3} a_l, \min_{l \in P_2} b_l \geq \max_{l \in P_3} b_l$.

2. If at optimality it holds that $e'_x e'_q > 0$ there exists a multitude of solutions. One solution (\mathbf{a}, \mathbf{b}) satisfies

$$a_l = \sqrt{\frac{e'_x}{|P_3|}}, b_l = \sqrt{\frac{e'_q}{|P_3|}} \text{ for all } l \in P_3, \text{ whence}$$

$$\lambda = \sqrt{\frac{e'_q}{e'_x}} \quad \mu = \sqrt{\frac{e'_x}{e'_q}}, \quad (8a)$$

$$\alpha_i = \beta_i = 0 \quad \forall i \in P_3. \quad (8b)$$

In particular, $\lambda\mu = 1$ and the values e'_x, e'_q need to be solutions to the following set of nonlinear equations:

$$\sum_{l \in P_1} \min\left(b_l^2 \frac{e'_x}{e'_q}, A^2\right) = e_x - e'_x, \quad (9a)$$

$$\sum_{l \in P_2} \min\left(a_l^2 \frac{e'_q}{e'_x}, B^2\right) = e_q - e'_q. \quad (9b)$$

3. At optimality, it is not possible to have $e'_x e'_q = 0$ unless $e'_x = e'_q = 0$.

¹ This has a natural interpretation as the Nash equilibrium of a 2-player game [33] in which Player 1 seeks to minimize the objective of (3) with respect to \mathbf{z} , and Player 2 seeks to minimize the same objective with respect to \mathbf{y} .

4. Consider the vectors \mathbf{a}, \mathbf{b} with $a_l = |X_l|$, $l \in P_2$, $a_l = |X_l|$, $l \in P_1$ and

$$\{a_l\}_{l \in P_1} = \text{waterfill}(\{b_l\}_{l \in P_1}, e_x, A), \quad (10a)$$

$$\{b_l\}_{l \in P_2} = \text{waterfill}(\{a_l\}_{l \in P_2}, e_q, B). \quad (10b)$$

If $e_x \leq |P_1|A^2$ and $e_q \leq |P_2|B^2$, whence $e'_x = e'_q = 0$, then by defining $a_l = b_l = 0$ for $l \in P_3$, we obtain a globally optimal solution (\mathbf{a}, \mathbf{b}) .

Proof See appendix 12.2.

Remark 2 One may be tempted to think that an optimal solution can be derived by waterfilling for the coefficients of $\{a_l\}_{l \in P_1}, \{b_l\}_{l \in P_2}$ separately, and then allocating the remaining energies e'_x, e'_q to the coefficients in $\{a_l, b_l\}_{l \in P_3}$ leveraging the Cauchy–Schwartz inequality, the value being $-\sqrt{e'_x} \sqrt{e'_q}$. However, the third and fourth parts of Theorem 1 state that this is *not* optimal unless $e'_x = e'_q = 0$.

We have shown that there are two possible cases for an optimal solution of (2): either $e'_x = e'_q = 0$ or $e'_x, e'_q > 0$. The first case is easy to identify by checking whether (10) yields $e'_x = e'_q = 0$. If this is not the case, we are in the latter case and need to find a solution to the set of nonlinear equations (9).

Consider the mapping $T : \mathbb{R}_+^2 \rightarrow \mathbb{R}_+^2$ defined by

$$T((x_1, x_2)) := \left(e_x - \sum_{l \in P_1} \min\left(b_l^2 \frac{x_1}{x_2}, A^2\right), e_q - \sum_{l \in P_2} \min\left(a_l^2 \frac{x_2}{x_1}, B^2\right) \right) \quad (11)$$

The set of nonlinear equations of (9) corresponds to a positive *fixed point* of T , i.e., $(e'_x, e'_q) = T(e'_x, e'_q)$, $e'_x, e'_q > 0$. Calculating a fixed point of T may at first seem involved, but it turns out that this can be accomplished *exactly* and with minimal overhead. The analysis can be found in appendix 12.3, where we prove that this problem is no different from the simplest numerical problem: finding the root of a scalar linear equation.

Remark 3 We note that coefficients $\{a_l\}_{P_2}, \{b_l\}_{P_1}$ are already sorted because of the way we perform compression—by storing high-energy coefficients. It is plain to check that all other operations can be efficiently implemented, with the average complexity being *linear* ($\Theta(N)$)—hence the term *minimum-overhead* algorithm.

Remark 4 (Extensions) It is straightforward to see that our approach can be applied without modification to the case that each point is compressed using a different number of coefficients. Additionally, we note here two important extensions of our problem formulation and optimal algorithm.

1. Consider the case that a data point is a *countably infinite* sequence. For example, we may express a continuous function via its Fourier series, or Chebyshev polynomials expansion. In that case, surprisingly enough, our algorithm can be applied without any alteration. This is because P_0, P_1, P_2 are finite sets as defined above, whereas P_3 is now infinite. The energy allocation in P_3 can be computed exactly by the same procedure (cf. appendix 12.3) and then in P_3 the Cauchy–Schwartz inequality is again applied².
2. Of particular interest is the case that data are compressed using an *over-complete* basis (also known as *frame* in the signal-processing literature [29]); this approach has recently become popular in the compressed-sensing framework [34]. Our method can be extended to handle this important general case, by storing the compression errors corresponding to each given basis, and calculating the lower/upper bounds in each one separately using our approach. We leave this direction for future research work.

8 Algorithm for Optimal Distance Estimation

In this section, we present an algorithm for obtaining the *exact optimal* upper and lower bounds on the distance between the original sequences, when fully leveraging all information available given their compressed counterparts. First, we present a simple numerical scheme using a convex solver such as *cvx* [32] and then use our theoretical findings to derive an analytical algorithm which we call ‘double-waterfilling’.

8.1 Convex Programming

We let $M := N - |P_0|$, and consider the nontrivial case $M > 0$. Following the discussion in Sec. 6, we set the $2M \times 1$ vector $\mathbf{v} = (\{a_l\}_{l \in P_1 \cup P_2 \cup P_3}, \{b_l\}_{l \in P_1 \cup P_2 \cup P_3})$ and consider the following convex problem directly amenable to a numerical solution via a solver such as *cvx*:

$$\begin{aligned} \min \quad & \sum_{l \in P_1 \cup P_2 \cup P_3} (a_l - b_l)^2 \\ \text{s.t.} \quad & a_l \leq A, \forall l \in p_x^-, \quad b_l \leq B, \forall l \in p_q^- \\ & \sum_{l \in p_x^-} a_l^2 \leq e_x, \quad \sum_{l \in p_q^-} b_l^2 \leq e_q \\ & a_l = |X_l|, \forall l \in P_2, \quad b_l = |Q_l|, \forall l \in P_1 \end{aligned}$$

² The proof of optimality in this case assumes a finite subset of P_3 and applies the same conditions of optimality that were leveraged before; in fact particular selection of the subset is not of importance, as long as its cardinality is large enough to accommodate the computed energy allocation (e'_x, e'_q) .

The lower bound (LB) can be obtained by adding $D' := \sum_{l \in P_0} |X_l - Q_l|^2$ to the optimal value of (1) and taking the square root; then the upper bound is given by $UB = \sqrt{2D' - LB^2}$, cf. (4).

Double-waterfilling algorithm

Inputs: $\{b_l\}_{l \in P_1}, \{a_l\}_{l \in P_2}, e_x, e_q, A, B$

Outputs: $\{a_l, \alpha_l\}_{l \in P_x}, \{b_l, \beta_l\}_{l \in P_q}, \lambda, \mu, v_{\text{opt}}$

1. **if** $p_x \cap p_q = \emptyset$ **then** use waterfilling algorithm (see Lemma 1 parts 1,2); **return; endif**
2. **if** $p_x = p_q$ **then** set $a_l = \sqrt{\frac{e_x}{|P_3|}}, b_l = \sqrt{\frac{e_q}{|P_3|}}, \alpha_l = \beta_l = 0$ for all $l \in p_x, v_{\text{opt}} = -\sqrt{e_x e_q}$; **return; endif**
3. **if** $e_x \leq |P_1|A^2$ **and** $e_q \leq |P_2|B^2$ **then**
 $\{a_l\}_{l \in P_1} = \text{waterfill}(\{b_l\}_{l \in P_1}, e_x, A)$
 $\{b_l\}_{l \in P_2} = \text{waterfill}(\{a_l\}_{l \in P_2}, e_q, B)$
 with optimal values $v_{\text{opt}}^{(a)}, v_{\text{opt}}^{(b)}$, respectively.
4. Set $a_l = b_l = \alpha_l = \beta_l = 0$ for all $l \in P_3, v_{\text{opt}} = -v_{\text{opt}}^{(a)} - v_{\text{opt}}^{(b)}$; **return;**
5. **endif**
6. Calculate the root $\bar{\gamma}$ as in Remark 5 (appendix 12.3) and define e'_x, e'_q as in (A-8).
7. Set
 $\{a_l\}_{l \in P_1} = \text{waterfill}(\{b_l\}_{l \in P_1}, e_x - e'_x, A)$
 $\{b_l\}_{l \in P_2} = \text{waterfill}(\{a_l\}_{l \in P_2}, e_q - e'_q, B)$
 with optimal values $v_{\text{opt}}^{(a)}, v_{\text{opt}}^{(b)}$, respectively.
8. Set $a_l = \sqrt{\frac{e'_x}{|P_3|}}, b_l = \sqrt{\frac{e'_q}{|P_3|}}, \alpha_l = \beta_l = 0, l \in P_3$ and set $v_{\text{opt}} = -v_{\text{opt}}^{(a)} - v_{\text{opt}}^{(b)} - \sqrt{e'_x e'_q}$

Fig. 6: Double-waterfilling algorithm for optimal distance estimation between two compressed sequences.

8.2 Double-waterfilling

Leveraging our theoretical analysis, we derive a simple efficient algorithm to obtain an *exact* solution to the problem of finding tight lower/upper bound on the distance of two compressed sequences; we call this the “double-waterfilling algorithm.” The idea is to obtain an exact solution of (2) based on the results of Theorems 1, 2, and Remark 5; then the lower/upper bounds are given by (4), (5). The algorithm is described in Fig. 6; its proof of optimality follows immediately from the preceding theoretical analysis.

9 Mining in the compressed domain

In the experimental section, we will demonstrate the performance of our methodology when operating di-

rectly in the compressed domain for distance-based operations. We will use two common search and mining tasks to showcase our methodology: (i) the k -NN search and (ii) the k -Means clustering. Performing such operations in the compressed domain may require modifications in the original algorithms, because of the uncertainty introduced in the distance estimation in the compressed domain. We discuss these modifications in the sections that follow. We also elaborate on previous state-of-art approaches.

9.1 k -NN search in the compressed domain

Finding the closest points to a given query point is an elementary subroutine to many problems in search, classification, and prediction. A brute-force approach via exhaustive search on the uncompressed data can incur a prohibitive cost [22]. Thus, the capability to work directly in a compressed domain provides a very practical advantage to any algorithm.

In this context, the k -NN problem [35] in the compressed domain can be succinctly described as:

k -NN PROBLEM: *Given a compressed query representation \mathbf{Y}_Q and $k \in \mathbb{Z}_+$, find the k closest elements $\mathbf{X}^{(i)} \in \mathcal{DB}$ with respect to the ℓ_2 -norm through their compressed representations $\mathbf{Y}^{(i)}$.*

Various diverse approaches exist to tackle this problem efficiently and robustly; cf., [36]. Here, we compare our methodology with two algorithmic approaches of k -NN search in a low-dimensional space: (i) Randomized projection-based k -NN and (ii) PCA-based k -NN. We describe these approaches in more detail.

Approximate k -NN using Randomized Projections (RP): One of the most established approaches for low-dimensional data processing is through the *Johnson Lindenstrauss (JL) Lemma*³:

Lemma 2 (JL Lemma) *Let $\mathcal{X} = \{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(V)}\}$ be any arbitrary collection of V points in N dimensions. For an isometry constant $\epsilon \in (0, 1)$, we can construct with high*

³ While JL Lemma applies for any set of points $\{\mathbf{X}^{(1)}, \dots, \mathbf{X}^{(V)}\}$ in high dimensions, more can be achieved if sparse representations of $\mathbf{X}^{(i)}, \forall i$, are known to exist *a priori*. Compressive sensing (CS) [37] [34] roughly states that a sparse signal, compared with its ambient dimension, can be *perfectly reconstructed* from far fewer samples than dictated by the well-known Nyquist–Shannon theorem. To this extent, CS theory exploits the sparsity to extend the JL Lemma to more general signal classes, not restricted to a collection of points \mathcal{X} . As a by-product of this extension, the CS version of the JL Lemma constitutes the Restricted Isometry Property (RIP).

Algorithm 1 RP-based approximate k -NN algorithm

Input: $k \in \mathbb{Z}_+$, \mathbf{Q} , $\epsilon \in (0, 1)$ \mathcal{X} with the transformed elements $\mathbf{X}^{(i)} = \mathcal{F}(\mathbf{x}^{(i)})$.	$\triangleright k$: # of Nearest Neighbors, \mathbf{Q} : the transformed <i>uncompressed</i> query
---	---

1: Select appropriate $d = \mathcal{O}(\epsilon^{-2} \log(V))$ 2: Construct $\Phi \in \mathbb{R}^{d \times N}$ where Φ_{ij} is i.i.d. $\sim \mathcal{N}(0, \frac{1}{\sqrt{d}})$ or $\sim \text{Ber}\{\pm \frac{1}{\sqrt{d}}\}$ or according to (12). 3: for each $\mathbf{X}^{(i)} \in \mathcal{X}$ do 4: Compute $\mathbf{Y}^{(i)} = \Phi \mathbf{X}^{(i)} \in \mathbb{C}^d$ 5: end for	(PRE-PROCESSING STEP)
--	-----------------------

6: Compute $\mathbf{Y}_Q = \Phi \mathbf{Q} \in \mathbb{C}^d$. 7: for each $\mathbf{Y}^{(i)}$ do 8: Compute $\ \mathbf{Y}_Q - \mathbf{Y}^{(i)}\ _2$ 9: end for 10: Sort and keep the k -closest $\mathbf{Y}^{(i)}$'s, in the ℓ_2 -norm sense.	(REAL-TIME EXECUTION STEP)
--	----------------------------

probability a linear mapping $\Phi : \mathbb{C}^N \rightarrow \mathbb{C}^d$, where $d = \mathcal{O}(\epsilon^{-2} \log(V))$, such that

$$(1 - \epsilon) \leq \frac{\|\Phi(\mathbf{X}^{(i)} - \mathbf{X}^{(j)})\|_2^2}{\|\mathbf{X}^{(i)} - \mathbf{X}^{(j)}\|_2^2} \leq (1 + \epsilon)$$

for all $\mathbf{X}^{(i)}, \mathbf{X}^{(j)} \in \mathcal{X}$.

Therefore, instead of working in the ambient space of N dimensions, we can construct a linear, nearly *isometric map* Φ that projects the data onto a lower subspace, *approximately preserving their relative distances* in the Euclidean sense. Variants of this approach have also been proposed in [38]. As we are not interested in *recovering* the entries of $\mathbf{X}^{(i)}$ in the d -dimensional space, rather than just performing data manipulations in this domain, one can control the distortion ϵ so that tasks such as classification and clustering can be performed quite accurately with low computational cost. However, we underline that the JL guarantees are probabilistic and asymptotic, whereas the lower and upper bounds provided by our technique cannot be violated.

The topic of constructing matrices Φ that satisfy the JL Lemma with nice properties (e.g., deterministic construction, low space-complexity for storage, cheap operations using Φ) is still an open question, although many approaches have been proposed. Similar questions also appear under the tag of *locality-sensitive hashing*, where *sketching* matrices “sense” the signal under consideration. Fortunately, many works have proved the existence of random universal matrix ensembles that satisfy the JL Lemma with *overwhelming probability*, thus ignoring the deterministic construction property.⁴ Representative examples include

⁴ Recent developments [39] describe *deterministic constructions* of Φ in polynomial time, based on the fact that the data \mathcal{X} is known *a priori* and fixed. The authors in [39] propose the NuMax algorithm, a SemiDefinite Programming (SDP) solver for convex nuclear norm minimization over ℓ_∞ -norm and pos-

random Gaussian matrices [40] and random binary (Bernoulli) matrices [41][22]. Ailon and Chazelle [19] propose a fast JL transform for the k -NN problem with faster execution time than its predecessors. Achlioptas [22] proposes a randomized construction for Φ , both simple and fast, which is suitable for standard SQL-based database environments; each entry Φ_{ij} independently takes one of the following values:

$$\Phi_{ij} = \begin{cases} 1 & \text{with probability } \frac{1}{6}, \\ 0 & \text{with probability } \frac{2}{3}, \\ -1 & \text{with probability } \frac{1}{6}. \end{cases} \quad (12)$$

This has the additional advantage of producing a *sparse* transformation matrix Φ which results in computation savings at the data-compression step.

In our experiments, we shall refer to the Gaussian distributed projection matrix as GRP, the Bernoulli distributed projection matrix as BRP and Achlioptas’ construction ARP. We omit sophisticated constructions of Φ because of their increased implementation complexity.

Using linear maps $\Phi \in \mathbb{R}^{d \times N}$ or $\Phi \in \mathbb{C}^{d \times N}$ satisfying the JL Lemma, one can compress $\mathbf{X}^{(i)}$ as $\mathbf{Y}^{(i)} = \Phi \mathbf{X}^{(i)} \in \mathbb{C}^d$ and store only $\mathbf{Y}^{(i)}$ for further processing. Given a compressed representation of a query \mathbf{Q} , $\mathbf{Y}_Q = \Phi \mathbf{Q}$, one can compute the distances of each $\mathbf{Y}^{(i)}$ to \mathbf{Y}_Q and pick the k nearest points in the Euclidean sense. We provide a pseudo-code description of the above in Algorithm 1.

Overall, Algorithm 1 requires $\mathcal{O}(\epsilon^{-2}VN \log(V))$ time to *preprocess* the entries of \mathcal{DB} and $\mathcal{O}(\epsilon^{-2}(V + N)\log(V))$ double-sized space-complexity, in the

itive semidefinite constraints. However, NuMax has $\mathcal{O}(C + N^3 + N^2C^2)$ time-complexity per iteration and overall $\mathcal{O}(C^2)$ space-complexity, where $C := \binom{V}{2}$; this renders NuMax prohibitive for real-time applications. Such an approach is not included in our experiments, but we mention it here for completeness.

Gaussian case. In the other two cases, the space complexity can be further reduced thanks to the binary representation of Φ . Given a query \mathbf{Y}_Q , Algorithm 1 requires $\mathcal{O}(\max\{V, N\} \cdot \epsilon^{-2} \log(V))$ time-cost.

Approximate k -Nearest Neighbors using PCA: Instead of projecting onto a randomly chosen low-dimensional subspace, one can use the *most informative* subspaces to construct a projection matrix, based on \mathcal{X} . PCA-based k -NN relies on this principle: let $\mathbf{X} := [\mathbf{X}^{(1)} \ \mathbf{X}^{(1)} \ \dots \ \mathbf{X}^{(V)}] \in \mathbb{C}^{N \times n}$ be the data matrix. Given \mathbf{X} , one can compute the Singular Value Decomposition (SVD) $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$ to identify the d most important subspaces of \mathbf{X} , spanned by the d dominant singular vectors in \mathbf{U} . In this case, $\Phi := \mathbf{U}(1 : d, :)$ works as a low-dimensional linear map, biased by the information contained in \mathbf{X} .

The main shortcoming of PCA-based k -NN search is the computation of the SVD of \mathbf{X} ; generally, such an operation has cubic complexity in the number of entries in \mathbf{X} . Moreover, PCA-based projection provides no guarantees on the order of distortion in the compressed domain: While in most cases $\Phi := \mathbf{U}(1 : d, :)$ outperforms RP-based approaches with JL guarantees, one can construct test cases where the pairwise point distances are heavily distorted such that points in \mathcal{X} might be mapped to a single point [22] [39]. Finally, note that computation of the SVD requires the presence of the entire dataset, whereas approaches such as ours operate on a per-object basis.

Optimal bounds-based k -NN: Our approach can easily be adapted to perform k -NN search operations in the compressed domain. Similar to Algorithm 1, instead of computing random projection matrices, we keep the largest coefficients for each transformed signal representation $\mathbf{X}^{(i)}$ (in Fourier, Wavelet or other basis) and also record the total discarded energy per object. Following a similar approach to compress the input query \mathbf{Q} , say \mathbf{Y}_Q , we perform the optimal bounds procedure to obtain upper (u_b) and lower bounds (ℓ_b) of the distance in the original domain. Therefore, we do not have only one distance, but can compute three *distance proxies* based on the upper and lower bounds on the distance.

- (i) We use the lower bound ℓ_b as indicator of how close the uncompressed $\mathbf{x}^{(i)}$ is to the uncompressed query \mathbf{q} .
- (ii) We use the upper bound u_b as indicator of how close the uncompressed $\mathbf{x}^{(i)}$ is to the uncompressed query \mathbf{q} .
- (iii) We define the average metric $\frac{\ell_b + u_b}{2}$ as indicator of how close the uncompressed $\mathbf{x}^{(i)}$ is to the uncompressed query \mathbf{q} .

In the experimental section, we evaluate the performance of these three metrics, and show that the last metric based on the average distance bound provides the most robust performance.

9.2 k -Means clustering in the compressed domain

Clustering is a rudimentary mining task for summarizing and visualizing large amounts of data. Specifically, the k -clustering problem is defined as follows:

k -CLUSTERING PROBLEM: Given a DB containing V compressed representations of $\mathbf{x}^{(i)}, \forall i$, and a target number of clusters k , group the compressed data into k clusters in an accurate way through their compressed representations.

This is an *assignment* problem and is in fact NP-hard [42]. Many approximations to this problem exist, one of the most widely-used algorithms being the k -Means clustering algorithm [43]. Formally, k -Means clustering involves partitioning the V vectors into k clusters, i.e., into k disjoint subsets $G^{(t)}$ ($1 \leq t \leq k$) with $\cup_t G^{(t)} = V$, such that the sum of intraclass variances

$$V := \sum_{t=1}^k \sum_{\mathbf{x}^{(i)} \in G^{(t)}} \|\mathbf{x}^{(i)} - C^{(t)}\|^2, \quad (13)$$

is minimized, where $C^{(t)}$ is the *centroid* of the k -th cluster.

There also exist other formalizations for data clustering, based on either hierarchical clustering (“top-down” and “bottom-up” constructions, cf. [44]); flat or centroid-based clustering, or on spectral-based clustering [45]. In our subsequent discussions, we focus on the k -Means algorithm because of its widespread use and fast runtime. Note also that k -Means is easily amenable for use by our methodology owing to its computation of distances between objects and the derived centroids.

Similar to the k -NN problem case, we consider low-dimensional embedding matrices based on both PCA and randomized constructions [46][15] [20]. We note that [20] theoretically proves that a specific random matrix construction achieves a $(2 + \epsilon)$ -optimal k -partition of the points in the compressed domain in $\mathcal{O}(VN \frac{k}{\epsilon^2 \log(N)})$ time. Based on simulated annealing clustering heuristics, [21] proposes an iterative procedure where sequential k -means clustering is performed, with increasing projection dimensions d , for better clustering performance. We refer the reader to [20] for a recent discussion of the above approaches.

Similar, in spirit, to our approach is the work of [42]. There, the authors propose 1-bit Minimum Mean

Algorithm 2 Optimal bounds-based k -Means

Input: $k \in \mathbb{Z}_+, \mathcal{Y} = \{\mathbf{Y}^{(1)}, \dots, \mathbf{Y}^{(V)}\}$	$\triangleright k$: # of clusters, $\mathbf{Y}^{(i)}$: Compressed data vectors.
1: Select randomly k centroids $C^{(t)}, t = 1, \dots, k$, in the compressed domain.	(INITIALIZATION STEP)
2: while $\mathbf{Y}^{(i)}$ assignment to $C^{(t)}$ changes do	
3: Compute optimal lower ℓ_b and upper u_b bounds between each $\mathbf{Y}^{(i)}$ and $C^{(t)}, \forall i, t$.	
4: For each pair $\mathbf{Y}^{(i)}, C^{(t)}$ calculate a distance metric $m_{i,t}$, based on ℓ_b and u_b .	
5: Assign $\mathbf{Y}^{(i)}$'s to groups $G^{(t)}$ such that: $G^{(t)} = \{\mathbf{Y}^{(i)} \mid m_{i,t} \leq m_{i,q}, \forall q \neq t\}$.	(ASSIGNMENT STEP)
6: Update the centroids: $C^{(t)} = \frac{1}{ G^{(t)} } \sum_{\mathbf{Y}^{(i)} \in G^{(t)}} \mathbf{Y}^{(i)}$.	(UPDATE STEP)
7: end while	

Square Error (MMSE) quantizers per dimension and cluster, and provide guarantees for cluster preservation in the compressed domain.

Optimal bounds-based k -Means: To describe how to use our proposed bounds in a k -clustering task, let $G^{(t)}, t = 1, \dots, k$, be the k groups of a partition with centroids $C^{(t)}, t = 1, \dots, k$. We use a modification of Lloyd's algorithm [43] which consists of the following steps:

Assignment step: Let $C^{(t)}, t = 1, \dots, k$, be the current centroids.⁵ For each compressed sequence $\mathbf{Y}^{(i)} \in \mathcal{DB}$, we compute the corresponding upper u_b and lower ℓ_b bounds with respect to every centroid $C^{(t)}$. We use a distance metric $m_{i,t}$, based on u_b and ℓ_b to decide the assignment of each $\mathbf{Y}^{(i)}$ to one of the centroids, i.e.,

$$G^{(t)} = \{\mathbf{Y}^{(i)} \mid m_{i,t} \leq m_{i,q}, \forall q \neq t\}.$$

Update step: To update the centroids $C^{(t)}$, we use the average rule using the current distribution of points at each cluster, i.e.,

$$C^{(t)} = \frac{1}{|G^{(t)}|} \sum_{\mathbf{Y}^{(i)} \in G^{(t)}} \mathbf{Y}^{(i)}.$$

Distance to new centroids: Recall that each of the compressed objects has information only about its high-energy coefficients. This set of coefficients may be different across objects. So, during the above averaging operation for computing the centroids, we may end up with the new centroids having (potentially) all coefficient positions filled with some energy. However, this does not pose a problem for the distance computation because the waterfilling algorithm can compute distance estimates even between compressed sequences with different number of coefficients. Therefore, we

⁵ For centroid initialization, one can choose $C^{(t)}$ to be (i) completely random points in the compressed domain; (ii) set randomly to one of the compressed representations of $\mathbf{X}^{(i)} \in \mathcal{DB}$, or (iii) use the better performing k -Means++ initialization algorithm [47, 42].

exploit all information available in the computed centroid, as this does not increase the space complexity of our technique. Alternatively, one could keep only the top high-energy coefficients for the new centroid. However, there is no need to discard this extra information.

The above steps are summarized in Algorithm 2.

10 Experiments

Here we conduct a range of experiments to showcase a) the tightness of bounds that we calculate; b) the low runtime to compute those bounds and, c) the comparative quality of various mining tasks when using the optimal distance estimates.

Datasets: We use two datasets: (i) a weblog time-series dataset and (ii) an image dataset consisting of Very Large Scale Integration (VLSI) layouts.

The weblog dataset consists of approximately 2.2 million data values, distilled from past logs of the IBM.com search page. We created time-series by assembling how many times important keywords were entered at the search page. We constructed 2,150 time-series $\mathbf{x}^{(i)} \in \mathbb{R}^{1024}$, corresponding to an equal number of text queries. Each time-series object captures how many times a particular text query was posed for 1024 consecutive days. We transform the data into a compressible form: for each $\mathbf{x}^{(i)}$, we compute its Fourier representation $\mathbf{X}^{(i)} \in \mathbb{C}^N$.

The second dataset considers images consisting of patterns from a database of VLSI layouts. The dataset is comprised of $\sim 150,000$ images of size 512×512 . To support translation-invariant matching from each image, we extract a signature $\mathbf{x}^{(i)} \in \mathbb{R}^{512}$. We describe this process in more detail later. Finally, we represent the resulting signature using the high-energy wavelet coefficients as basis. Detailed information on this application will be provided later on.

We take special care to perform a fair comparison of all techniques. For our approach, per compressed

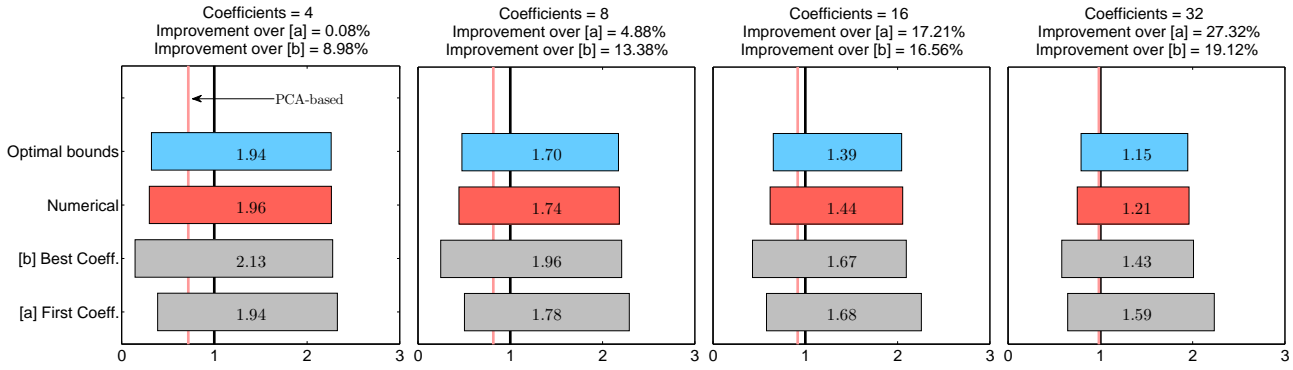


Fig. 7: Comparison of lower and upper bounds for distances for various compression approaches. Distances are shown normalized with respect to the original distance (black vertical line) on the uncompressed data. The red vertical line indicates the bound given the PCA-based approach, which uses all the dataset to compute the appropriate basis.

object we need to store the following: (i) the values $\mathbf{X}(p_x^+)$ of each high-energy coefficient, as s double complex values for the Fourier case (16 bytes each) and s double values for the Wavelet case (8 bytes each); (ii) The positions p_x^+ of the high-energy coefficients, as s integer values (4 bytes each) and, (iii) the total remaining energy e_x of the discarded coefficients, which can be represented with one double variable (8 bytes). Overall, our approach allocates space for $r = \lceil 2s + \frac{s}{2} + 1 \rceil$ double values for the Fourier case and $r = \lceil s + \frac{s}{4} + \frac{1}{2} \rceil$ double values for the Wavelet case. We make sure that all approaches use the *same space*. So, methods that do not require recording of the explicit position of a coefficient, in essence, use more coefficients than our technique.

10.1 Tightness of bounds and time complexity

First, we illustrate how tight the bounds computed by both (i) deterministic and (ii) probabilistic approaches are.

Deterministic approaches: We consider the following schemes:

- (i) *First Coeffs.*: this scheme only exploits the *first* s coefficients of each Fourier-transformed $\mathbf{X}^{(i)}$ to succinctly represent the uncompressed data $\mathbf{x}^{(i)}$. No further computation is performed.
- (ii) *Best Coeffs.*: this scheme only exploits the *best* s coefficients (in magnitude sense) of each Fourier-transformed $\mathbf{X}^{(i)}$ to succinctly represent the uncompressed data $\mathbf{x}^{(i)}$. Similarly to (i), no further computation is performed.

- (iii) *PCA-based.* This technique uses the PCA-based dimensionality reduction approach. Note that this approach requires as input the complete data, and not each object separately. Given the complete dataset, one can compute its SVD to extract the most dominant subspaces that explain the most variance in the data. To achieve dimensionality reduction, one projects the time-series vectors onto the best d -dimensional subspace by multiplying the data points with the set of d dominant left singular vectors.

- (iv) *Optimal bounds - Numerical:* Here, we use off-the-shelf convex solvers to numerically solve problem (1) through second-order optimization schemes. Numerical approaches are not exact and the minimizer lies within a predefined numerical tolerance ϵ . In our experimental setup, we use the well established CVX library where (1) is solved with tolerance $\epsilon = 10^{-8}$.

- (v) *Optimal bounds:* our approach in which the upper and lower bounds on the distance are solved using the closed-form waterfilling ideas described.

To provide a fair comparison, all approaches use the same amount of space per compressed object for all experiments.

Probabilistic approaches: Here, any estimation of distances in the original data space holds only in probability and in the asymptotic sense. The performance recorded represents only an average behavior; i.e., we can always construct adversarial cases where these schemes perform poorly for a fixed projection matrix.

- (i) *RP-based approach:* in this case, we randomly generate GRP/BRP/ARP d -dimensional mappings Φ . As this approach is probabilistic, we perform 10^3

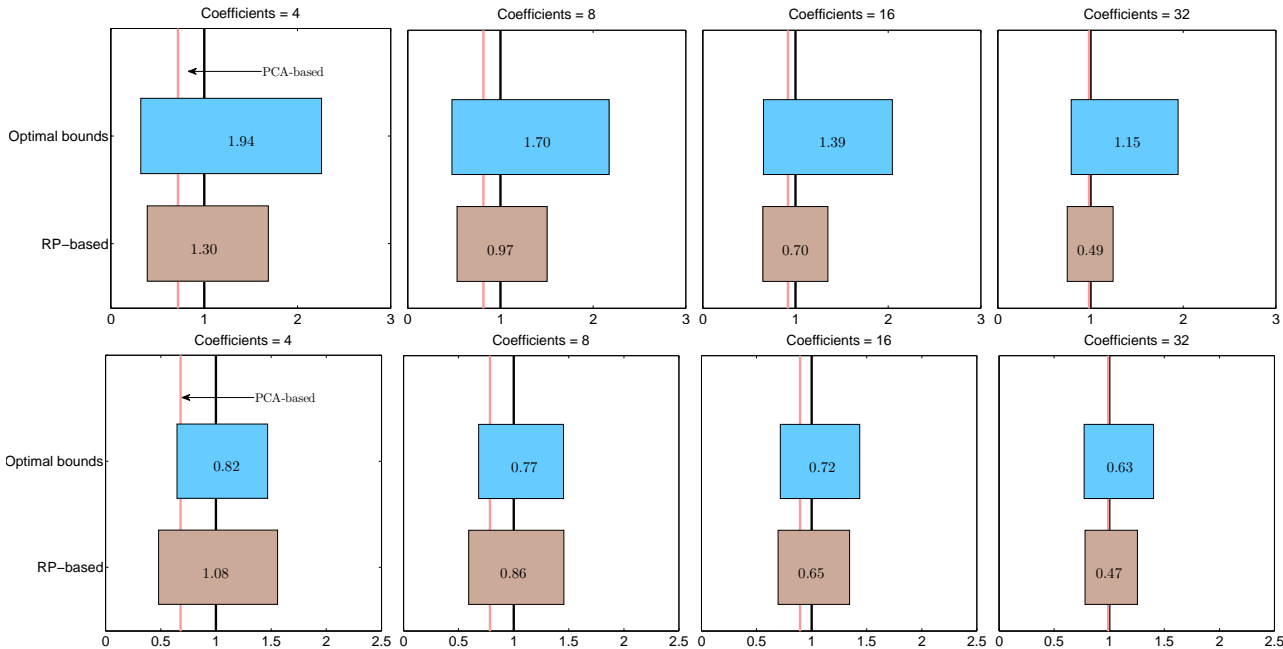


Fig. 8: Comparison of lower and upper bounds for the RP-based approach with the PCA-based and optimal bounds approaches. **Top row:** dense case; **Bottom row:** sparse case. All bounds are shown normalized with respect to the original distance (black vertical line) on the uncompressed data. The average PCA-based distance over all pairwise distances is denoted by a vertical red line.

Monte Carlo iterations to independently generate Φ 's and implicitly extract an approximate range of lower and upper bounds.

Quality of approximation: Using the weblog time-series data, Figure 7 illustrates the results for various compression ratios (i.e. number of coefficients used). Using the `Optimal bounds` scheme, we can achieve up to 27% tighter bounds. The `Optimal bounds - Numerical` approach provides marginally worse results and has a very high computational cost.

The comparison with probabilistic approaches based on Random Projections is shown in Figure 8 (top row). Among the RP-based approaches, the GRP matrices on average attain the best approximation of the original distance. We should highlight, though, that our approach computes nontrivial bounds *for any pair of compressed sequences*, in contrast to RP-based schemes, where the guarantees hold in probability.

Furthermore, to measure the efficiency and robustness of our approach when the data is *naturally* sparse in some basis, i.e., most of the entries in N -dimensions are zero, we synthetically “sparsify” the weblog data: given each uncompressed sequence $\mathbf{X}^{(i)}$, we assume that $\mathbf{X}^{(i)}$ is perfectly represented by using only $3s$ Fourier coefficients, where $s = \{16, 32, 64\}$; i.e., we subsample the signal such that only $3s$ among N coefficients are nonzero. Then, we keep s coefficients. Fig-

ure 8 (bottom row) illustrates the performance of our approach as compared to RP-based and PCA-based approaches. As the data under consideration are sparse, e_x and e_q estimates are tighter, providing better upper and lower bounds than in the non-sparse case (see Figure 8, top row).

Running time: The time complexity of each approach under comparison is given in Figure 9. The graph reports the average running time (in msec) for computing the distance estimates between one pair of sequences. It is evident that the proposed analytical solution based on double-waterfilling is at least two orders of magnitude faster than the numerical approach. More importantly, the optimal solution through waterfilling is not computationally burdening: competing approaches require 1-2 msec for computation, whereas the waterfilling approach requires around 2.5 msec. The small additional time is attributed to the fact that the algorithm distributes the currently remaining energy over two to three iterations, thus incurring only minimal overhead. The numerical solution runs for more than 1 sec and is considered impractical for large mining tasks.

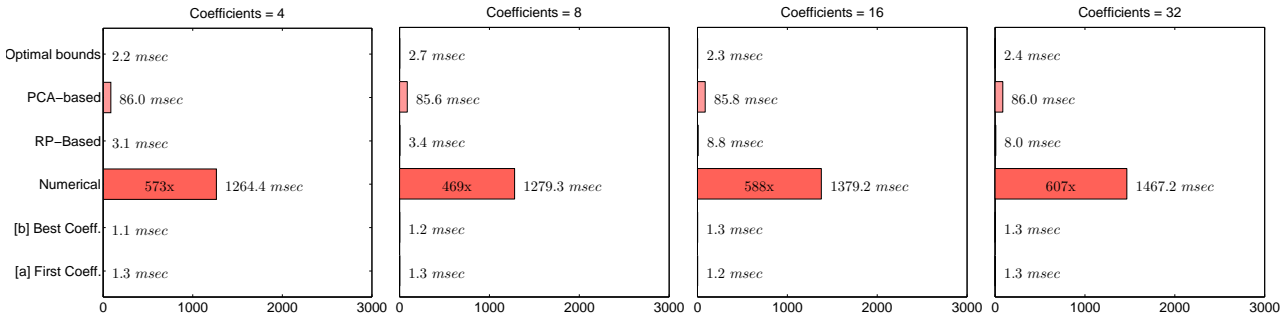


Fig. 9: Runtime of various techniques to compute a distance estimate for one pair of objects.

10.2 Mining in the compressed domain

We evaluate the quality of mining operations when operating directly on the compressed data. We compare with techniques based on PCA and Random Projections.

Which distance proxy?: While techniques based on PCA and RP provide only a single distance estimate in the compressed domain, our approach provides both a lower and an upper bound. Earlier, we explained that one can use one of three potential proxies for the distance: the upper bound u_b , the lower bound ℓ_b , or the average of the two. So, first, we evaluate which of the three metrics provides better distance estimation using a k -NN task on the weblog dataset. Figure 10 shows how much large a percentage of the common k -NN objects is returned in the compressed domain, versus those we would have gotten on the uncompressed data. The experiment is conducted under increasing number of coefficients. One can observe that the average of the lower and upper bounds shows overall superior performance, and this is the distance proxy we use for the remaining of the experiments.

k -NN performance: As mentioned, we perform comparisons under *fair* settings. Each object under our methodology is represented using $r = \lceil 2s + \frac{s}{2} + 1 \rceil$ double variables using s coefficients in the Fourier basis. To compress each object using RP-based or PCA-based k -NN, we project each sequence onto d dimensions such that the resulting low-dimensional projection point does not exceed the memory size of r double values. The random matrices Φ in the RP-based k -NN are *universally* near-isometric; i.e., with high probability, the same Φ matrix serves as a good linear map for *any* input signal, the creation of Φ 's is performed once offline for each case; thus, we assume that this operation requires $\mathcal{O}(1)$ time and space complexity.

Figure 11 displays the results, averaged over 100 queries. Naturally, the PCA-based k -NN approach returns the best results because it uses all the data to

construct the appropriate basis on which to project the data. However, it requires the computation of a Singular Value or Eigenvalue Decomposition, a $\mathcal{O}(dN)$ (using Krylov power methods [48]) and $\mathcal{O}(d^2N)$ time complexity operation in the most favorable and average scenario, respectively.

Sacrificing accuracy for low complexity, the RP-based approaches constitute inexpensive solutions: constructing matrix Φ is easy in practice, while binary-based matrix ensembles, such as BRP or ARP matrices, constitute low space-complexity alternatives. However, RP-based schemes are probabilistic in nature; they might “break down” for a fixed Φ : one can construct adversarial inputs where their performance degrades significantly.

Our methodology presents a balanced approach with low time and space complexity, while retaining high accuracy of results. Our approach exhibits better performance over all compression rates, compared to the RP-based approaches, see in Figure 11.

When sparsity is present: Using a similar methodology as in the previous section, we test the k -NN performance of the various approaches when sparsity of the signals is present. We create sparse representations of the weblog data by subsampling their Fourier representation. Figure 12 illustrates the quality of k -NN using such a sparsified dataset. Per $\Phi \in \mathbb{C}^{d \times N}$ matrix in the RP-based and the PCA-based k -NN, we project each weblog signal onto a d -dimensional (complex valued) space, where $d = \lceil s + \frac{s}{4} + \frac{1}{2} \rceil$.

Figure 12 reveals a notable behavior of our model: When $\mathbf{X}^{(i)}$ is sparse, each $\mathbf{X}^{(i)}$ can be more accurately represented (and thus compressed) using fewer coefficients. Thus, on average, the energy discarded e_x is also limited. Alternatively put, the constraint $\sum_{l \in p_x} |\mathbf{X}_l^{(i)}|^2 \leq e_x$ in (1) *highly restricts the candidate space that the uncompressed signal $\mathbf{x}^{(i)}$ resides in*, resulting in tighter upper and lower bounds. On the other hand, when compressing *dense* $\mathbf{X}^{(i)}$'s into s coefficients, where $s \ll N$, e_x provides a large amount of uncertainty as to the

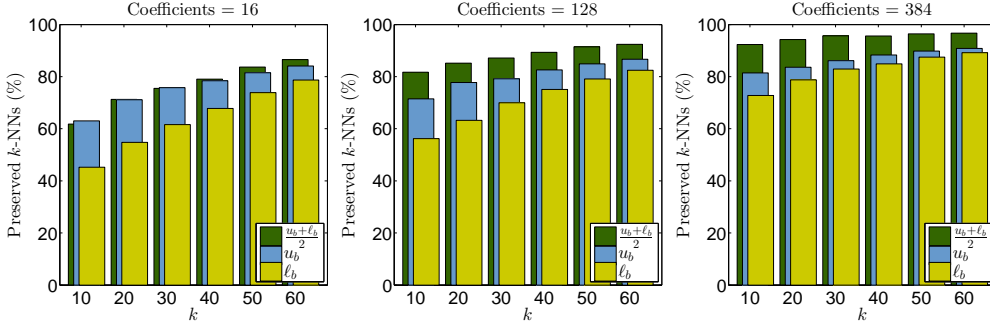


Fig. 10: Comparison of distance-estimation metrics for the k -NN task using the weblog data. The bars depict mean values of 100 Monte Carlo iterations.

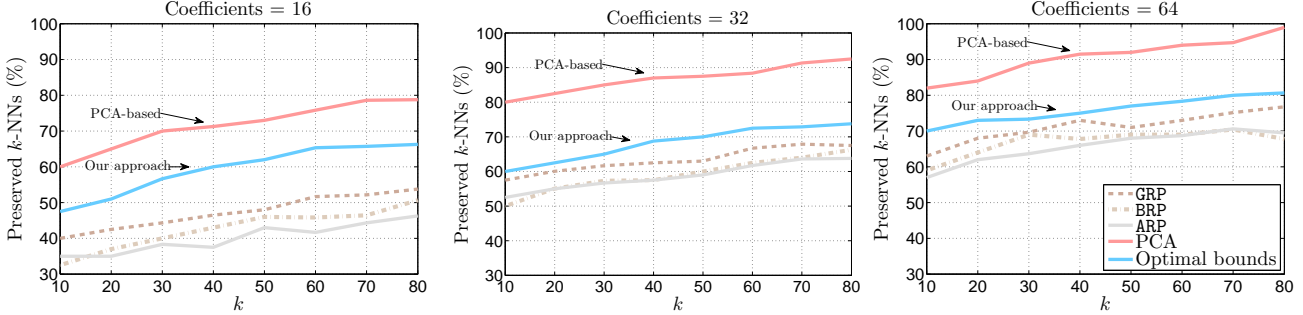


Fig. 11: Comparison of the algorithms under consideration for the k -NN task, where $\mathbf{X}^{(i)}/\mathbf{x}^{(i)}$ is originally dense. The curves depict mean values of 100 Monte Carlo iterations.

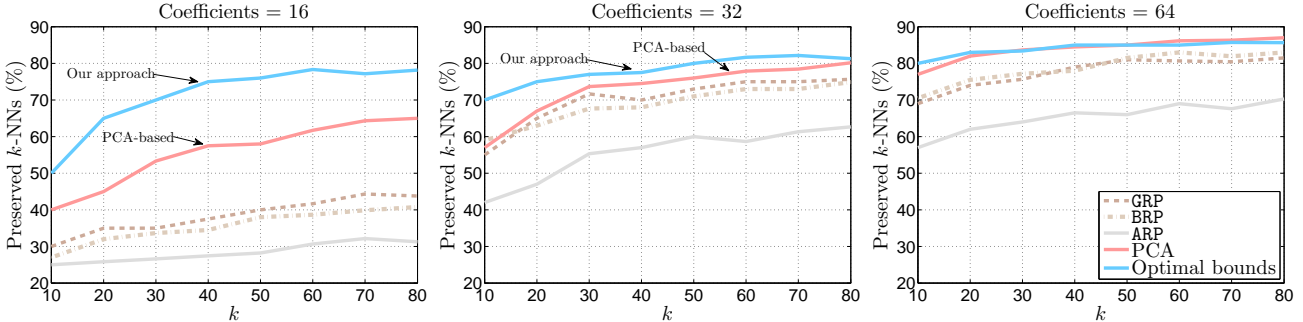


Fig. 12: Comparison of the algorithms under consideration for the k -NN task, where each weblog sequence is sparsified to contain only $3s$ coefficients, where $s = \{16, 32, 64\}$. The curves depict mean values of 100 Monte Carlo iterations.

reconstruction of $\mathbf{x}^{(i)}$. This leads to less tight distance bounds and thus degraded performance.

In summary, under high data-sparsity, our approach provides superior results in revealing the true k -NNs in the uncompressed domain. Our approach even outperforms PCA-based techniques, and more importantly, our method has a very low computational cost.

Clustering quality: We assess the quality of k -Means clustering operations in the compressed domain using the original weblog dataset. The quality is evaluated

in terms of how similar the clusters are before and after compression when k -Means is initialized using the same seed points. So, we use the same centroid points $C^{(t)}$ as in the uncompressed domain and then compress each $C^{(t)}$ accordingly, using the dimensionality reduction strategy dictated by each algorithm.

The quality results of the algorithms under comparison are depicted in Figure 13. We perform k -Means for different compression ratios (coefficients) and for different numbers of clusters k . The PCA-based approach returns the best performance, but in-

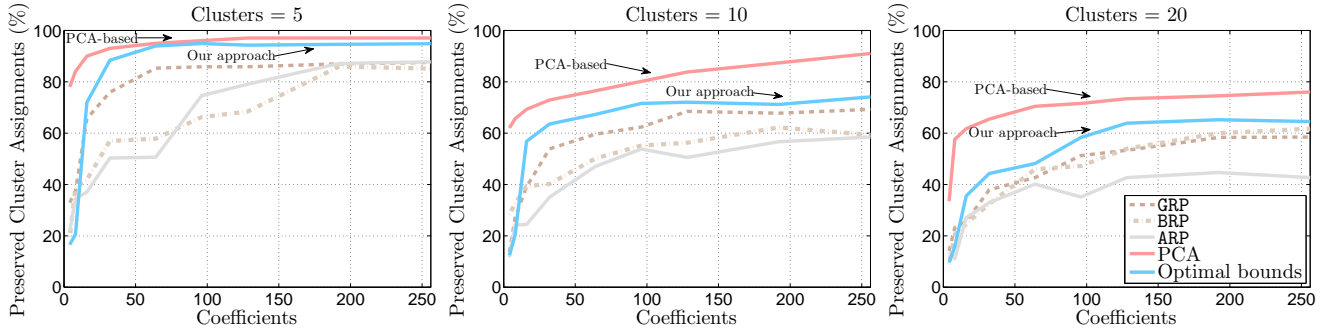


Fig. 13: Assessing the quality of k -Means clustering in the compressed domain.

roduces very large computational demands due to SVD/Eigenvalue computation. The performance of our methodology lies in-between PCA and Random-Projection techniques.

10.3 Using a different basis (wavelets)

In the preceding sections we used Fourier decomposition as our compression method. Now we use wavelets [49] to show the generality of our technique. We also use a very large image dataset consisting of VLSI patterns obtained by the semiconductor department of our organization.

In the remainder of this subsection we (i) provide an overview of the tasks and challenges related to the specific domain and, (ii) show the performance of k -NN search operations on this large real-world dataset.

Pattern detection on VLSIs: During the production of VLSI circuits (i.e., CPUs), a series of parameters and design protocols should be satisfied to ensure that the resulting product will not fail during the manufacturing process. For example, there are various *layouts configurations*, that have been known to cause short-circuits and jeopardize the reliability of a circuit. The absence of such artifacts guarantees that the circuit will connect as desired and ensures a margin of safety. A nonexhaustive list of parameters includes the width of metal wires, minimum distances between two adjacent objects, etc. We refer the reader to Figure 14 for some illustrations. Design-rule checking (DRC) is the process of checking the satisfiability of these rules.

As a result of a testing process, a collection of VLSI designs are annotated as faulty or functional. Now, each newly-produced circuit layout is classified as potentially-faulty based on its similarity to an already annotated circuit. Novel designs, never seen before, need to be tested further. Therefore, the testing process can be relegated to a k -NN search operation. The whole process needs to be both expedient and

accurate, so that design and production are properly streamlined.

Based on an algebra of polygons [51], a strategy to compare different layouts is by projecting their binary, 2D description onto one of the axes. This generates a descriptive vector per layout as the summary scanned from left to right or from top to bottom, see Figure 14. Thus, each VLSI layout can be approximately (but not uniquely) represented by a *signature* as the sum along rows or columns. Note that not only this format can be stored and processed more efficiently, but it also allows a translation-invariant matching of the shape. One can use both row-wise and column-wise signatures, but for the purposes of our experiments, we only use the column-wise signature.

VLSI dataset: Our original dataset consists of approximately 150,000 binary VLSI images, each of dimension 512×512 . We convert each image into a signature $\ell^{(i)} \in \mathbb{R}^{512}, \forall i$, as the column-sum of each image. Figure 15 depicts an instance of a layout image and its resulting signature.

Afterwards, we compress each signature using the wavelet transformation [49]:

$$\mathcal{L}^{(i)} = \text{WVL}(\ell^{(i)}) \in \mathbb{R}^{512}, \forall i,$$

where $\text{WVL}(\cdot)$ represents the wavelet linear operator with minimum scale $J_{\min} = 2$. Observe in the right part of Figure 15 that $\mathcal{L}^{(i)}$ is highly compressible: the energies of wavelet components decay rapidly to zero according to a power-law decay model of the form:

$$\left| \left(\mathcal{L}^{(i)} \right)_j \right| \leq R \cdot j^{-1/p}, \quad R > 0, \forall j, \quad (14)$$

for some r and p . This suggests that each signature is highly compressible with minimal loss in accuracy.

k -NN results on VLSI layouts: We evaluate the quality of the results of k -NN operations in the compressed domain as compared to the k -NN results when operating on the uncompressed image data. We limit our

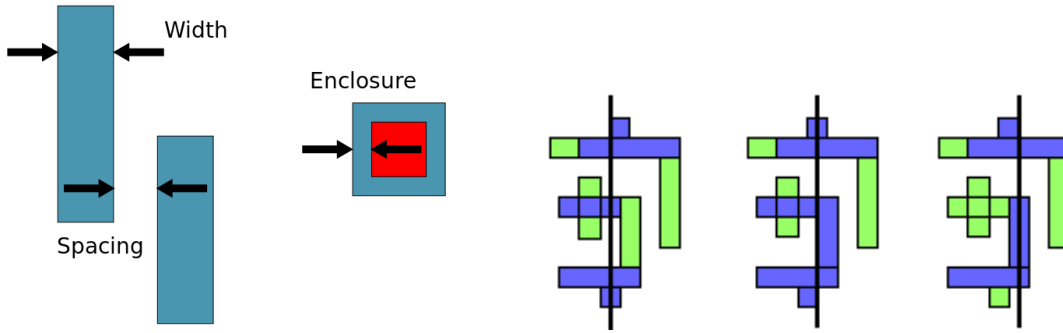


Fig. 14: **Left:** Subset of design parameters to be satisfied during the manufacturing [50]. Width and spacing are single layer rules, where the VLSI layout is seen as a 2D object. A width rule specifies the minimum width of objects; a spacing rule specifies the minimum distance between two adjacent objects. Enclosing deals with multi-layer rules is not considered here. **Right:** In the left-to-right scanning strategy, moving a vertical scan line horizontally across the layout, we can maintain the sum of polygons observed. As the scan line advances, new objects are added and old ones are removed.

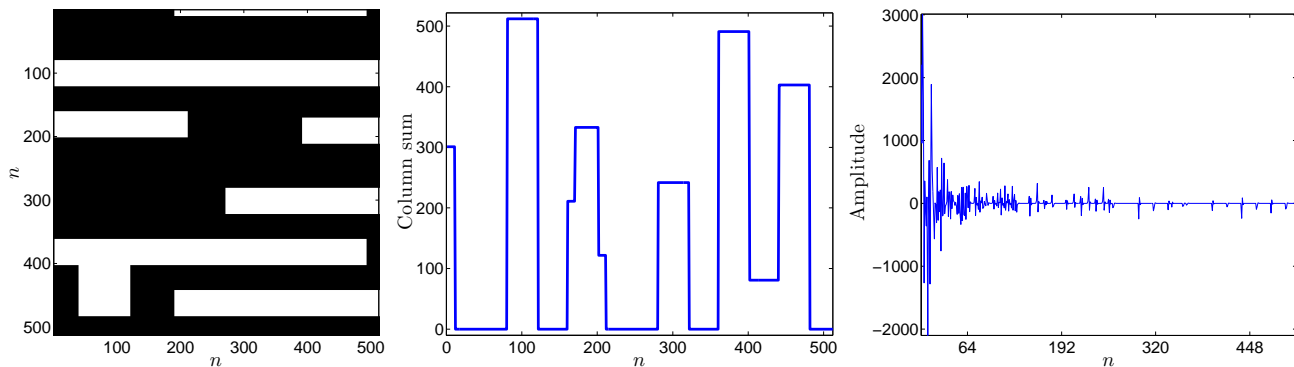


Fig. 15: **Left panel:** Original 2D layout example. White areas indicate the presence of objects, e.g., metal wires. **Center panel:** Column-sum representation of polygons in \mathbb{R}^{512} . **Right panel:** Amplitude of wavelet transformation on the column-sum representation: the majority of the energy can be captured in only a few coefficients.

evaluation to comparing our methodology to Random-Projection techniques, since only these approaches are scalable for large datasets.

Figure 16 illustrates the performance of the following approaches: (i) RP-based k -NN for three different random matrix ensembles (Gaussian, Bernoulli and Achlioptas’ based) and (ii) our Optimal Bounds approach. We observe that our method can improve the relative efficiency of matching by up to 20%, compared with the best random-projection approach. Finally, Figure 17 provides some representative examples of the $k = 5$ nearest neighbors for four randomly selected queries. Using the layout signature derived, we can discover very flexible translation-invariant matches.

Clustering quality on VLSI layouts: We assess the quality of k -Means clustering operations in the compressed domain. As before, the quality is evaluated as to how

similar the clusters are before and after compression when k -Means is initialized using the same seed points. So, we use the same centroid points $C^{(t)}$ as in the uncompressed domain and then compress each $C^{(t)}$ accordingly, using the dimensionality reduction strategy dictated by each algorithm. Again, we consider the k -Means algorithm as our baseline procedure.

Figure 18 depicts the results for three clustering levels k : 5, 10 and, 20 clusters. We perform k -Means for different compression ratios (coefficients) in the range $s = \{4, 8, 16, 32, 64, 128\}$. We evaluate how strong the distortion in clustering assignment is when operating in the compressed domain compared with the clustering on the original data. For all cases, our approach provides cluster output that aligns better with the original clustering. For this dataset we observe a consistent 5 – 10% improvement in the cluster quality returned. These trends are captured in Fig. 18.

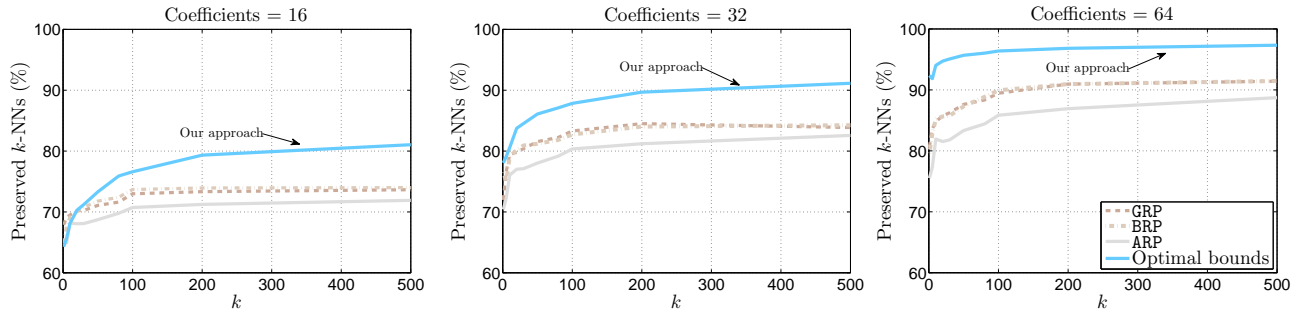


Fig. 16: k -NN preservation performance as function of k . Here, the cardinality of \mathcal{DB}_r is $|V| = 149,245$ and the byte size of each sequence is d bytes.

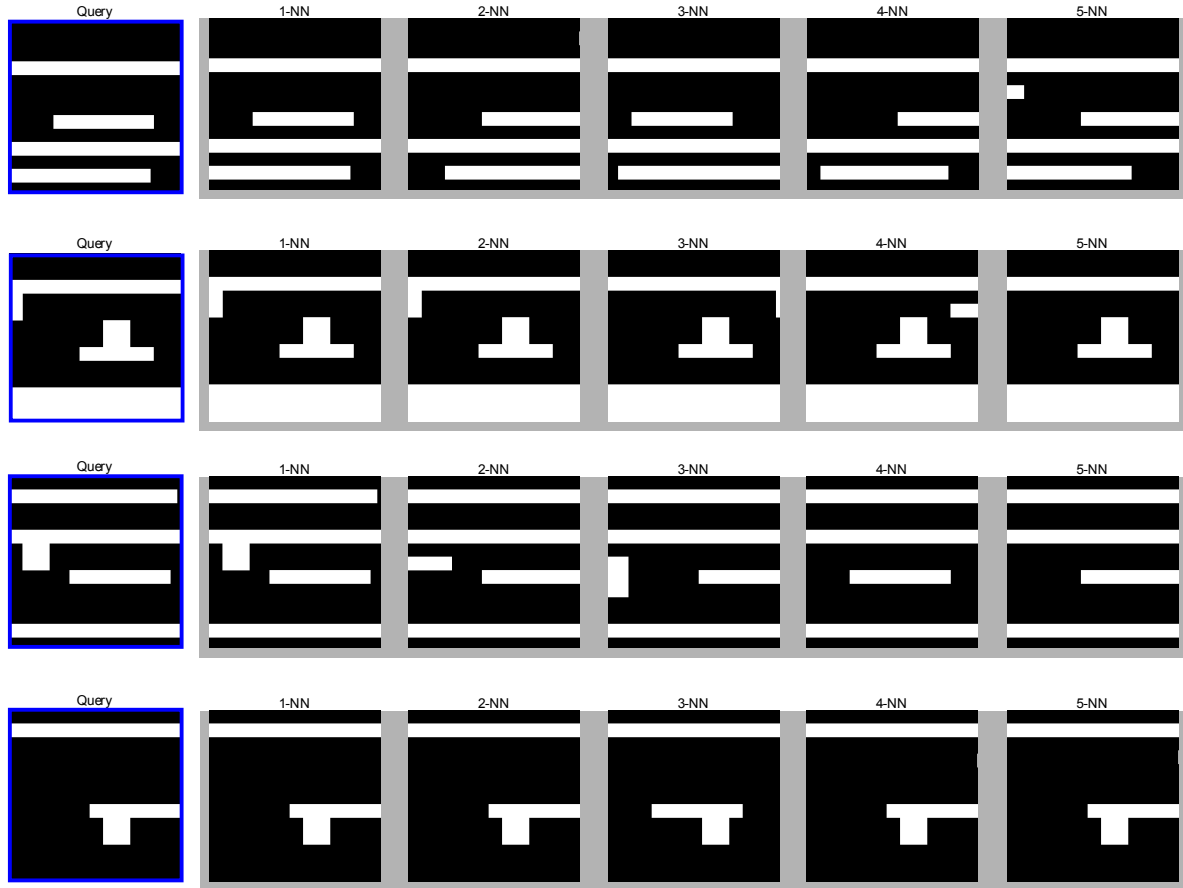


Fig. 17: Illustrative examples of the Optimal Bounds approach performance for the k -NN problem. The leftmost images are the query images, and on the right we depict the $k = 5$ nearest neighbors as computed in the compressed domain. Observe that owing to the signature extracted from each image, we can also detect translation-invariant matches.

In summary, the experiments have provided strong evidence that our methodology can offer better mining quality in the compressed domain than random projection techniques, both in the traditional and in the compressed-sensing sense (i.e., high data-sparsity). Finally, because our approach is also very fast to compute (e.g. compared with PCA), we believe that it will

provide an important building block for transitioning many mining operations into the compressed data space.

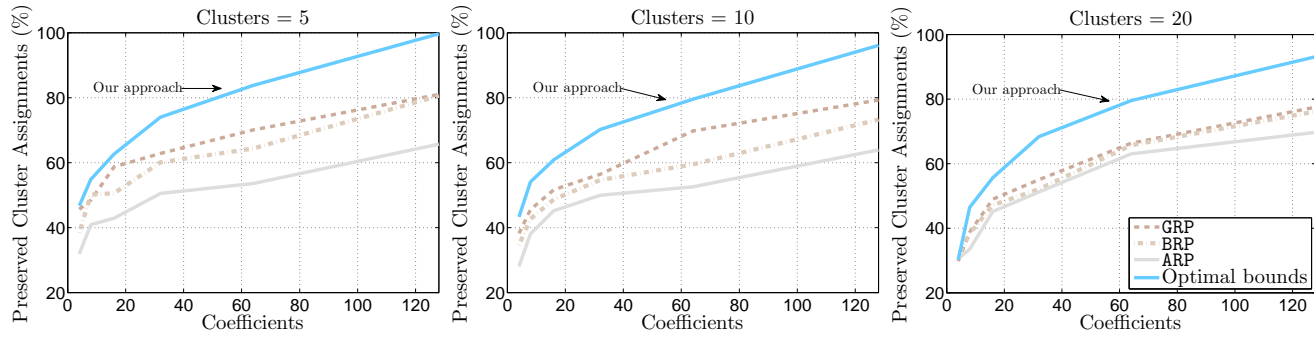


Fig. 18: Clustering preservation performance as function of the number of coefficients s . RP-based approaches project onto a d -dimensional subspace, where $d = \lceil 2s + \frac{s}{2} + 1 \rceil$. Here, the cardinality of the dataset is $|V| = 50,000$ and the byte size of each sequence is d bytes.

11 Conclusion

We have examined how to compute optimally-tight distance bounds on compressed data representations under any orthonormal transform. We have demonstrated that our methodology can retrieve more relevant matches than competing approaches during mining operations on the compressed data. A particularly interesting result is that for data with very high redundancy/sparsity (see, for example, magnetic resonance imaging [37]), our approach may even provide better search performance than compressed sensing approaches, which have been designed to specifically tackle the sparsity issue. In such scenarios, our method may even outperform PCA-based techniques, owing to its capability to use different sets of high-energy coefficients per object.

As future work, we intend to continue to investigate the merits of our methodology under a broader variety of distance-based operations such as anomaly detection and density-based clustering.

Acknowledgements: The research leading to these results has received funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 259569.

References

1. A. Souza and J. Pineda, “Tidal mixing modulation of sea surface temperature and diatom abundance in Southern California,” in *Continental Shelf Research*, 21(6-7), pp. 651–666, 2001.
2. P. Noble and M. Wheatland, “Modeling the sunspot number distribution with a fokker-planck equation,” *The Astrophysical Journal*, vol. 732, no. 1, p. 5, 2011.
3. G. Baechler, N. Freris, R. Quick, and R. Crochiere, “Finite rate of innovation based modeling and compression of ECG signals,” in *Proc. of ICASSP*, pp. 1252–1256, 2013.
4. S. Chien and N. Immorlica, “Semantic similarity between search engine queries using temporal correlation,” in *Proc. of WWW*, 2005.
5. B. Liu, R. Jones, and K. L. Klinkner, “Measuring the meaning in time series clustering of text search queries,” in *Proceedings of the 15th ACM international conference on Information and knowledge management*, pp. 836–837, ACM, 2006.
6. E. Nygren, R. K. Sitaraman, and J. Wein, “Networked systems research at akamai,” *ACM SIGOPS Operating Systems Review*, vol. 44, no. 3, pp. 1–1, 2010.
7. R. Agrawal, C. Faloutsos, and A. Swami, *Efficient similarity search in sequence databases*. Springer, 1993.
8. D. Rafiei and A. Mendelzon, “Efficient retrieval of similar time sequences using dft,” in *In Proc. FODO Conference, Kobe*, Citeseer, 1998.
9. F.-P. Chan, A.-C. Fu, and C. Yu, “Haar wavelets for efficient similarity search of time-series: with and without time warping,” *Knowledge and Data Engineering, IEEE Transactions on*, vol. 15, no. 3, pp. 686–705, 2003.
10. V. Eruhimov, V. Martyanov, P. Raulefs, and E. Tuv, “Combining unsupervised and supervised approaches to feature selection for multivariate signal compression,” in *Intelligent Data Engineering and Automated Learning*, pp. 480–487, 2006.
11. P. Y. M. Vlachos, S.S. Kozat, “Optimal distance bounds for fast search on compressed time-series query logs,” in *ACM Transactions on the Web*, 4(2), pp. 6:1–6:28, 2010.
12. M. Vlachos, S. Kozat, and P. Yu, “Optimal Distance Bounds on Time-Series Data,” in *Proc. of SIAM Data Mining (SDM)*, pp. 109–120, 2009.
13. Y. Cai and R. Ng, “Indexing spatio-temporal trajectories with chebyshev polynomials,” in *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pp. 599–610, ACM, 2004.
14. C. Wang and X. S. Wang, “Multilevel filtering for high dimensional nearest neighbor search,” in *In proceedings of ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge*, Citeseer, 2000.
15. S. Dasgupta, “Experiments with random projection,” in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 143–151, Morgan Kaufmann Publishers Inc., 2000.
16. R. Calderbank, S. Jafarpour, and R. Schapire, “Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain,” *Preprint*, 2009.
17. W. B. Johnson and J. Lindenstrauss, “Extensions of Lipschitz mappings into a Hilbert space,” *Contemporary mathematics*, vol. 26, pp. 189–206, 1984.

18. P. Indyk and A. Naor, "Nearest-neighbor-preserving embeddings," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 3, p. 31, 2007.
19. N. Ailon and B. Chazelle, "Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform," in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pp. 557–563, ACM, 2006.
20. C. Boutsidis, A. Zouzias, and P. Drineas, "Random Projections for k -means Clustering," in *Advances in Neural Information Processing Systems*, pp. 298–306, 2010.
21. A. Cardoso and A. Wichert, "Iterative random projections for high-dimensional data clustering," *Pattern Recognition Letters*, vol. 33, no. 13, pp. 1749–1755, 2012.
22. D. Achlioptas, "Database-friendly random projections," in *Proc. of ACM PODS*, pp. 274–281, 2001.
23. E. Bingham and H. Mannila, "Random projection in dimensionality reduction: applications to image and text data," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 245–250, ACM, 2001.
24. N. M. Freris, M. Vlachos, and S. S. Kozat, "Optimal distance estimation between compressed data series.," in *SDM*, pp. 343–354, 2012.
25. M. Vlachos, P. Yu, and V. Castelli, "On periodicity detection and structural peroidal similarity," in *SIAM International Conference on Data Mining*, pp. 449–460, 2005.
26. A. Mueen, S. Nath, and J. Liu, "Fast approximate correlation for massive time-series data," in *Proceedings of the 2010 ACM SIGMOD International Conference on Management of data*, pp. 171–182, ACM, 2010.
27. E. Keogh and S. Kasetty, "On the Need for Time Series Data Mining Benchmarks: A Survey and Empirical Demonstration," in *Proc. of KDD*, 2002.
28. A. Mueen, E. J. Keogh, and N. B. Shamlou, "Finding Time Series Motifs in Disk-Resident Data," in *Proc. of ICDM*, pp. 367–376, 2009.
29. A. V. Oppenheim, R. W. Schaffer, J. R. Buck, et al., *Discrete-time signal processing*, vol. 5. Prentice Hall Upper Saddle River, 1999.
30. E. Keogh, K. Chakrabarti, S. Mehrotra, and M. Pazzani, "Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases," in *Proc. of ACM SIGMOD*, pp. 151–162, 2001.
31. S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 1st ed., 2004.
32. M. Grant and S. Boyd, "Graph Implementations for Nonsmooth Convex Programs," *Recent Advances in Learning and Control*, Springer, pp. 95–110, 2008.
33. T. Basar and G. J. Olsder, *Dynamic Noncooperative Game Theory*. Academic Press, 2nd ed., 1995.
34. D. L. Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.
35. T. Cover and P. Hart, "Nearest neighbor pattern classification," *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 21–27, 1967.
36. P. W. Jones, A. Osipov, and V. Rokhlin, "Randomized approximate nearest neighbors algorithm," *Proceedings of the National Academy of Sciences*, vol. 108, no. 38, pp. 15679–15686, 2011.
37. E. J. Candes, J. K. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on pure and applied mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
38. E. Kushilevitz, R. Ostrovsky, and Y. Rabani, "Efficient search for approximate nearest neighbor in high dimensional spaces," *SIAM Journal on Computing*, vol. 30, no. 2, pp. 457–474, 2000.
39. C. Hegde, A. Sankaranarayanan, W. Yin, and R. Baraniuk, "A convex approach for learning near-isometric linear embeddings," *preprint, August*, 2012.
40. S. Dasgupta, "Learning mixtures of Gaussians," in *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pp. 634–644, IEEE, 1999.
41. R. I. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," in *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pp. 616–623, IEEE, 1999.
42. N. M. Freris, M. Vlachos, and D. S. Turaga, "Cluster-Aware Compression with Provable K-means Preservation," in *SDM*, pp. 82–93, 2012.
43. S. Lloyd, "Least squares quantization in PCM," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.
44. A. Tanay, R. Sharan, and R. Shamir, "Biclustering algorithms: A survey," *Handbook of computational molecular biology*, vol. 9, pp. 26–1, 2005.
45. I. S. Dhillon, "Co-clustering documents and words using bipartite spectral graph partitioning," in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 269–274, ACM, 2001.
46. P. J. Huber, "Projection pursuit," *The annals of Statistics*, No. 41, SIAM, 1985.
47. D. Arthur and S. Vassilvitskii, "k-Means++: The Advantages of Careful Seeding," in *Proc. of Symposium of Discrete Analysis*, 2005.
48. J. K. Cullum and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations: Volume 1, Theory*. No. 41, SIAM, 2002.
49. S. Mallat, *A wavelet tour of signal processing*. Elsevier, 1999.
50. Wikipedia, "http://en.wikipedia.org/wiki/Design_rule_checking."
51. B. Crawford, "Design rules checking for integrated circuits using graphical operators," in *Proceedings of the 2nd annual conference on Computer graphics and interactive techniques*, pp. 168–176, ACM, 1975.

12 Appendix

12.1 Existence of solutions and necessary and sufficient conditions for optimality

The constraint set is a compact convex set, in fact, a compact *polyhedron*. The function $g(x, y) := -\sqrt{x}\sqrt{y}$ is convex but not strictly convex on \mathbb{R}_+^2 . To see this, note that the *Hessian* exists for all $x, y > 0$ and equals

$$\nabla^2 g = \frac{1}{4} \begin{pmatrix} x^{-\frac{3}{2}}y^{-\frac{1}{2}} & -x^{-\frac{1}{2}}y^{-\frac{1}{2}} \\ -x^{-\frac{1}{2}}y^{-\frac{1}{2}} & x^{-\frac{1}{2}}y^{-\frac{3}{2}} \end{pmatrix}$$

with eigenvalues $0, \frac{1}{\sqrt{xy}}(\frac{1}{x} + \frac{1}{y})$, and hence is positive semi-definite, which in turn implies that g is convex [31]. Furthermore, $-\sqrt{x}$ is a strictly convex function of x so that the objective function of (3) is convex, and strictly convex only if $p_x^- \cap p_q^- = \emptyset$. It is also a continuous function so solutions exist, i.e., the optimal value is bounded and is attained. It is easy to check that the *Slater condition* holds, whence the problem satisfies *strong duality* and there exist Lagrange multipliers [31]. We skip the technical details for simplicity,

but we want to highlight that this property is crucial because it guarantees that the Karush–Kuhn–Tucker (KKT) necessary conditions [31] for Lagrangian optimality are also *sufficient*. Therefore, if we can find a solution that satisfies the KKT conditions for the problem, we have found an *exact* optimal solution and the *exact* optimal value of the problem. The Lagrangian is

$$\begin{aligned} L(\mathbf{y}, \mathbf{z}, \lambda, \mu, \alpha, \beta) := & -2 \sum_{i \in P_1} b_i \sqrt{z_i} - 2 \sum_{i \in P_2} a_i \sqrt{y_i} - 2 \sum_{i \in P_3} \sqrt{z_i} \sqrt{y_i} \\ & + \lambda \left(\sum_{i \in p_x^-} (z_i - e_x) \right) + \mu \left(\sum_{i \in p_q^-} (y_i - e_q) \right) \\ & + \sum_{i \in p_x^-} \alpha_i (z_i - Z) + \sum_{i \in p_q^-} \beta_i (y_i - Y). \end{aligned} \quad (\text{A-1})$$

The KKT conditions are as follows⁶:

$$0 \leq z_i \leq Z, \quad 0 \leq y_i \leq Y, \quad (\text{PF}) \quad (\text{A-2a})$$

$$\begin{aligned} \sum_{i \in p_x^-} z_i \leq e_x, \quad \sum_{i \in p_x^-} z_i \leq e_Q \\ \lambda, \mu, \alpha_i, \beta_i \geq 0 \quad (\text{DF}) \end{aligned} \quad (\text{A-2b})$$

$$\alpha_i (z_i - Z) = 0, \quad \beta_i (y_i - Y) = 0 \quad (\text{CS}) \quad (\text{A-2c})$$

$$\begin{aligned} \lambda \left(\sum_{i \in p_x^-} (z_i - e_x) \right) = 0, \quad \mu \left(\sum_{i \in p_q^-} (y_i - e_q) \right) = 0 \\ i \in P_1: \quad \frac{\partial L}{\partial z_i} = -\frac{b_i}{\sqrt{z_i}} + \lambda + \alpha_i = 0 \quad (\text{O}) \end{aligned} \quad (\text{A-2d})$$

$$i \in P_2: \quad \frac{\partial L}{\partial y_i} = -\frac{a_i}{\sqrt{y_i}} + \mu + \beta_i = 0$$

$$i \in P_3: \quad \frac{\partial L}{\partial z_i} = -\frac{\sqrt{y_i}}{\sqrt{z_i}} + \lambda + \alpha_i = 0$$

$$\frac{\partial L}{\partial y_i} = -\frac{\sqrt{z_i}}{\sqrt{y_i}} + \mu + \beta_i = 0,$$

where we use shorthand notation for *Primal Feasibility* (PF), *Dual Feasibility* (DF), *Complementary Slackness* (CS), and *Optimality* (O) [31].

12.2 Proof of theorem 1

For the first part, note that problem (3) is a double minimization problem over $\{z_i\}_{i \in p_x^-}$ and $\{y_i\}_{i \in p_q^-}$. If we fix one vector in the objective function of (3), then the optimal solution with respect to the other one is given by the waterfilling algorithm. In fact, if we consider the KKT conditions (A-2) or the KKT conditions to (2), they correspond exactly to (7). The waterfilling algorithm has the property that if $\mathbf{a} = \text{waterfill}(\mathbf{b}, e_x, A)$, then $b_i > 0$ implies $a_i > 0$. Furthermore, it has a monotonicity property in the sense that $b_i \leq b_j$ implies $a_i \leq a_j$. Assume that, at optimality, $a_{l_1} < a_{l_2}$ for some $l_1 \in P_1, l_2 \in P_3$. Because $b_{l_1} \geq B \geq b_{l_3}$ we can swap these two values to decrease the objective function, which is

a contradiction. The exact same argument applies for $\{b_l\}$, so $\min_{l \in P_1} a_l \geq \max_{l \in P_3} a_l, \min_{l \in P_2} b_l \geq \max_{l \in P_3} b_l$.

For the second part, note that $-\sum_{i \in P_3} \sqrt{z_i} \sqrt{y_i} \geq -\sqrt{e'_x} \sqrt{e'_q}$. If $e'_x e'_q > 0$, then at optimality this is attained with equality for the particular choice of $\{a_l, b_l\}_{l \in P_3}$. It follows that all entries of the optimal solution $\{a_l, b_l\}_{l \in p_x^- \cup p_q^-}$ are strictly positive, hence (A-2d) implies that

$$a_i = \frac{b_i}{\lambda + \alpha_i}, \quad i \in P_1 \quad (\text{A-3a})$$

$$b_i = \frac{a_i}{\mu + \beta_i}, \quad i \in P_2 \quad (\text{A-3b})$$

$$a_i = (\mu + \beta_i) b_i, \quad i \in P_3 \quad (\text{A-3c})$$

$$b_i = (\lambda + \alpha_i) a_i, \quad i \in P_3.$$

For the particular solution with all entries in P_3 equal $(a_l = \sqrt{e'_x / |P_3|}, b_l = \sqrt{e'_q / |P_3|})$, (8a) is an immediate application of (A-3.c). The optimal entries $\{a_l\}_{l \in P_1}, \{b_l\}_{l \in P_2}$ are provided by waterfilling with available energies $e_x - e'_x, e_q - e'_q$, respectively, so (9) immediately follows.

For the third part, note that the cases that either $e'_x = 0, e'_q > 0$ or $e'_x > 0, e'_q = 0$ are excluded at optimality by the first part, cf. (7).

For the last part, note that when $e'_x = e'_q = 0$, equivalently $a_l = b_l = 0$ for $l \in P_3$, it is not possible to take derivatives with respect to any coefficient in P_3 , so the last two equations of (A-2) do not hold. In that case, we need to perform a standard perturbation analysis. Let $\epsilon := \{\epsilon_l\}_{l \in P_1 \cup P_2}$ be a sufficiently small positive vector. As the constraint set of (3) is linear in z_i, y_i , any *feasible* direction (of potential decrease of the objective function) is of the form $z_i \leftarrow z_i - \epsilon_i, i \in P_1, y_i \leftarrow y_i - \epsilon_i, i \in P_2$, and $z_i, y_i \geq 0, i \in P_3$ such that $\sum_{i \in P_3} z_i = \sum_{i \in P_1} \epsilon_i, \sum_{i \in P_3} y_i = \sum_{i \in P_2} \epsilon_i$. The change in the objective function is then equal to (modulo an $o(\|\epsilon\|^2)$ term)

$$\begin{aligned} g(\epsilon) & \approx \frac{1}{2} \sum_{i \in P_1} \frac{b_i}{\sqrt{z_i}} \epsilon_i + \frac{1}{2} \sum_{i \in P_2} \frac{a_i}{\sqrt{y_i}} \epsilon_i - \sum_{i \in P_3} \sqrt{z_i} \sqrt{y_i} \\ & \geq \frac{1}{2} \sum_{i \in P_1} \frac{b_i}{\sqrt{z_i}} \epsilon_i + \frac{1}{2} \sum_{i \in P_2} \frac{a_i}{\sqrt{y_i}} \epsilon_i - \sqrt{\sum_{i \in P_1} \epsilon_i} \sqrt{\sum_{i \in P_2} \epsilon_i} \\ & \geq \frac{1}{2} \min_{i \in P_1} \frac{b_i}{\sqrt{z_i}} \epsilon_1 + \frac{1}{2} \min_{i \in P_2} \frac{a_i}{\sqrt{y_i}} \epsilon_2 - \sqrt{\epsilon_1 \epsilon_2}, \end{aligned} \quad (\text{A-4})$$

where the first inequality follows from an application of the Cauchy–Schwartz inequality to the last term, and in the second one we have defined $\epsilon_j = \sum_{i \in P_j} \epsilon_i, j = 1, 2$. Let us define $\epsilon := \sqrt{\epsilon_1 / \epsilon_2}$. From the last expres-

⁶ The condition (A-2d) excludes the cases that for some i $z_i = 0$, or $y_i = 0$, which will be treated separately.

sion, it suffices to test for any $i \in P_1, j \in P_2$:

$$g(\epsilon_1, \epsilon_2) = \frac{1}{2} \frac{b_i}{\sqrt{z_i}} \epsilon_1 + \frac{1}{2} \frac{a_j}{\sqrt{y_j}} \epsilon_2 - \sqrt{\epsilon_1} \sqrt{\epsilon_2} = \frac{1}{2} \sqrt{\epsilon_1} \sqrt{\epsilon_2} g_1(\epsilon) \quad (\text{A-5})$$

$$g_1(\epsilon) := \frac{b_i}{\sqrt{z_i}} \epsilon + \frac{a_j}{\sqrt{y_j}} \frac{1}{\epsilon} - 2 \geq \frac{1}{\epsilon} g_2(\epsilon)$$

$$g_2(\epsilon) := \frac{b_i}{A} \epsilon^2 - 2\epsilon + \frac{a_j}{B},$$

where the inequality above follows from the fact that $\sqrt{z_i} \leq A, i \in P_1$ and $\sqrt{y_j} \leq B, j \in P_2$. Note that $h(\epsilon)$ is a quadratic with a nonpositive discriminant $\Delta := 4(1 - \frac{a_i b_i}{AB}) \leq 0$ as, by definition, we have that $B \leq b_i, i \in P_1$ and $A \leq a_j, j \in P_2$. Therefore $g(\epsilon_1, \epsilon_2) \geq 0$ for any (ϵ_1, ϵ_2) both positive and sufficiently small, which is a necessary condition for local optimality. By convexity, the vector pair (\mathbf{a}, \mathbf{b}) obtained constitutes an optimal solution. \blacksquare

12.3 Energy allocation in double-waterfilling

Calculating a fixed point of T is of interest only if $e'_x, e'_q > 0$ at optimality. We know that we are not in the setup of Theorem 1.4, therefore we have the additional property that either $e_x > |P_1|A^2, e_q > |P_2|B^2$ or both. Let us define

$$\gamma_a := \inf \left\{ \gamma > 0 : \sum_{l \in P_2} \min \left(a_l^2 \frac{1}{\gamma}, B^2 \right) \leq e_q \right\} \quad (\text{A-6})$$

$$\gamma_b := \sup \left\{ \gamma \geq 0 : \sum_{l \in P_1} \min \left(b_l^2 \gamma, A^2 \right) \leq e_x \right\}.$$

Clearly if $e_x > |P_1|A^2$ then $\gamma_b = +\infty$, and for any $\gamma \geq \max_{l \in P_1} \frac{A^2}{b_l^2}$ we have $\sum_{l \in P_1} \min(b_l^2 \gamma, A^2) = |P_1|A^2$. Similarly, if $e_q > |P_2|B^2$ then $\gamma_a = 0$, and for any $\gamma \leq \min_{l \in P_2} \frac{a_l^2}{B^2}$ we have $\sum_{l \in P_2} \min(a_l^2 \frac{1}{\gamma}, B^2) = |P_2|B^2$. If $\gamma_b < +\infty$, we can find the exact value of γ_b analytically by sorting $\{\gamma_l^{(b)} := \frac{A^2}{b_l^2}\}_{l \in P_1}$, -i.e., by sorting $\{b_l^2\}_{P_1}$ in decreasing order—and considering

$$h_b(\gamma) := \sum_{l \in P_1} \min(b_l^2 \gamma_i^{(b)}, A^2) - e_x$$

and $v_i := h_b(\gamma_i^{(b)})$. In this case, $v_1 < \dots < v_{|P_1|}$, and $v_{|P_1|} > 0$, and there are two possibilities: 1) $v_1 > 0$ whence $\gamma_b < \gamma_1^{(b)}$, or 2) there exists some i such that $v_i < 0 < v_{i+1}$ whence $\gamma_i^{(b)} < \gamma_b < \gamma_{i+1}^{(b)}$. For both ranges of γ , the function h becomes *linear* and strictly increasing, and it is elementary to compute its root γ_b . A similar argument applies for calculating γ_a if γ_a is strictly positive, by defining

$$h_a := \sum_{l \in P_2} \min \left(a_l^2 \frac{1}{\gamma}, B^2 \right) - e_q$$

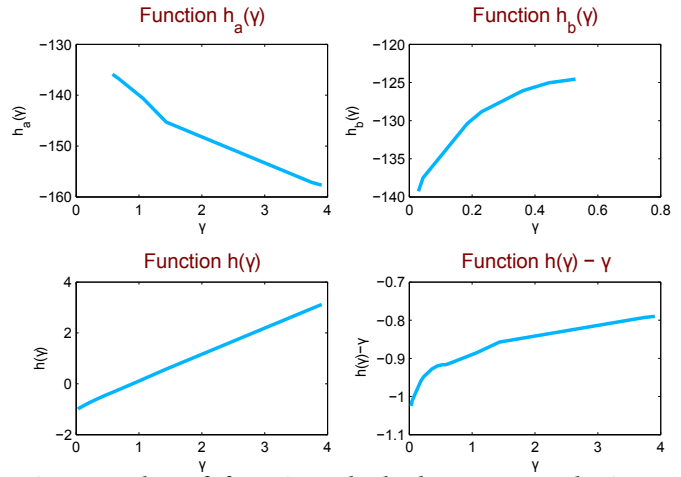


Fig. 19: Plot of functions h_a, h_b, h . Top row: h_a is a bounded decreasing function, which is piecewise linear in $\frac{1}{\gamma}$ with nonincreasing slope in $\frac{1}{\gamma}$; h_b is a bounded increasing piecewise linear function of γ with nonincreasing slope. Bottom row: h is an increasing function; the linear term γ dominates the fraction term, which is also increasing, see bottom right.

Theorem 2 (Exact solution of (9))

If either $e_x > |P_1|A^2, e_q > |P_2|B^2$ or both, then the nonlinear mapping T has a unique fixed point (e'_x, e'_q) with $e'_x, e'_q > 0$. The equation

$$\frac{e_x - \sum_{l \in P_1} \min(b_l^2 \gamma, A^2)}{e_q - \sum_{l \in P_2} \min(a_l^2 \frac{1}{\gamma}, B^2)} = \gamma \quad (\text{A-7})$$

has a unique solution $\bar{\gamma}$ with $\gamma_a \leq \bar{\gamma}$ and $\gamma_a \leq \gamma_b$ when $\gamma_b < +\infty$. The unique fixed point of T (solution of (9)) satisfies

$$e'_x = e_x - \sum_{l \in P_1} \min(b_l^2 \bar{\gamma}, A^2) \quad (\text{A-8})$$

$$e'_q = e_q - \sum_{l \in P_2} \min \left(a_l^2 \frac{1}{\bar{\gamma}}, B^2 \right).$$

Proof Existence⁷ of a fixed point is guaranteed by existence of solutions and Lagrange multipliers for (3), as by assumption we are in the setup of Theorem 1.2. Define $\gamma := \frac{e'_x}{e'_q}$; a fixed point $(e'_x, e'_q) = T((e'_x, e'_q))$, $e'_x, e'_q > 0$, corresponds to a root of

$$h(\gamma) := -\frac{e_x - \sum_{l \in P_1} \min(b_l^2 \gamma, A^2)}{e_q - \sum_{l \in P_2} \min(a_l^2 \frac{1}{\gamma}, B^2)} + \gamma \quad (\text{A-9})$$

⁷ An alternative and more direct approach of establishing the existence of a fixed point is by considering all possible cases and defining an appropriate compact convex set $E \subset \mathbb{R}_+^2 \setminus (0, 0)$ so that $T(E) \subset E$, whence existence follows by the Brouwer's fixed point theorem [33], as T is continuous.

For the range $\gamma \geq \gamma_a$ and $\gamma \leq \gamma_b$, if $\gamma_b < +\infty$, we have that $h(\gamma)$ is continuous and strictly increasing. The fact that $\lim_{\gamma \searrow \gamma_a} h(\gamma) < 0$, $\lim_{\gamma \nearrow \gamma_b} h(\gamma) > 0$ shows the existence of a unique root $\bar{\gamma}$ of h corresponding to a unique fixed point of T , cf. (A-8). ■

Remark 5 (Exact calculation of a root of h) We seek to calculate the root of h exactly and efficiently. In doing so, consider the points $\{\gamma_l\}_{l \in P_1 \cup P_2}$, where $\gamma_l := \frac{A}{b_l^2}$, $l \in$

P_1 , $\gamma_l := \frac{a_l^2}{B}$, $l \in P_2$. Then, note that for any $\gamma \geq \gamma_l, l \in P_1$ we have that $\min(b_l^2 \gamma, A^2) = A^2$. Similarly, for any $\gamma \leq \gamma_l, l \in P_2$, we have that $\min(a_l^2 \frac{1}{\gamma}, B^2) = B^2$. We order all such points in increasing order, and consider the resulting vector $\gamma' := \{\gamma'_i\}$ excluding any points below γ_a or above γ_b . Let us define $h_i := h(\gamma'_i)$. If for some i , $h_i = 0$ we are done. Otherwise there are three possibilities: 1) there is an i such that $h_i < 0 < h_{i+1}$; 2) $h_1 > 0$, or 3) $h_N < 0$. In all cases, the numerator (denominator) of h is linear in γ ($\frac{1}{\gamma}$) for the respective range of γ . Therefore, $\bar{\gamma}$ is obtained by solving the linear equation

$$f(\gamma) := e_x - \sum_{l \in P_1} \min(b_l^2 \gamma, A^2) - \gamma \left(e_q - \sum_{l \in P_2} \min\left(a_l^2 \frac{1}{\gamma}, B^2\right) \right). \quad (\text{A-10})$$

Note that there is no need for further computation to set this into the form $f(\gamma) = \alpha \gamma + \beta$ for some α, β . Instead, we use the elementary property that a linear function f on $[x_0, x_1]$ with $f(x_0)f(x_1) < 0$ has a unique root given by

$$\bar{x} = x_0 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_0) .$$