Cluster-Aware Compression with Provable K-means Preservation

Nikolaos M. Freris* Michail Vlachos* Deepak S. Turaga †

Abstract

This work rigorously explores the design of cluster-preserving compression schemes for high-dimensional data. We focus on the K-means algorithm and identify conditions under which running the algorithm on the compressed data yields the same clustering outcome as on the original. The compression is performed using single and multi-bit minimum mean square error quantization schemes as well as a given clustering assignment of the original data. We provide theoretical guarantees on post-quantization cluster preservation under certain conditions on the cluster structure, and propose an additional data transformation that can ensure cluster preservation unconditionally; this transformation is invertible and thus induces virtually no distortion on the compressed data. In addition, we provide an efficient scheme for multi-bit allocation, per cluster and data dimension, which enables a trade-off between high compression efficiency and low data distortion. Our experimental studies highlight that the suggested scheme accurately preserved the clusters formed in all cases, while incurring minimal distortion on the data shapes.

Our results can find many applications, e.g., in a) clustering, analysis and distribution of massive datasets, where the proposed data compression can boost performance while providing provable guarantees on the clustering result, as well as, in b) cloud computing services, as the optional transformation provides a data-hiding functionality in addition to preserving the K-means clustering outcome.

Keywords: Clustering, K-means, Compression, MMSE quantization, Cluster preservation

1 Introduction

Data clustering is one of the most important operations in the areas of data mining, machine learning and business analytics. It is also one of the most computationally challenging tasks, compounded also by the exponentially increasing dataset sizes.

This work explores methods for cluster-aware data compression. Specifically, we seek to address the following question: Can we design quantization-based compression schemes that yield in the same clustering outcome before and after compression?

The focus of our analysis is on the K-means algorithm, because of its widespread use in a variety of settings and applications ranging from image segmentation [1] to co-clustering [2], and even analysis of biological datasets [3]. Recently, many alternative clustering algorithms with more desirable stability properties (e.g., spectral methods [4]) have been derived. However, K-means is a widely used approach because of its simplicity of implementation, amenity to parallelization and speed of execution.

This work initiates a formal study for determining when the outcome of K-means is preserved by compression / data-simplification methods. We show, both analytically and experimentally, that using 1-bit Minimum Mean Square Error (MMSE) quantizers per dimension and cluster is sufficient to preserve the clustering outcome, provided that the clusters are 'well-separated.' When the latter does not hold, we devise an invertible data transformation which can *always* assure preservation of the clustering outcome. Moreover, we consider multi-bit quantization schemes that provide better balance between data compression and data reconstruction, while also ensuring cluster preservation. To that regard, we provide an efficient greedy algorithm for bit allocation that minimizes the mean squared compression error given storage constraints.

Applications and related work: Our work provides analytical results with important insights for practical applications in data mining such as:

a) Reducing space/time complexity of cluster operations based on K-means. This comes as a direct outcome of our analysis, because the proposed methodology provides guarantees for undistorted clustering results even when operating on the compressed domain. In addition, storing the quantized dataset requires less space than storing the original one. The level of compression is tunable based on the number of bits allocated to the scalar quantizers. Other approaches that investigate scaling-up the K-means algorithm include [5, 6]; they examine the problem either from a dimensionality reduction or from a sampling perspective, whereas our work views the problem from a *quantizer design* angle. In addition, those techniques make no assertions with regards to cluster preservation.

b) Enabling data-hiding. The proposed scheme provides an explicit *encoding* of the original dataset via lossy (non-invertible) data compression based on quantization, followed by a lossless (invertible) data trans-

*Nikolaos M. Freris and Michail Vlachos are with the Department of Mathematical and Computational Sciences, IBM-Research Zürich, Säumerstrasse 4, 8803 Rüschlikon, Switzerland.

†Deepak S. Turaga is with IBM T.J. Watson Research Center, Hawthorne, NY, USA. E-mail: turaga@us.ibm.com.

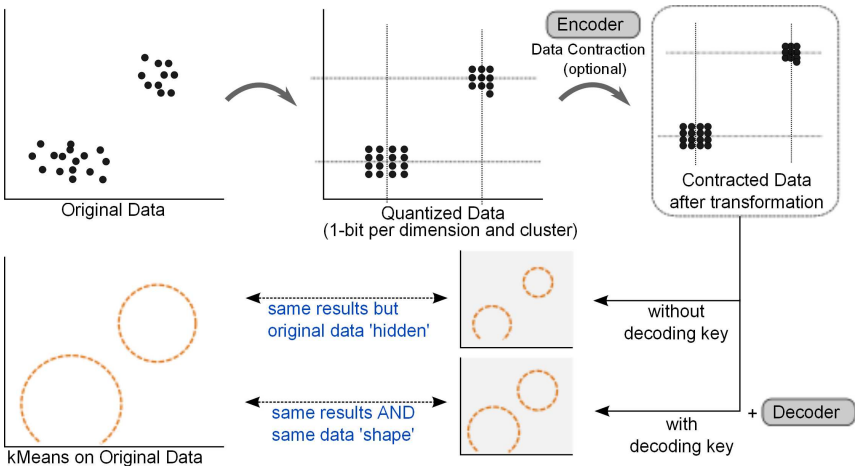

Figure 1: Quantization scheme for K-means cluster preservation. Our main contributions are: a) cluster-aware MMSE quantization, and b) an optional data-contraction component with provable cluster preservation guarantees and additional data-hiding

formation based on cluster contraction, or vice versa, because we prove that these two operations are interchangeable. The contractive transformation can be encoded in a scalar key which is then only revealed to appropriate recipients, who will consequently be in the position to revert back the data to their untransformed quantized state. While we make no additional provisions on the security aspect of such a protocol, our analytical derivations are novel and suggest a clear path for supporting such functionality. Given this aspect, our work contributes to the area of privacy-enabling data mining. There exist privacy-preserving variations for K-means that consider the scenario when data are segregated either vertically [7] or horizontally [8]. In contrast, using our approach the data are not separated but distributed as a whole. Similar in spirit to our work are also the works of Parmeswaran and Blough [9], who presented cluster preservation techniques through Nearest Neighbor (NN) data substitution, and of Oliveira and Zaane [10], who proposed rotation-based transformations (RBT) that retain the clustering outcome by changing the object values while maintaining the pairwise object distances. Our approach has the additional advantages of reduced storage requirements in addition to guaranteed preservation and minimal distortion of the original data structure.

c) Supporting high-quality data reconstruction. By providing tunable preservation of the underlying structural properties of the original data in the compressed domain, the compressed data can also be used for a variety of other mining and visualization applications. In this respect, our work overlaps with various simplification techniques. Bagnall *et al.* [11] proposed a *binary clipping* method for time series data, where the data are converted into 0 and 1 if they lie above or below the mean value baseline; this approach has been applied to speed up the execution of the K-means algorithm. Relevant is also the work of Aßfalg *et al.* [12] who proposed threshold-based representations for querying and indexing time series data. Megalooikonomou *et al.* [13] presented a piecewise vector-quantization approximation for time series data that preserves the shape of the original sequences with high accuracy. Finally, approaches such as wavelet or Fourier approximations have been used extensively for time series simplification; however none of these approaches are inherently designed for providing guarantees on the clustering outcome, which constitutes the key contribution of our work.

In [14], the authors proposed using 1-bit Moment Preserving Quantization (MPQ) per cluster and dimension, and showed experimentally that clusters are accurately preserved in some cases. The current work goes well beyond that by providing a rigorous analysis and theoretical guarantees: by using MMSE quantization together with a contractive data transformation, we can provide provable guarantees on cluster preservation. In addition, this work also introduces the notion of multi-bit quantization to support a fine-grained trade-off between compression and shape preservation.

An overview of the proposed methodology is summarized in Figure 1. The remainder of this paper is organized as follows: we briefly review the K-means optimization problem in Section 2, and describe our objectives in Section 3. In Section 4, we introduce the 1-bit MMSE quantization scheme and study its properties. We describe one-bit and multiple-bit MMSE com-

pression with K-means preservation in Section 5, and present results on real datasets in Section 6. We discuss the outcome of our work and potential extensions of our results in Section 7. We conclude the paper and propose future work in Section 8.

2 K-means Clustering

Consider a set $\mathcal{S}$ of N vectors $\mathbf{x}_j$ $(1 \leq j \leq N)$, each containing T dimensions x_{ji} $(1 \leq i \leq T)$. K-means clustering involves partitioning the N vectors into K clusters, i.e., into N disjoint subsets S_k $(1 \leq k \leq K)$, with $\cup_k S_k = \mathcal{S}$, such that the sum of intra-class variances

$$V := \sum_{k=1}^{K} \sum_{\mathbf{x}_j \in S_k} ||\mathbf{x}_j - \boldsymbol{\mu}_k||^2, \tag{2.1}$$

is minimized, where $\boldsymbol{\mu}_k := \frac{1}{|S_k|} \sum_{\mathbf{x}_j \in S_k} \mathbf{x}_j$ is the *centroid* of the k-th cluster, and $||\cdot||$ represents the standard Euclidean (L_2) norm on $\mathbb{R}^T$. The objective function can be expanded in terms of the individual dimensions x_{ji} of each datapoint $\mathbf{x}_j$ as

$$V = \sum_{k=1}^{K} \sum_{i=1}^{T} \sum_{\mathbf{x}_j \in S_k} (x_{ji} - \mu_{ki})^2 \tag{2.2}$$

$$= \sum_{k=1}^{K} \sum_{i=1}^{T} [\sum_{\mathbf{x}_j \in S_k} x_{ji}^2 - \frac{1}{|S_k|} \big(\sum_{\mathbf{x}_j \in S_k} x_{ji} \big)^2], \tag{2.3}$$

where we have used the definition of $\boldsymbol{\mu}_k$.

Therefore, the objective function depends on the first $(\sum x_{ji})$ and the second data sample moment $(\sum x_{ji}^2)$ per cluster, as well as on the object to cluster assignment.

3 Problem Description

Our goal is to design a quantization scheme that retains the K-means clustering after quantization, i.e., one that guarantees that using the K-means algorithm to cluster the quantized data results in exactly the same clustering assignment as for the original data. In order to achieve this objective, we use the above derivation of the K-Means to drive the design of our compression scheme. We prove that when using 1-bit MMSE data quantizers, per cluster and dimension the following hold:

- The first moment is preserved for every cluster.
- The second moment is reduced for every cluster.
- The optimal cluster assignment does not change for "well-behaved" clusters (to be defined later).

Therefore the above ensure the clustering on the simplified dataset will result in identical clusters as on the original dataset, under the mentioned conditions. We tackle the case of non well-behaved clusters and show that an additional linear transformation will result in unconditional cluster preservation .

4 MMSE Quantization

Quantization schemes have been widely used for compressing data in image and video processing [15]. Quantization is a form of *lossy* compression. Therefore typically several quantization levels are needed to preserve the quality and usability of the original objects. Here we show that a proper single threshold (1-bit) quantization is sufficient to retain the K-means outcome.

We consider MMSE quantization, in which the threshold level is set equal to the mean value and each data sample can be represented by a 0 or 1, indicating whether it lies above or below the threshold value, respectively. Formally, let us consider a dataset of scalar values $X = \{x_i\}_{i=1}^N$ with sample mean μ and sample variance σ^2. Let a, b denote the lower and upper quantization values, respectively, whereas the quantization threshold is set equal to μ. Selecting a, b so as to minimize the Mean Square Error (MSE) due to quantization amounts to solving the following optimization problem:

$$\min_{a \leq \mu \leq b} C(a,b) := \frac{1}{N} [\sum_{x_i < \mu} (x_i - a)^2 + \sum_{x_i \geq \mu} (x_i - b)^2]. \tag{4.4}$$

The solution is given by

$$\hat{x}_i = \begin{cases} \frac{1}{N_g} \sum\limits_{x_j \geq \mu} x_j, & x_i \geq \mu \\ \frac{1}{N - N_g} \sum\limits_{x_j < \mu} x_j, & x_i < \mu \end{cases} \tag{4.5}$$

where N_g denotes the number of datapoints with values greater than or equal to μ. We will use one such quantizer *per dimension* and *per cluster*, i.e., to say we quantize each dimension of all data belonging to the same cluster using a separate 1-bit MMSE quantizer.

Let us define the upper and lower dataset *extent* values $d_{\max} := \max_j(x_j - \mu)$ and $d_{\min} := -\min_j(x_j - \mu)$. Similarly, for the quantized data denoted by $\hat{X} = \{\hat{x}_i\}_{i=1}^N$ let $\hat{d}_{\max} := \max_j(\hat{x}_j - \hat{\mu})$ and $\hat{d}_{\min} := -\min_j(\hat{x}_j - \hat{\mu})$. For cluster preservation, it is desirable that clusters "shrink" after quantization, in the sense that the extent in each direction (below or above the mean) is not increased. Formally, we require that

$$\hat{d}_{\max} \leq d_{\max} \quad \text{and} \quad \hat{d}_{\min} \leq d_{\min}. \tag{4.6}$$

We prove that this always holds true for MMSE quantization in Lemma 4.1. We illustrate the property

of dynamic range reduction for MMSE quantization in Figure 2. In contrast, the MPQ used in [14] for cluster preservation may violate this property when the cluster is "ill-behaved," e.g., when it contains a small number of points that are far from the mean in one direction of the mean, combined with a large number of points in the other direction (see Figure 2.b). For "well-behaved" clusters (see Figure 2.a), MPQ also satisfies property (4.6), cf. [14].

LEMMA 4.1. (PROPERTIES OF 1-BIT MMSE QUANTIZER *For the 1-bit MMSE quantizer the following holds true:*

1. *The mean ($\hat{\mu}$) and variance ($\hat{\sigma}^2$) of the quantized dataset $\{\hat{x}_i\}_{i=1}^N$ satisfy*

$$\hat{\mu} = \mu \quad \hat{\sigma}^2 \leq \sigma^2, \tag{4.7}$$

where the inequality is strict if and only if the dataset contains more than two distinct points.

2. *The Mean Square Error (MSE) due to quantization satisfies*

$$MSE \leq \sigma^2, \tag{4.8}$$

where the inequality is strict if and only if $\sigma^2 > 0$.

3. *The extent of the quantized dataset is not increased in each direction, i.e.,*

$$\hat{d}_{\max} \leq d_{\max} \quad \textit{and} \quad \hat{d}_{\min} \leq d_{\min}, \tag{4.9}$$

where each inequality is strict if there are more than one distinct samples in the original dataset above or below the mean μ, respectively.

4. *The 1-bit MMSE quantization scheme is a quasi-linear[1] operation on the dataset $\{x_i\}_{i=1}^N$, in the sense that for any $a, b \in \mathbb{R}$:*

$$\widehat{ax_i} = a\hat{x}_i, \tag{4.10}$$

$$\widehat{x_i + b} = \hat{x}_i + b, \tag{4.11}$$

$$[\mu + \widehat{a(x_i - \mu)}] = \mu + a(\hat{x}_i - \mu). \tag{4.12}$$

Proof. The fact that $\hat{\mu} = \mu$ follows from the fact that the means of the subsets $\{x_i \geq \mu\}, \{x_i < \mu\}$ are preserved,

[1] It is not generally true that quantizing the sum of two datasets yields the same result as the sum of their quantized versions, i.e., for $\{x_i\}, \{y_i\}$ we generally have $\widehat{x_i + y_i} \neq \hat{x}_i + \hat{y}_i$.

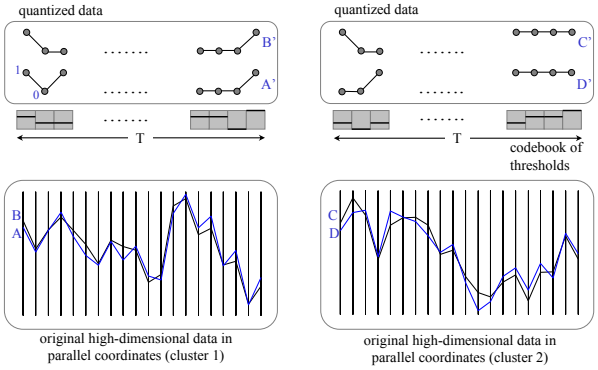

Figure 3: Overview of the proposed quantizer design

cf. (4.5). To prove that the variance is not increased note that

$$\begin{aligned}
\sigma^2 &= \frac{1}{N}[\sum_{x_i<\mu}(x_i-\mu)^2 + \sum_{x_i\geq\mu}(x_i-\mu)^2] \\
&\geq \frac{1}{N}[(N-N_g)(\frac{1}{N-N_g}\sum_{x_i<\mu}x_i - \mu)^2 + \\
&\qquad +N_g(\frac{1}{N_g}\sum_{x_i\geq\mu}x_i - \mu)^2] \qquad (4.13)\\
&= \frac{1}{N}\sum_{i=1}^{N}(\hat{x}_i - \hat{\mu})^2 = \hat{\sigma}^2,
\end{aligned}$$

where we have used Jensen's inequality [16]. As the quadratic function $f(x) = x^2$ is strictly convex, the inequality is strict unless the dataset contains only two distinct points. To prove the second part, note that choosing $a = b = \mu$ is a feasible solution for (4.4) and $C(\mu, \mu) = \sigma^2$. This is suboptimal, unless $x_i = \mu$ for all i (or equivalently $\sigma^2 = 0$). When $\sigma^2 > 0$, the strict convexity of the quadratic function yields a unique solution with $MSE < \sigma^2$. Now consider showing $\hat{d}_{\max} \leq d_{\max}$ with strict inequality unless $x_i = b$ for all $x_i \geq \mu$,; the case $\hat{d}_{\min} \leq d_{\min}$ being analogous. This follows from the fact that

$$\begin{aligned}
\hat{d}_{\max} &= \frac{1}{N_g}\sum_{x_j\geq\mu}(x_j - \mu) \\
&\leq \frac{1}{N_g}N_g \max_{x_j\geq\mu}(x_j - \mu) \\
&= d_{\max}, \qquad (4.14)
\end{aligned}$$

and the inequality is strict unless as specified in the statement of the lemma. To show quasi-linearity of the 1-bit MMSE quantization scheme, consider first the dataset $\{y_i\}_{i=1}^N : y_i = ax_i$. For any a, we have $\mu_y = a\mu$. For $a = 0$ the result is trivial. For $a \neq 0$, let us denote the number of points above the mean by

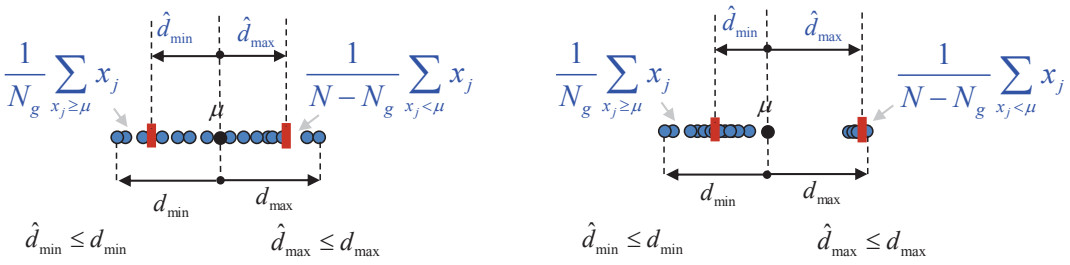

Figure 2: The proposed MMSE quantization always shrinks the cluster extent for both a) well-behaved, and b) ill-behaved clusters

$N_g^{(a)}$; for $a > 0$ we have $N_g^{(a)} = N_g$ and from (4.5) we plainly have $\hat{y}_i = a\hat{x}_i, i = 1, \ldots, N$. For $a < 0$ we have $N_g^{(a)} = N - N_g$ and again the result follows from (4.5). To establish (4.11) note that for $\{z_i\}_{i=1}^N : z_i = x_i + b$, we have $\mu_z = \mu + b$, and N_g remains unaltered, hence using again (4.5) the result follows. Finally (4.12) is a simple application of (4.10), (4.11). ■

We note, in passing, that one could also use a 1-bit Minimum Absolute Error (MAE) quantizer [15] and that it is not hard to show that analogous properties hold, in particular that the dynamic range of the dataset decreases.

5 K-means and MMSE Quantization

We now show how 1-bit MMSE quantization (per dimension and cluster) is sufficient to guarantee preservation of the K-means clustering result. Given are: a) a dataset $X = \{\mathbf{x}_i\}_{i=1}^N$ consisting of N datapoints in T dimensions, and b) the clustering partition outcome of K-means $\mathcal{S} := \{S_k\}_{k=1}^K$. A requirement of the proposed quantization scheme is pre-clustering of the data so that cluster labels can be extracted. This is **necessary** to properly determine where bits are allocated during the quantization phase. Note also that other cluster-preserving techniques that do not require pre-clustering, e.g., [9], do not preserve the 'original shape' of the data at all, because they transform the data into a new space.

We build T 1-bit scalar quantizers per cluster; each scalar quantizer for cluster S_k operates on $N_k := |S_k|$ samples of the same dimension and appropriately maps them into two distinct values via (4.5). Each datapoint is consequently converted to a binary sequence of T ones (1) and zeros (0). Each value corresponds to different reconstruction levels at different dimensions and clusters. This approach is illustrated in Figure 3, where high-dimensional objects are plotted in $2D$ using parallel coordinates: each point in the horizontal axis corresponds to a different dimension.

In what follows, we focus on the standard Euclidean distance, i.e., the L_2-norm which is used in the K-means objective function (2.1). However, the analysis can be extended to hold for any L_q-norm, $1 \leq q \leq \infty$.

5.1 Preservation of the K-means Clustering Outcome

In the case of MMSE quantization, the first moment per cluster is preserved, while cluster variances and dynamic ranges decrease for each dimension, as was shown in Lemma 4.1. This, in turn, implies that the optimal clustering outcome (clusters and cluster centroids) is intuitively expected to remain the same. We also have that the optimal K-means value for the quantized data, $\hat{V}^{opt}$, satisfies $\hat{V}^{opt} \leq \hat{V} \leq V^{opt}$, where $\hat{V}$ is the value corresponding to the optimal assignment for the original dataset applied to the quantized data, and V^{opt} is the optimal clustering value for the original dataset. Note that the inequality $\hat{V} \leq V^{opt}$ (and hence the inequality $\hat{V}^{opt} \leq V^{opt}$) is strict if at least one cluster contains more than two distinct values for a given dimension (cf. Lemma 4.1.3). For each cluster, we consider the smallest *box* (hyper-rectangle) centered at its centroid and containing all points of the cluster.

Definition 1. (Dynamic range box of a cluster) *Let a cluster S be represented by a finite set of points $\{\mathbf{x}_i\}_{i=1}^n \subset \mathbb{R}^T$ belonging to it. The* dynamic range box *B_S of the cluster is defined as*

(5.15)
$$B_S := \{\mathbf{y} \in \mathbb{R}^T : y_j \in [\min_{1\leq i\leq n} x_{ij}, \max_{1\leq i\leq n} x_{ij}], 1 \leq j \leq T\},$$

where y_j is the j-th entry of the T-dimensional vector $\mathbf{y}$.

For an example of the dynamic range on two dimensions see Figure 5. It follows from Lemma 4.1 that for

each cluster S_k the dynamic range box of its quantized version, $B_{\bar{S}_k}$, is a subset of its original dynamic range box B_{S_k}. It is natural to expect that the clustering outcome will be preserved after quantization, at least when the clusters are sufficiently "separated" from one another. We now show that a simple (pre- or post-) processing scheme is *always* sufficient to guarantee cluster preservation with a minimal storage requirement of one additional value for the entire dataset. The idea is to *contract* clusters so that datapoints become more concentrated around centroids; this achieves better cluster "separation."

Given a cluster partition $\{S_k\}_{k=1}^K$ with corresponding centroids $\{\mathbf{c}_k\}_{k=1}^K$, consider a transformation of the original data $\{\mathbf{x}_i\}_{i=1}^N$ so that $\mathbf{x}_i \in S_k$ is transformed to $\bar{\mathbf{x}}_i$ via

$$\bar{\mathbf{x}}_i := \mathbf{c}_k + \alpha(\mathbf{x}_i - \mathbf{c}_k), \quad \alpha \in (0,1]. \tag{5.16}$$

For a given cluster, this transformation is an affine *contraction*, i.e., it reduces the distance between two points in the same cluster, as well as the distance between a given point and the centroid of the cluster it belongs to, by a factor of α. The dynamic range box of the contracted cluster $\bar{S}_k$ is $B_{\bar{S}_k} = c_k + \alpha(B_{S_k} - c_k)$, which is a proper subset of B_{S_k} for $\alpha < 1$; its inter-cluster variance is $\sigma_{\bar{S}_k} = \alpha\sigma_{S_k} \leq \sigma_{S_k}$. The process is schematically depicted in Figure 4.

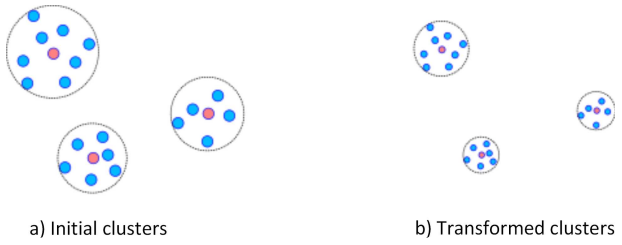

Figure 4: Contractive cluster transformation

As $\alpha \searrow 0$, it is evident that all points collapse to their corresponding cluster centroids, therefore there exists a critical (dataset-dependent) value α_{crit}, such that the aforementioned transformation can guarantee post-quantization preservation of the clustering outcome. This is the essence of the following theorem:

THEOREM 5.1. *Given a dataset $X = \{\mathbf{x}_i\}_{i=1}^N$, there exists a sufficiently small $\alpha \in (0,1]$ such that performing K-means clustering to the dataset obtained by applying the transformation (5.16) to the quantized dataset $\hat{\bar{X}}$ yields the same optimal clustering as for the original dataset X.*

Proof. For a given dataset of N points in a Euclidean space, let us denote the set of all different assignments of the points into K clusters by $\mathcal{S}^{(K)}$; this is a finite set with atoms data partitions $\mathcal{S} = (S_1, \ldots, S_K)$. The set $\mathcal{X} := \{\hat{\bar{x}}_i(\alpha)\}_{\alpha\in[0,1], 1\leq i\leq N}$ is a compact subset of the hyper-rectangle B_X because quantization does not increase (decrease) the maximum (minimum) values of a dataset (cf. Lemma 4.1.3). For a given $\alpha \in [0,1]$, the K-means clustering problem for the contracted quantized data becomes

$$\min_{\mathcal{S}\in\mathcal{S}^{(K)}} F(\mathcal{S},\alpha) := \sum_{k=1}^K \sum_{\hat{\bar{\mathbf{x}}}_j(\alpha)\in S_k} ||\hat{\bar{\mathbf{x}}}_j(\alpha) - \frac{1}{|S_k|}\sum_{\hat{\bar{\mathbf{x}}}_i(\alpha)\in S_k} \hat{\bar{\mathbf{x}}}_i(\alpha)||^2. \tag{5.17}$$

Let $\mathcal{S}^*$ be the given optimal data partitioning corresponding to the original dataset, which we use to perform both quantization and data transformation. For a given $\alpha \in [0,1]$, let the set of optimal cluster assignments for the transformed version of the quantized dataset via (5.16) be denoted by $\mathcal{S}_{opt}^{(\alpha)} \subset \mathcal{S}^{(K)}$. For $\alpha = 0$, we have that $\mathcal{S}_{opt}^{(0)} = \{\mathcal{S}^*\}$ is the unique optimal clustering assignment for the transformed quantized dataset with $F(\mathcal{S}^*, 0) = 0$, while $F(\mathcal{S}, 0) > 0$ for any $\mathcal{S} \neq \mathcal{S}^*$. Note that for a given S the function $g(\alpha) := F(S,\alpha)$ is a uniformly continuous function on $[0,1]$, so there exists an $\alpha^* > 0$, such that $\mathcal{S}^*$ is the *unique* optimal clustering partition for each $\alpha \in [0, \alpha^*)$. ■

The proposed transformation can be considered an encoding-decoding procedure: first, a user (encoder) clusters the original dataset using K-means. Given the calculated cluster partition $\{S_k\}_{k=1}^K$, it quantizes the data using 1-bit MMSE quantizers per cluster and dimension. After that, it selects an $\alpha \in (0,1)$, which can be considered as a coding key, and transforms the quantized data via (5.16) [2]. The transformed quantized dataset, $\hat{\bar{X}} = \{\hat{\bar{\mathbf{x}}}_i\}_{i=1}^N$, is stored along with the value α and can be transmitted to another user (decoder), who can then run K-means on that. The result will be identical to performing K-means clustering on X. While the clustering outcome is maintained, the distortion due to applying the data transformation described above might be significant; this scheme can be of interest for data-hiding applications. To retrieve the quantized version of the original dataset, the user (decoder) must have the key α to apply the transformation:

$$\hat{\mathbf{x}}_i = \frac{\hat{\bar{\mathbf{x}}}_i - (1-\alpha)\mathbf{c}_k}{\alpha}. \tag{5.18}$$

Picking the contraction factor: We have seen that a sufficiently small value of $\alpha \in (0,1]$ is guaranteed to

[2] These two operations can also be carried out in reversed order, i.e., transformation can precede quantization; the result is the same, cf. (4.12).

preserve K-means clustering after quantization. The question of practical interest is two-fold: a) is it necessary to perform the transformation, and b) if so, how small should α be?

The answer lies in how well clusters are "separated" from one another. Let us consider the following *separation* property between clusters in an optimal data partitioning $\mathcal{S}$: for each $1 \leq k \leq K$, each point in the dynamic range box B_{S_k} is closer to the centroid $\mathbf{c}_k$ than any other centroid $\mathbf{c}_l, l \neq k$, formally:

(5.19)
$$||\mathbf{x} - \mathbf{c}_k|| = \min_{1 \leq l \leq K} ||\mathbf{x} - \mathbf{c}_l||, \quad \forall x \in B_{S_k}, 1 \leq k \leq K.$$

Figure 5.a) depicts an example of two "well-separated" clusters, whereas Figure 5.b) shows a case where the clusters are not "well-separated". Figure 5.c) illustrates how the clusters can be made sufficiently separated using the proposed contraction operation.

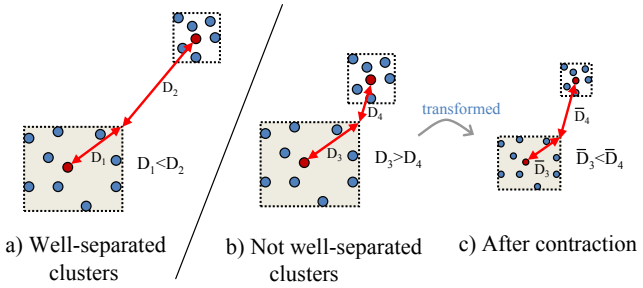

Figure 5: The dashed rectangles depict dynamic range boxes. a) Example of two clusters that are "well-separated" according to our definition, b) Two clusters that are not "well-separated". c) Two clusters made sufficiently separable after applying the contraction transformation

Note that after quantization, each datapoint remains in the dynamic range box of its corresponding cluster, as it follows from Lemma 4.1.3 that $B_{\hat{S}_k} \subset B_{S_k}$. Therefore, property (5.19) guarantees that every point remains, post-quantization, closer to its corresponding cluster centroid than any other cluster centroid. This is the notion of *well-separateness* among clusters, which is actually a sufficient condition to guarantee preservation of (at least) local optimality with regard to Lloyd's algorithm. Furthermore, we can establish that this condition cannot be relaxed. We skip the details of the analysis for the sake of brevity, but show how to pick a value for α to guarantee that property (5.19) holds for the transformed dataset if it does not hold for the original dataset. If this property is already satisfied, then using 1-bit MMSE quantization can preserve the clustering structure without the need to perform the transformation.

For a given $\alpha \in (0, 1]$, consider the following set of $K(K-1)$ Quadratic Programs (QPs), Q_{kl}, $1 \leq k \neq l \leq K$:

$$\begin{aligned} (5.20) \qquad \min_{\mathbf{x}} \quad & ||\mathbf{x} - \mathbf{c}_l|| \\ \text{s.t.} \quad & \mathbf{x} \in B_{S_k^{(\alpha)}} \end{aligned}$$

where $B_{S_k^{(\alpha)}} := c_k + \alpha(B_{S_k} - c_k)$ is the dynamic range box corresponding to the α-contracted dataset $S_k^{(\alpha)}$. For ease of representation let us define the two extreme points $\mathbf{l}_k^{(\alpha)}, \mathbf{u}_k^{(\alpha)}$ that fully characterize $B_{S_k^{(\alpha)}}$:

$$(5.21) \qquad l_{kj}^{(\alpha)} := (1-\alpha)c_{kj} + \alpha \min_{\mathbf{x}_i \in S_k} x_{ij}$$

$$(5.22) \qquad u_{kj}^{(\alpha)} := (1-\alpha)c_{kj} + \alpha \max_{\mathbf{x}_i \in S_k} x_{ij}$$

for each $1 \leq j \leq T$. The unique solution $\mathbf{x}^*(k, l, \alpha)$ to (5.20) is given by

$$(5.23) \qquad x_j^*(k,l,\alpha) = \begin{cases} c_{lj}, & \text{if} \quad c_{lj} \in [l_{kj}^{(\alpha)}, u_{kj}^{(\alpha)}] \\ u_{kj}^{(\alpha)}, & \text{if} \quad c_{lj} > u_{kj}^{(\alpha)} \\ l_{kj}^{(\alpha)}, & \text{if} \quad c_{lj} < l_{kj}^{(\alpha)} \end{cases}$$

Calculating the *critical* value α_{crit}, i.e., the maximal value for α such that property (5.19) holds for the α-transformed dataset, can be done as follows: for each Q_{kl}, define α_{kl} as the maximal value of $\alpha \in (0, 1]$ such that $||\mathbf{x}^*(k, l, \alpha) - \mathbf{c}_l|| \geq ||\mathbf{x}^*(k, l, \alpha) - \mathbf{c}_k||$, which can be efficiently calculated using (5.23). Setting

$$(5.24) \qquad \alpha_{crit} := \min_{k,l} \alpha_{kl}$$

and performing the transformation with α_{crit} is sufficient to satisfy (5.19); we show in the experimental section that performing the transformation using this value did indeed preserve the clustering assignments in *all* test cases. The above process is summarized in Figure 6.

5.2 Compression Efficiency

For a set of N objects with T dimensions clustered into K clusters, let each unquantized sample be represented by B bits per dimension; the total storage requirement is BTN bits for the unquantized data. Using a 1-bit quantizer we need only store a total of $\log_2(2TK)TN$ bits to represent the quantized data. This is because there are only two possible values that each of the T dimensions can take per cluster, and a total of K clusters, i.e., at most $2TK$ values. A better compression ratio can be achieved by noting that because the quantization preserves the underlying clustering structure, one does not explicitly need to store

Given is a dataset $X = \{\mathbf{x}_i\}_{i=1}^N \subset \mathbb{R}^T$ to be partitioned into K clusters via K-means. Our approach involves the following steps:

1. Numerically solve the K-means problem to partition the dataset into K clusters, e.g., using Lloyd's algorithm [17]. The outcome is a partition $\mathcal{S} = \{S_1, \ldots, S_K\}$, with corresponding cluster centroids $\{\mathbf{c}_1, \ldots, \mathbf{c}_K\}$.

2. For each cluster S_k, quantize its elements using 1-bit MMSE quantizers per dimension via (4.5). This results in a total storage requirement of $TN + 2BKT$ bits, B being the number of bits used to represent unquantized data.

3. Calculate the value α_{crit} through (5.24), and set $\alpha = \alpha_{crit}$. Transform the quantized dataset $\hat{X}$ and store $\bar{\hat{X}}$, where $\hat{\mathbf{x}}_i \in S_k$ is transformed to $\bar{\hat{\mathbf{x}}}_i$ via

$$\bar{\hat{\mathbf{x}}}_i := \mathbf{c}_k + \alpha(\hat{\mathbf{x}}_i - \mathbf{c}_k). \tag{5.25}$$

4. At the stored data, a user performing K-means clustering obtains the same results (both cluster assignments and centroids). If the user in addition knows α, he or she can retrieve the quantized version of the original dataset, $\{\hat{\mathbf{x}}_i\}_{i=1}^N$, by calculating for each $\bar{\hat{\mathbf{x}}}_i \in S_k$

$$\hat{\mathbf{x}}_i = \frac{\bar{\hat{\mathbf{x}}}_i - (1-\alpha)\mathbf{c}_k}{\alpha}. \tag{5.26}$$

Figure 6: Algorithm for cluster-aware quantization and contractive transformation

the cluster labels. Hence, it suffices to use TN bits for all objects, along with $2BTK$ bits to store the two reconstruction levels per dimension and per cluster. The threshold does not need to be explicitly stored as it can be deduced from the reconstructed samples, since MMSE quantization (with or without contraction) does not distort the mean. The compression ratio $\rho := \frac{\text{bytes quantized}}{\text{bytes unquantized}}$ achieved by our quantization scheme is

$$\rho = \frac{TN+2BTK}{BTN} = \frac{1}{B} + \frac{2K}{N}. \tag{5.27}$$

For the case of highest practical interest $B > 1$ and $N >> K$, the compression ratio satisfies $\frac{1}{B} < \rho < 1$. In the experimental part Section 6.3, we evaluate the compression ratio of the quantization scheme on real datasets.

5.3 Multi-bit Quantization

We have shown that a single-bit quantization scheme is sufficient to maintain the optimal clustering structure. We now consider a multi-bit extension that can provide a better reconstruction of the original objects. This may be useful when the user is interested in supporting additional tasks, such as data visualization.

Let $Q = 2^q$ be the number of quantization levels, with $q \geq 1$ being the number of bits needed to represent the dimension of each sample. The interesting case is when $Q < N$. The proposed quantization scheme amounts to sequentially breaking the dataset into Q sub-datasets by recursively constructing a hierarchical binary tree with at most Q leaves. The root represents the entire dataset to be quantized. Then, at level (i.e., depth) $i < q$, each node represents a sub-dataset. For each such node, if it contains two or more datapoints, we calculate the mean of the corresponding sub-dataset and further divide it into two subsets, one containing the values greater than or equal to the mean, and the other the values lower than the mean. The mean of the sub-dataset is set as a threshold value. Finally, when the node represents a singleton sub-dataset, we set this as a leaf node and keep the data-point value as is. At the final stage ($i = q$), for each resulting sub-dataset $n = 1, \ldots, \bar{N}$, $\bar{N} \leq Q$, we perform 1-bit MMSE quantization to quantize its values using (4.5). Figure 7 presents an example of the proposed multi-bit quantization scheme for $q = 3$ bits.

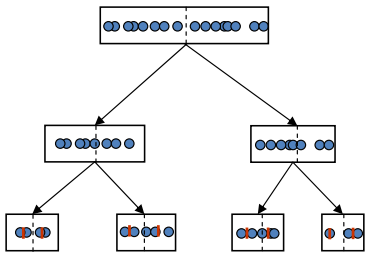

Figure 7: Multi-bit MMSE quantization ($q = 3$ bits). Quantization levels are depicted by red marks at the leaf nodes; threshold values are represented by dashed lines at the non-leaf nodes

It is not difficult to verify that the proposed scheme satisfies all the properties of Lemma 4.1, but we skip the proof for space considerations. Note also that because of the suggested design the mean is preserved for every sub-dataset corresponding to a given node of the binary tree. Therefore, the threshold levels need not be stored as they can be reconstructed from the quantized dataset.

Increasing Q improves the fidelity of the compressed

vectors so that we may use different values of Q for different clusters based on the desired fidelity. We present a Rate-Distortion Optimization problem and a simple greedy algorithm for multi-bit allocation with the goal of minimizing the mean square compression error, given constraints on the available storage capacity.

Greedy algorithm for multi-bit allocation

Inputs: $\{\{\text{MSE}_k(B_k)\}_{B_k=1}^{U_k}\}_{k=1}^{K}$

Outputs: $\{B_k\}_{k=1}^{K}, \text{MSE} := \frac{1}{N}\sum_{k=1}^{K} N_k \text{MSE}_k(B_k)$

1. For each cluster $1 \le k \le K$, define the relative MSE improvement attained when using q instead of $q-1$ bits ($q = 2, \ldots, U_k$) by $I_k(q) = \text{MSE}_k(q) - \text{MSE}_k(q-1)$
2. Set $B_k \leftarrow 1$ for all k, and set the unused budget: $R \leftarrow \bar{B} - NT - 2BTK$
3. If $R \le 0$ return $\{B_k\}_{k=1}^{K}$ and MSE
4. else define $\mathcal{K} = \{1, \ldots, K\}$
5. Let $k^* = \text{argmin}_{k \in \mathcal{K}} N_k I_k(B_k + 1)$
6. If $R - TN_{k^*} - BT2^{B_{k^*}} > 0$
   $B_{k^*} \leftarrow B_{k^*} + 1$;
   $R \leftarrow R - TN_{k^*} - BT2^{B_{k^*}}$;
   go to step 3;
7. else set $\mathcal{K} \leftarrow \mathcal{K} \setminus k^*$
8. If $\mathcal{K} == \emptyset$ return $\{B_k\}_{k=1}^{K}$ and MSE
9. else go to step 5
10. endif
11. endif
12. endif

Figure 8: Algorithm for multi-bit quantization

Optimal bit allocation for multi-bit quantization: We consider using different values of Q for different clusters based on their relative importance or desired fidelity. We focus on the case where there is a total budget of $\bar{B}$ bits and formulate the allocation problem as one of minimizing the MSE due to quantization. Denoting the number of bits allocated to cluster k (per sample and dimension) by B_k, we need to have $\sum_{k=1}^{K}(TB_kN_k + BT2^{B_k}) \le \bar{B}$, and we trivially need to assume that $\bar{B} \ge TN + 2BTK$ as we need at least one bit per cluster and dimension. From this, it also follows that $B_k \le U_k$, where

(5.28)
$$U_k := \min\{B_k : T(B_k-1)N_k + BT2^{B_k-1} \le \bar{B} - TN - 2BTK\}.$$

For cluster k, we can perform multi-bit MMSE quantization with $B_k = 1, \ldots, U_k$ bits and calculate the corresponding MSE. The hierarchical structure of the multi-bit MSE quantizer guarantees linear complexity in all T, N_k, U_k in calculating $\text{MSE}_k(B_k)$. Then the allocation problem becomes a combinatorial problem of the form:

$$\text{(5.29)} \quad \min_{B_1,\ldots,B_K} \quad \frac{1}{N}\sum_{k=1}^{K} N_k \text{MSE}_k(B_k)$$

$$\text{(5.30)} \quad \text{s.t.} \quad \sum_{k=1}^{K}(TB_kN_k + BT2^{B_k}) \le \bar{B}.$$

An exhaustive search is intractable even for small values of K for a large enough $\bar{B}$. Therefore, we propose a simple greedy algorithm with linear complexity in $\sum_k U_k$ and K. The idea is to sequentially allocate one extra bit to the cluster that will decrease the objective function the most until the storage constraints have been reached. We provide a description of the greedy algorithm in Figure 8.

6 Experiments

In this section, we validate the performance of the proposed quantization schemes on real datasets: we examine the effect on cluster preservation and assess the distortion incurred to the original data due to quantization. We compare 1-bit MMSE quantization vs. 1-bit MPQ [14], and show that they both achieve excellent cluster preservation when Lloyd's algorithm is used with the K-means++ centroid initialization scheme [18], whereas MMSE quantization leads to lower distortion. We further illustrate that the proposed transformation (5.16) indeed preserves the clustering outcome for *all* instances. We show that using multi-bit MMSE quantization has the benefit of significantly reducing object distortion while accurately preserving the clustering outcome. For our experiments, we use data from publicly available stock market time series corresponding to 2169 stock symbols from companies listed on NASDAQ, reporting the stock values for a period of approximately three years.

6.1 Cluster Preservation

Because the exact solution for K-means is NP-hard, we use the popular gradient-descent Lloyd's algorithm for computational efficiency, and experimentally evaluate the discrepancy between the clustering results. We do not have the cluster labels for the aforementioned dataset, so we cluster the original time series

Table 1: Cluster preservation after quantization with optional contractive data transformation. We present the fraction of data belonging to the same clusters as before quantization using a) MPQ, b) q-bit MMSE for $q = 1, 2, 4$ denoted by MMSE(q). The quantized version of the dataset after the contractive transformation is denoted by a "_t" after the name of the quantization scheme. We also show the critical value for the contraction parameter α_{crit}. Experiments are for $K = 3, 5, 8$ clusters. Note that the contraction indeed yields perfect cluster preservation in all instances.

K	MPQ	MMSE(1)	MMSE(2)	MMSE(4)	α_{crit}	MPQ_t	MMSE_t(1)	MMSE_t(2)	MMSE_t(4)
3	1	1	0.991	0.990	0.505	1	1	1	1
5	1	1	0.994	0.994	0.503	1	1	1	1
8	0.999	1	0.980	0.988	0.098	1	1	1	1

into $K = 3, 5, 8$ clusters using Lloyd's algorithm with a maximum number of 100 iterations. Because we would like to have a near-optimal set of clusters, we repeat Lloyd's algorithm with multiple starting centroids and select the set that achieves the smallest value for the K-means objective function. We found that using 5 runs of Lloyd's algorithm with different starting points performed well in all cases. We tested two alternatives for choosing starting centroids: a) selecting K out of the N datapoints uniformly at random, and b) using the K-means++ [18] initialization scheme to select K out of the N datapoints so as to achieve an initial estimate that is expected to be a good approximation with provable properties.

All results reported correspond to the K-means++ initialization, as it showed a uniformly better performance and convergence speed in our experiments. In executing Lloyd's algorithm, it is quite common that a cluster may become empty, i.e., that no points are closer to its corresponding centroid than to any other given centroid. In that case, we consider keeping the centroid as a "singleton" cluster, and proceed with Lloyd's algorithm steps. In Figure 9, we consider $K = 8$ and illustrate the clusters formed, along with the centroids and upper and lower quantization values. After obtaining the clustering assignment via Lloyd's algorithm, we use it to quantize the time series using separate 1-bit MMSE quantizers (one per cluster and dimension). We also use MPQ quantizers for comparison with the approach of [14], as well as multi-bit MMSE quantizers with $q = 2, 4$ bits. We then perform clustering on the quantized dataset, and compare the resulting cluster centroids and cluster assignments before and after quantization. We also consider performing the transformation (5.16) after quantization, using the value α_{crit} as defined in (5.24).

To quantify cluster preservation accuracy, we consider matching clusters as follows: let us denote the clusters formed in the original dataset by $\{S_k\}_{k=1}^K$ and

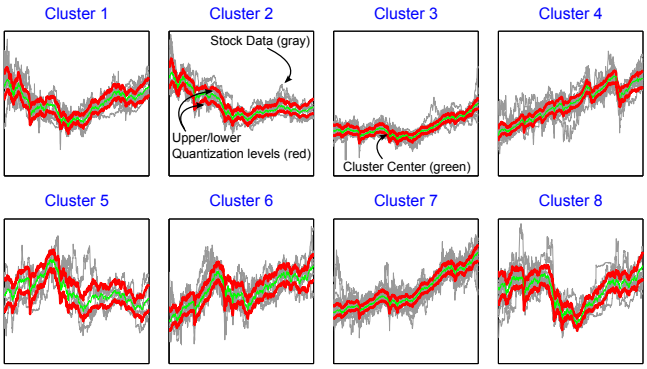

Figure 9: Stock data grouped into 8 clusters. Note that the upper and lower quantization values (depicted in red) accurately track the cluster trend, and that clusters 'shrink' after quantization

those formed in the quantized dataset by $\{S'_k\}_{k=1}^K$. Denote the cardinality of the intersection of the quantized datapoints of S_k with the members of S'_l by $\text{int}_{k,l} := |\{x_i : x_i \in S_k, \hat{x}_i \in S'_l\}|$. After calculating $\text{int}_{k,l}$, we consider the permutation $p(\cdot)$ of $\{1, \ldots, K\}$ such that $\sum_{k=1}^K \text{int}_{k,p(k)}$ is maximized. The *cumulative preservation metric cp* taking values in $[0, 1]$ is then defined as the percentage of points that belong to the intersection of the original and the matched clusters, namely

$$cp := \frac{1}{N} \sum_{k=1}^{K} \text{int}_{k,p(k)}. \tag{6.31}$$

We present the values for the cumulative preservation metric for a) MPQ, b) q-bit MMSE for $q = 1, 2, 4$ denoted by MMSE(q) as well as the quantized version of the transformed dataset, denoted by a "_t" after the name of quantization in Table 1; we considered $K = 3, 5, 8$ clusters. We note that the quality of cluster preservation is excellent: more than 98% of the samples belong to the same clusters before and after quantization, whereas the transformation *always* yields per-

fect cluster preservation. Note also that the MMSE(2) and MMSE(4) lead to slightly worse cluster preservation than MMSE(1); this is attributed to Lloyd's algorithm, as a higher data resolution (more bits) typically leads to a lower dynamic range reduction post-quantization.

6.2 Data Distortion

We study the distortion induced on the original data by the proposed quantization schemes. We record the normalized MSE per dimension, MSE/T in Table 2. It is evident that using 1-bit MMSE quantization incurs significantly less distortion than MPQ, namely 37% less MSE on average in all cases.

Table 2: MSE due to quantization (per dimension)

K	MPQ	MMSE(1)	MMSE(2)	MMSE(4)
3	131.4	89.3	57.1	9.6
5	45.3	27.3	10.0	1.2
8	29.5	18.1	7.0	0.7

Using multi-bit quantization further reduces the MSE substantially over 1-bit MPQ, by, on average, $70\% and 96\%$, on average, for $q = 2$ and $q = 4$ bits, respectively. Furthermore, the MSE is decreased by increasing the number of clusters. This is because our quantization schemes are cluster-centric, whence increasing K increases the number of quantizers, while decreasing the number of datapoints to be jointly quantized (those that belong to the same cluster).

We provide one visual example of how multi-bit quantization reduces the data distortion in Figure 10. We show a sample time series for the stock dataset and its quantized version using 1-bit and 4-bit MMSE, as well as the absolute error due to quantization.

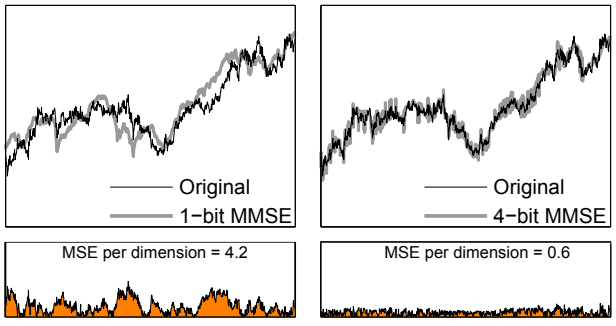

Figure 10: Original and quantized time series for a stock: on the left side we show the quantized time series using 1-bit MMSE. On the right side we depict the quantized time series using 4-bit MMSE. The bottom panels capture the absolute quantization error

Table 3: Compression Efficiency

# of clusters	# of bits (q)	Compression (ρ)
K=3	1	0.128
	2	0.256
	4	0.522
K=5	1	0.13
	2	0.259
	4	0.537
K=8	1	0.132
	2	0.265
	4	0.559

6.3 Compression Efficiency

We have seen that increasing the number of bits to represent each quantized datapoint and also increasing the number of clusters helps better preserve the shape of the time series while maintaining excellent cluster preservation performance. This, however, comes at the price of increased storage requirements. We quantify this by calculating the compression ratio:

$$\rho = \frac{qTN + 2^q BTK}{BTN} \tag{6.32}$$

when we use q bits for each cluster and dimension. We present the compression efficiency for all cases in Table 3, by assuming that the original (non-quantized) data are represented using $B = 8$ bits.

As can be seen from the table, the compression can result in a storage reduction of almost a factor of 8. The compression ratio varies with the number of clusters and bits, deteriorating as $K and q$ increase. Selecting the optimal trade-off between compression, distortion, and cluster label preservation is an important practical consideration for the proposed scheme.

7 Discussion and Extensions

A key feature of the proposed scheme is that data have to be pre-clustered and, consequently, that quantization is performed separately for each cluster. This is justified because that the original data is considered unlabeled and cluster-unaware quantization schemes cannot exploit the cluster structure for asserting post-compression cluster preservation. In particular, it is plain to see that applying quantization to the dataset as a whole using a few bits will typically alter the data topology to the extent that clustering will be vitally distorted. In our approach, we can control the compression granularity by allocating different numbers of bits per cluster and dimension, so that the compressed data sam-

ples retain the original object structure and neighborhood relationships. Therefore, the proposed compression method can also be used for driving other distance-based mining operations, including, but not limited to: hierarchical clustering, visualization, and approximate search.

The result of Theorem 5.1 can be extended in two directions of practical interest. First, it also applies to *any* quantization scheme, not just MMSE quantization. In particular, it applies to MPQ used in [14], which we have tested in the experimental section. To see this, note that in the proof of the theorem, $\hat{\mathbf{x}}_i$ is simply a constant independent of α and there exists a small enough $\alpha \in (0, 1)$ to guarantee preservation of K-means clustering. Of course, in that case, α depends on the quantization scheme and needs to be, in general, smaller for schemes that do not guarantee dynamic range reduction, such as MPQ. Second, the proof of Theorem 5.1 also carries over for *any given* cluster assignment $\mathcal{S}$, not just the optimal assignment for K-means clustering S^*; again α depends on $\mathcal{S}$.

The latter extension is important for two reasons: First, by using Lloyd's algorithm, we typically acquire a suboptimal assignment, whereas K-means++ is a randomized algorithm that will yield a different outcome in each realization. In both cases, it is desirable to maintain the clustering outcome obtained, which is guaranteed by our theory.

8 Conclusions and Future Work

We have showcased compression schemes for high-dimensional datasets that preserve the outcome of K-means clustering. Our analytical derivations indicate that a single-bit MMSE quantizer, per cluster and dimension, is sufficient to preserve the optimal cluster assignment if the clusters satisfy a certain "separation" property. We have presented a minimum-overhead linear data transformation that can guarantee such a property for the transformed data, and have further proved that such a scheme can *always* guarantee post-quantization cluster optimality. Finally, we have considered multi-bit quantization and proposed an efficient greedy algorithm for bit allocation in order to minimize the MSE due to quantization. Our experimental evaluations have shown that the quantization schemes designed indeed preserve the clustering outcome, while inducing only minimal distortion on the original data.

References

[1] M. Luo, Y.-F. Ma, and H.-J. Zhang, "A Spatial Constrained K-Means Approach to Image Segmentation," in *IEEE Int. International Conference on Information, Communications and Signal Processing*, 2003, pp. 738–742.

[2] A. Anagnostopoulos, A. Dasgupta, and R. Kumar, "Approximation algorithms for co-clustering," in *Proc. of PODS*, 2008, pp. 201–210.

[3] S. Das and S. Idicula, "KMeans greedy search hybrid algorithm for biclustering gene expression data," in *Advances in Exp. Medicine and Biology. 2010*, 2010, pp. 181–188.

[4] F. Bach and M. Jordan, "Learning spectral clustering," in *Proc. of NIPS*, 2004.

[5] J. Lin, M. Vlachos, E. J. Keogh, and D. Gunopulos, "Iterative Incremental Clustering of Time Series," in *Proc. of EDBT*, 2004, pp. 106–122.

[6] P. S. Bradley, U. M. Fayyad, and C. Reina, "Scaling Clustering Algorithms to Large Databases," in *Proc. of SIGKDD*, 1998, pp. 9–15.

[7] J. Vaidya and C. Clifton, "Privacy-preserving k-means clustering over vertically partitioned data," in *Proc. of SIGKDD*, 2003, pp. 206–215.

[8] G. Jagannathan and R. N. Wright, "Privacy-preserving distributed k-means clustering over arbitrarily partitioned data," in *Proc. of SIGKDD*, 2005, pp. 593–599.

[9] R. Parameswaran and D. Blough, "A Robust Data Obfuscation Approach for Privacy Preservation of Clustered Data," in *Workshop on Privacy and Security Aspects of Data Mining*, 2005, pp. 18–25.

[10] S. R. M. Oliveira and O. R. Zaane, "Privacy Preservation When Sharing Data For Clustering," in *Intl. Workshop on Secure Data Management in a Connected World*, 2004.

[11] A. J. Bagnall, C. A. Ratanamahatana, E. J. Keogh, S. Lonardi, and G. J. Janacek, "A Bit Level Representation for Time Series Data Mining with Shape Based Similarity," in *Data Min. Knowl. Discov. 13(1)*, 2006, pp. 11–40.

[12] J. Aßfalg, H.-P. Kriegel, P. Kröger, P. Kunath, A. Pryakhin, and M. Renz, "T-Time: Threshold-Based Data Mining on Time Series," in *Proc. of ICDE*, 2008, pp. 1620–1623.

[13] V. Megalooikonomou, G. Li, and Q. Wang, "A dimensionality reduction technique for efficient similarity analysis of time series databases," in *Proc. of CIKM*, 2004, pp. 160–161.

[14] D. Turaga, M. Vlachos, and O. Verscheure, "On K-Means Cluster Preservation Using Quantization Schemes," in *IEEE International Conference on Data Mining (ICDM)*, 2009, pp. 533–542.

[15] E. Delp, M. Saenz, and P. Salama, "Block Truncation Coding (BTC)," in *The Handbook of Image and Video Processing*. Academic Press, 2000.

[16] H. L. Royden, *Real Analysis*. Prentice-Hall, Englewood Cliffs, NJ, 1988.

[17] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley-Interscience, 1973.

[18] D. Arthur and S. Vassilvitskii, "k-Means++: The Advantages of Careful Seeding," in *Proc. of Symposium of Discrete Analysis*, 2005.