Right-Protected Data Publishing with Hierarchical Clustering Preservation

Michail Vlachos
IBM Research - Zurich

Aleksander Wieczorek
Poznan University of Technology, Poland

Johannes Schneider
IBM Research - Zurich

ABSTRACT

The emergence of cloud-based storage services is opening up new avenues in data exchange and data dissemination. This has amplified the interest in right-protection mechanisms to establish ownership in case of data leakage. Current right-protection technologies, however, rarely provide strong guarantees on the dataset utility after the protection process. This work presents techniques that explicitly address this shortcoming and provably preserve the outcome of certain mining operations. In particular, we take special care to guarantee that the outcome of hierarchical clustering operations remains the same before and after right protection. We encode data ownership using watermarking principles. In the process, we derive fundamental bounds on the distortion incurred by the watermarking. We leverage our theoretical analysis to design fast algorithms for right protection without exhaustively searching the vast design space.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering

Keywords

Watermarking, Distortion Estimation

1. INTRODUCTION

Data exchange and data sharing have become an inherent part of business and academic efforts. Both practices encourage scientific enquiry, ease validation of research efforts and maximize transparency. As such, data sharing and data publishing are recognized as important productivity catalysts in diverse research efforts. To offer a concrete example, it is widely recognized that initiatives such as the human genome project [1] that advocated data sharing lead to "rapid scientific breakthroughs that otherwise would not have occurred" [1]. Recently, even fields that viewed data sharing through a negative prism (e.g., banking industry), have promoted the establishment of consortia to ease data exchange [6]. Owing to the high availability of cloud-based services in the near future, similar initiatives are projected to experience a surge in demand, as attested in many recent studies [11, 17, 10].

Data owners, nonetheless, need also maintain principal rights over the shared datasets, datasets which in many cases have been obtained after laborious procedures. This work presents a protection mechanism that can deliver detectable evidence on the legal ownership of a shared dataset, without compromising its usability for a class of mining operations. To achieve this, we guarantee that important distance-based relationships between the dataset objects remain unaltered.

We embed ownership evidence using watermarking techniques. Watermarking has emerged over the years as a successful method for establishing data progeny. It has been used extensively in many multimedia applications, on image, video, and audio data. Traditional watermarking techniques focus on a single data object and are not tailored for preserving relationships between multiple objects. In that sense, our technique augments and strengthens existing watermarking methodologies. Our goal is two-fold: to guarantee right-protection and, at the same time, preserve the original relationships between the dataset objects. Having accomplished this, any learning or retrieval task that depends on the preserved structural properties will remain undistorted even after the watermark application.

In this work, we explicitly show how to **preserve hierarchical clustering** (HC). HC is a popular knowledge extraction tool, because it can visually communicate the similarity between objects and groups of objects. Because of its descriptive power and ease of implementation it is a valuable tool in many disciplines, including:

- Biology and Bioinformatics, for the construction of phylogenetic trees between species [16].
- Natural Sciences, for the taxonomic categorization of plants or animals based on their similarity to previously categorized objects [23].
- Business Analytics and Marketing, for performing customer-base segmentation and aiding the discovery of common customer profiles [28].

Our objective is to maximize the knowledge we can garner from the watermarked data. Here, we give provable guarantees of identical outcome for HC algorithms on the original and watermarked dataset. To achieve this we provide a **theoretical analysis** of the distance distortion due

[1] `http://scientificdatasharing.com/about/`

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.

to watermarking. We derive tight bounds on the expansion/contraction of distances caused by multiplicative watermarking techniques. We exploit these results to engineer **fast watermarking variants** that drastically prune the parameter search space, compared to the exhaustive algorithms.

2. OVERVIEW OF OUR APPROACH

Our goal is to discover how to right-protect a dataset so that the dendrogram resulting from the hierarchical clustering after the right protection is isomorphic to the one on the original data (see Fig. 1). This translates into studying with what watermark *intensity* to protect the dataset so that important parts of the dataset graph are not distorted. We study how to achieve this goal for both single- and complete-linkage hierarchical clustering.

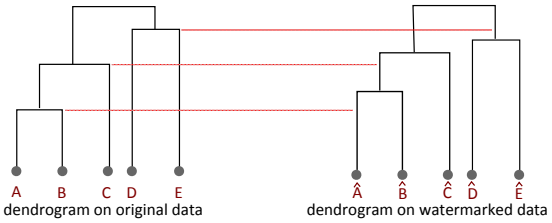

Figure 1: Our goal is to guarantee 'isomorphic' dendrograms before and after right-protection

It is essential to discover the maximum watermark intensity for right protection. This provides assurances of better detectability and hence security for the right-protection scheme. Therefore, we first study how (Euclidean) distances between the objects are distorted as a parameter of the watermark embedding strength. This provides insight into designing fast variants of our algorithms that still guarantee hierarchical clustering preservation, but operate significantly faster than the exhaustive algorithms.

Our paper is structured as follows: First, we describe how right-protection can be materialized via a spread-spectrum watermarking approach. We also show how to detect the watermark. Subsequently, we study the distortion of distances due to watermarking. We describe single- and complete-linkage algorithms and the necessary conditions to preserve them post-watermarking. We provide a theorem that gives tight lower and upper bounds on the distance distortion. We use it to design faster HC-preservation algorithms that are based on the bounds derived. Finally, we provide a comprehensive set of experiments, and we review the related work.

3. RIGHT PROTECTION THROUGH WATERMARKING

We describe first how watermarking mechanisms can embed a secret key (watermark) on a collection of objects. We demonstrate the techniques for 2D sequence data (image contours, trajectories, etc). We later demonstrate how to detect the watermark using a correlation filter.

3.1 Watermark Embedding

Assume an object represented as a vector of complex numbers $x = \{x_1, \ldots, x_n\}$, where $x_k = a_k + b_k i$ (i is the imaginary unit, $i^2 = -1$), and where the real and imaginary parts, a_k and b_k respectively, describe the coordinates of the k-th point of object x on the imaginary plain. Such a model can describe data trajectories or even image contour data which capture coordinates of a shape perimeter, as shown later in Figure 10.

We adapt a spread-spectrum approach [7]. This embeds the watermark across multiple frequencies of each object and across multiple objects of the dataset. As such, it renders the removal of the watermark particularly difficult without substantially compromising the data utility. An object x is mapped into the frequency domain using its complex Fourier descriptors $X = \{X_1, \ldots, X_n\}$. The mapping from the space domain to the frequency domain is described by the normalized discrete Fourier transform, $DFT(x)$, and its inverse, $IDFT(X)$. Every coefficient X_j can be expressed as a function of its *magnitude* δ_j and *phase* ϕ_j as $X_j = \delta_j e^{\phi_j i}$. The watermark constitutes a piece of *secret information* to be hidden inside each sequence. In our approach, we consider the watermark to be a vector $W \in \{-1, 0, +1\}^n$, which is embedded in all objects of the dataset.

DEFINITION 3.1. *(Watermark Embedding (W, p)) Given are a sequence $x \in \mathbb{C}^n$ with corresponding set of Fourier descriptors X, a watermark $W \in \mathbb{R}^n$ and power $p \in [0, 1]$, which specifies the intensity of the watermark. A* ***multiplicative watermark embedding*** *(W, p) generates a watermarked sequence $\widehat{x}$ by replacing the magnitudes of each Fourier descriptor of x with a watermarked magnitude $\widehat{\delta_j}$ while not altering the phases, specifically:*

$$\widehat{\delta_j} = \delta_j \cdot (1 + pW_j) \ , \ and \ \widehat{\phi_j} = \phi_j$$

Using the modified magnitudes $\widehat{\delta_j}$ and the original phases ϕ_j, we can revert from the frequency domain to the space domain and obtain the watermarked sequence using the inverse discrete Fourier transform. An overview of the methodology described is given in Fig. 2.

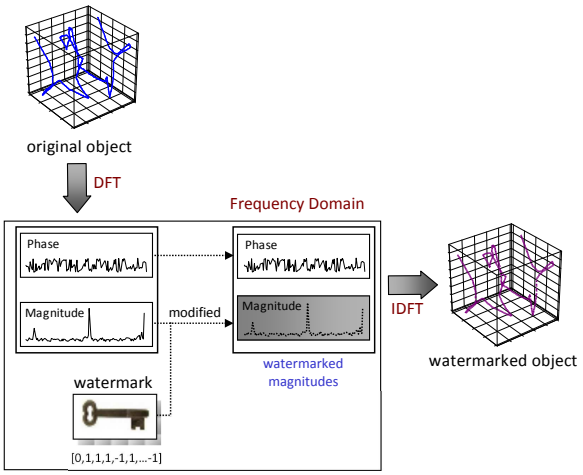

Figure 2: Overview of the right-protection process.

The robustness of the watermark embedding depends on the choice of coefficients. We embed the watermark in the coefficients that exhibit, on average over the dataset, the largest Fourier magnitudes. This makes the removal of the watermark difficult; masking it out (e.g. by noise addition) would mean that important frequencies of the dataset will be distorted. This would diminish the dataset utility. Fig. 3 shows the reconstruction of a shape from a dataset when

approximated using the highest-energy coefficients. It is apparent that the high-energy coefficients capture important characteristics of the dataset.

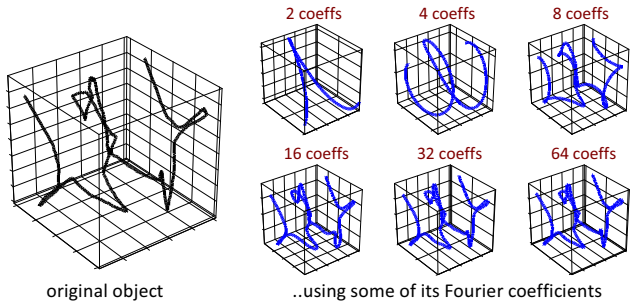

Figure 3: Object reconstruction for different number of Fourier coefficients that contain the highest energy.

Watermark Choice: Given dataset $\mathcal{D}$ and an even integer $2 \leq l \leq n$, we focus on the following class of watermarks:

DEFINITION 3.2. *(Class of watermarks $\mathcal{W}_l(\mathcal{D})$ with l non-zero elements, compatible with dataset $\mathcal{D}$) The* class of watermarks with l non-zero elements, compatible with dataset $\mathcal{D}$, *denoted by $\mathcal{W}_l(\mathcal{D})$, is the set of all $W \in \{-1, 0, +1\}^n$ that satisfy:*

$$W_j = \begin{cases} 0 & \text{if } j = 1 \quad \textit{(DC component)} \\ \{-1, 1\} & \text{if } \mu_j(\mathcal{D}) \textit{ is among the } l \textit{ largest } \mu_{i \neq 1}(\mathcal{D}) \\ 0 & \textit{otherwise} \end{cases} \tag{1}$$

as well as $\sum_{j=1}^{n} W_j = 0$.

Note that in the above definition we do not embed any part of the watermark in the first Fourier descriptor, X_1 (also called the DC component), but leave it intact. The DC component captures the center of mass of object x and is therefore highly susceptible to translational attacks. For example, if a part of the watermark were embedded on the DC component of an object then a simple translation would shift the center of mass of the object, thus rendering this part of the watermark useless without affecting the general shape of the object at all.

In summary, we embed the watermark in the magnitudes of the Fourier descriptors and leave the phases unchanged; we leave the DC component intact, and we watermark the Fourier descriptors with the largest average magnitudes.

Resilience to transformations: By construction, our right-protection mechanism provides resilience to geometric data transformations, such as rotation, translation, and scaling. Global object rotation is an intelligent attack because it distorts all coordinates of the objects, but pairwise distances remain the same. However, rotation in the frequency domain affects *only the phases* but not the magnitudes. Our watermark being embedded in the magnitude space will remain unaffected. Similarly, global translation of all objects only distorts the DC component, in which no part of the watermark was embedded. Scaling attacks can be addressed simply by normalizing all objects/sequences appropriately before watermark detection.

3.2 Watermark Detection

We measure the probability of existence of a watermark by evaluating the correlation between a tested watermark and the right-protected dataset. Measuring directly the correlation between the watermark and the magnitudes of Fourier descriptors may prove ineffective. The reason being that the original level of the average of magnitudes acts as background noise, masking the embedded watermark we seek to detect. We address this issue by explicitly recording the bias of average magnitudes before embedding the watermark, and removing it before the detection. We also record this bias vector along with the watermark W, and both are used jointly as the key.

For a dataset $\mathcal{D} = \{x^{(1)}, \ldots, x^{(|\mathcal{D}|)}\}$, we denote as $\delta_j^{x^{(i)}}$ the magnitude of the jth Fourier descriptor of object $x^{(i)}$ before watermarking. The average of the magnitudes of the jth Fourier descriptor across all objects in $\mathcal{D}$, denoted as $\mu_j(\mathcal{D})$, is given by

$$\mu_j(\mathcal{D}) := \frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \delta_j^{x^{(i)}}. \tag{2}$$

We measure the correlation between a watermarked dataset $\widehat{\mathcal{D}}$ and watermark W as follows:

$$\chi(W, \widehat{\mathcal{D}}) := \left(\frac{\mu(\widehat{\mathcal{D}}) - \mu(\mathcal{D})}{\mu(\mathcal{D})} \right)^T W,$$

where the division is *element-wise*, excluding elements where $\mu_j(\mathcal{D}) = 0$. In other words, we remove the bias of average magnitudes before computing the correlation. The scheme enables a very effective detection of the watermark. We show briefly why; after elementary algebraic calculations, the correlation between a dataset watermarked with W and any other watermark W' is reduced to:

$$\chi(W', \widehat{\mathcal{D}}) \quad = pW^T W'$$

The quantity is maximized for $W' = W$, giving $\chi(W', \widehat{\mathcal{D}}) = pl$. So, for any $W' \neq W$, $\chi(W', \widehat{\mathcal{D}}) < \chi(W, \widehat{\mathcal{D}})$.

Recall that $\mu(\mathcal{D})$ is part of the watermark (key). This results in a very secure protocol. The reason being that a malicious attacker may try to discover the embedded key by probing different watermarks. However, the correlation depends on the vector $\mu(\mathcal{D})$, which the attacker has no way of knowing without access to the non-watermarked data.

4. HIERARCHICAL CLUSTERING PRESERVATION

Hierarchical clustering (HC) builds a nested hierarchy of groups of objects according to a given distance function. This nested hierarchy is called a dendrogram (see Fig. 1). A popular method of building a dendrogram is to use an agglomerative "bottom-up" approach: each data point starts in its own cluster, and pairs of clusters are merged iteratively until a single cluster remains. There exist different functions for evaluating the distance between clusters, leading to many variants of HC approaches, such as single- or complete linkage. We explain these in detail later on, and also show how to preserve them post-watermarking.

Our right-protection scheme operates in such a way that important object distances are preserved, leading to identical clustering structure irrespective of whether the data contains a watermark or not. It is important to strike a balance between security, i.e. detectability of the watermark,

visual distortion of objects, and correctness of the clustering. Therefore, we seek to find the *maximum* embedding power p^* that does not distort the original dendrogram of the objects.

In certain cases, it may be possible to embed a watermark with high intensity and still maintain the dendrogram structure. This however, may lead to a visible distortion in an object's shape. Fig. 4 depicts how an object is distorted for increasingly larger watermark embedding powers. Therefore, in practice, we set an upper limit on the maximum allowed power, i.e., $p^* \in [p_{\min}, p_{\max}]$. In our experiments we used $p_{max} = 0.01$, therefore we allow up to 1% relative distortion. This assures that objects before and after watermarking will look virtually the same.

Figure 4: One object from a handwritten dataset (`hand` dataset) for different embedding powers of the watermark. Left: original object ($p = 0$). Middle: object distortion for $p = 0.01$. Right: object is more visibly distorted when the watermark is embedded with $p = 0.05$, corresponding to a 5% relative distortion.

Here, we study HC-preservation approaches that use the Euclidean distance between objects. We can also view pairwise relationships between objects as the edges of a complete graph. On this graph each edge $e = (x, y)$ connecting two objects x, y has weight (or length) equal to their distance $D(x, y)$. After right-protection, the edges of the dataset graph may change, because the new distance of objects x, y watermarked with power p will be $\widehat{D}_p(x, y)$. We want to ensure that important parts of the graph remain the same.

Note, also, that the function $\widehat{D}_p(x, y) = \sum_{j=1}^{n} \|(1 + pW_j)(X_j - Y_j)\|^2$ is a **parabola** with respect to the embedding power p (see Figure 7 for an example). We use this important observation to discover the appropriate lower or upper envelope of the parabolas, when we deal with single- or complete-linkage, respectively. More details on this will be provided later.

The next three definitions formalize the goal of hierarchical clustering preservation.

Definition 4.1. *(Dendrogram) A dendrogram over $(\mathcal{D}, D)$ is a triplet (T, M, l) where T is a binary rooted tree, $M : leaves(T) \to \mathcal{D}$ is a bijection, and $l : V(T) \to \{0, \ldots, h\}$ (for some integer $h \geq 0$) such that (i) for every leaf node $u \in V(T), l(u) = 0$ and (ii) if $(u, v) \in E(T)$ then $l(u) > l(v)$.*

Definition 4.2. *(Isomorphic Dendrograms) Dendrograms (T_0, M_0, l_0) and (T_1, M_1, l_1) are (order) isomorphic, if there exists a graph isomorphism $\phi : V(T_0) \to V(T_1)$ between the two trees T_0 and T_1, such that $\forall v, u \in V(T_0)$: $l_0(u) < l_0(v) \Leftrightarrow l_1(\phi(u)) < l_1(\phi(v))$.*

Definition 4.3. *(Hierarchical Clustering Preservation Problem) Given a set of objects $\mathcal{D}$ and a range of feasible powers $[p_{\min}, p_{\max}]$, find the maximal embedding power p^* such that the dendrograms computed on $\mathcal{D}$ and $\widehat{\mathcal{D}}$ are isomorphic.*

4.1 Single-Linkage Clustering

Single-linkage hierarchical clustering operates as follows: initially each object belongs to its own cluster. The two clusters with the smallest distance are merged and the distances of the newly formed cluster to the old ones are updated. The process is repeated until only one cluster remains. The linkage function L that evaluates the distance between two clusters C_0 and C_1 is given by $L(C_0, C_1) := \min D(u, v)$, where $u \in C_0, v \in C_1$. Algorithm 1 gives a high-level description of the process. Note, that the naive algorithm requires $O(n^3)$ runtime, given n objects. More efficient algorithms exist in the literature that leverage additional data structures, such as priority queues or data structures for finding the *next-best-match*. These effectively reduce the runtime to $O(n^2)$. One such algorithm is SLINK [22].

Algorithm 1 Single-Linkage Algorithm

1: **INPUTS: dataset** $\mathcal{D}$
2: **OUTPUT: Clustering** $\mathcal{C}(i)$**,** $i \in [1, |\mathcal{D}|]$
3: $\mathcal{C}(1) := \mathcal{D}$ {Each object is its own cluster}
4: **for** $i = 1 \to |\mathcal{D}| - 1$ **do** {repeat until one cluster remains}
5: Find clusters $M(i) := \{C_{m1}, C_{m2}\}$ of minimum distance $\{L(C_{m1}, C_{m2}) = \min_{C,C' \in \mathcal{C}(i)} L(C, C')\}$
6: $\mathcal{C}(i+1) = \mathcal{C}(i) \setminus M(i) \cup \{C_{m1} \cup C_{m2}\}$ {Merge clusters}
7: **end for**

To guarantee that the dataset after watermarking yields the same dendrogram, we ensure that all mergers between clusters are the same, and that they are also executed in the same order. However, it is not important which objects between the clusters lead to the merger (i.e., which edge on the distance graph is shortest), as long as the same clusters are merged for both watermarked objects and non-watermarked objects. We wish to ensure that for every feasible watermark embedding power p the shortest edge is an edge between the two merged clusters and not between any other two clusters. Formally, if C_{m1} and C_{m2} are the clusters merged in the current step then:

$$p \text{ is feasible} \quad \Leftrightarrow \quad \forall i \text{ with } M(i) = \{C_{m1}, C_{m2}\} : \quad \min_{\substack{u \in C_{m1} \\ v \in C_{m2}}} \widehat{D}_p(u, v) \leq \min_{\substack{r \in C_1 \\ s \in C_2 \\ C_1 \neq C_2}} \widehat{D}_p(r, s) \tag{3}$$

To get the feasible powers p for a merger $M(i)$, defined by inequality (3), one must determine for every $p \in [p_{min}, p_{max}]$ whether the shortest distance is between the two merged clusters C_{m1} and C_{m2} or between any other two clusters. This is illustrated in Fig. 5. The two solid lines originating from $\widehat{C}$ represent graph edges between the two clusters. The three dashed lines from $\widehat{D}$ correspond to all graph edges in between any other two clusters. In terms of inequality (3), the solid edges belong to the left-hand side and the dashed edges to the right-hand side of the inequality.

Therefore, to discover the proper power p, we consider each edge $e = (x, y)$ on the distance graph and find the ranges of power p where the examined parabola $\widehat{D}_p(e)$ is smaller than all other relevant parabolas. All power ranges for which the current edge is not the smallest should be eliminated (see Fig. 6), because otherwise the dendrogram mergers will not happen in the same order.

Analytically, this boils down to finding the lower envelope of a set of parabolas. The lower envelope EN^l (see Figure

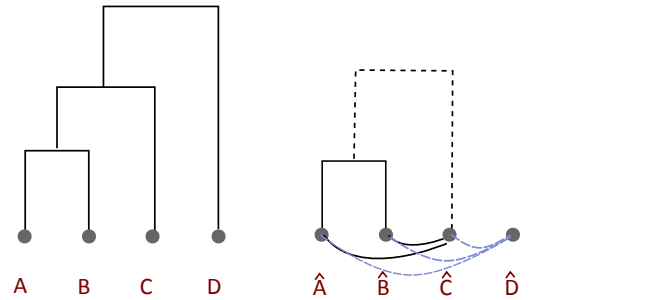

Figure 5: All relevant distances when computing the feasible powers to ensure the merger of clusters $\{\widehat{A}, \widehat{B}\}$ and $\widehat{C}$.

7) corresponds to a sequence of interleaving edges e of minimum distance and intersection points of the feasible powers p, i.e., $EN^l = (p_0 = p_{min}, e_0, p_1, e_1, p_2, e_2, ...e_m, p_{max})$, where $p_i \in [p_{min}, p_{max}]$ is an intersection point of e_i and e_{i+1}, i.e. $\widehat{D}_{p_i}(e_i) = \widehat{D}_{p_i}(e_{i+1})$. One might compute a separate lower envelope for all constraints on the left-hand side and right-hand side of inequality (3) and then compare the two envelopes to determine the feasible powers for a merger $M(i)$. It is somewhat simpler to just build a single parabola containing all parabolas from the left- and right-hand side of inequality (3). This is essentially what our Algorithm 2 does. We can achieve that very efficiently using the algorithm presented in [9] to compute the compound envelope.

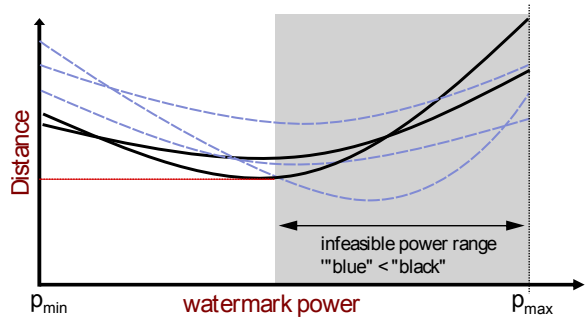

Figure 6: The feasible range of powers guaranteeing a merger of the two clusters $\{\widehat{A}, \widehat{B}\}$ and $\widehat{C}$ in Figure 5. The edges between the merged clusters are given by solid lines. All other edges are shown as dashed lines. For single-linkage clustering, an embedding power will be feasible if the shortest edge between two merged clusters is smaller than any other edge between any two other clusters.

More precisely, we iteratively compute a new envelope EN^l using all edges E in the prior envelope except the edges E' (line 8) that are between the previously merged clusters C_{m1} and C_{m2}. We go through all mergers (see line 5) in ascending order, i.e., let $M(i) = \{C_{m1}, C_{m2}\}$ be the merger currently considered. We compute the lower envelope EN^l according to [9] (line 9). Then we consider all edges $e \in E'$ that have been added to EN^l (line 10). If an edge e is not in the envelope then there are edges of smaller distance for every power $p \in [p_{min}, p_{max}]$. Thus, edge e has no influence on the feasible powers. It no longer has to be considered. If edge e is part of EN^l, say e corresponds to edge $e_j \in EN$, then the range of powers $[p_j, p_{j+1}]$ is feasible. This range $[p_j, p_{j+1}]$ is added to the set of feasible power ranges $pRanges(M(i))$. At the end of the algorithm (line 15), we compute the intersection of all $pRanges(M(i))$ to determine the maximum power p^* that yields the same mergers for watermarked and non-watermarked data.

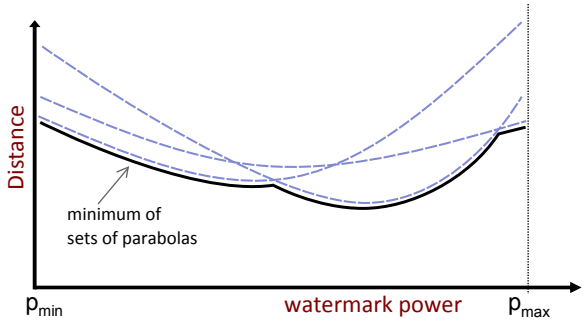

Figure 7: Dashed line shows the distance function $\widehat{D}_p(x, y)$ for an edge $e = (x, y)$. The solid line corresponds to the lower envelope, i.e., the minimum distance for every power p for a set of edges, e.g., between two clusters. For single-linkage clustering only the minimum distance is relevant.

Algorithm 2 Single-Linkage Preservation Algorithm

1: **INPUTS: dataset $\mathcal{D}$, watermark** $W \in \mathcal{W}(D), p_{min}, p_{max}$
2: **OUTPUT:** p^*
3: $den \leftarrow$ dendrogram on original data $\mathcal{D}$, e.g. using SLINK[22]
4: $E := \{(u, v) | u, v \in \mathcal{D}\}$ {set of all edges}
5: **for** $i = 1 \rightarrow |\mathcal{D}| - 1$ **do**
6: $\quad \mathcal{C} \leftarrow$ set of clusters before ith merger $M(i)$ in den
7: $\quad \{C_{m1}, C_{m2}\} \leftarrow$ clusters merged at ith merger $M(i)$ in den
8: $\quad E' \leftarrow \{(u, v) | u \in C_{m1}, C_{m2}\}$
9: $\quad EN^l :=$ lower envelope EN^l of edges E using [9]
10: $\quad$ **for all** $e_j \in (E' \cap EN^l)$ **do** {edges between the merged clusters that are in the envelope}
11: $\quad\quad pRanges(M(i)) := pRanges(M(i)) \cup [p_j, p_{j+1}]$
12: $\quad$ **end for**
13: $\quad E := E \setminus E'$ {Remove edges between merged clusters}
14: **end for**
15: $p^* = \max\{p \in \left(\bigcap_{1 \le i < |\mathcal{D}|} pRanges(M(i))\right) \bigcap [p_{min}, p_{max}]\}$

Complexity: Construction of the Single-Linkage Dendrogram requires $O(|\mathcal{D}|^2)$ time [22]. Computation of the lower envelope using $O(|\mathcal{D}|^2)$ elements takes $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$ using [9]. When computing the lower envelope EN^l, we become immediately see which edges $e \in E'$ are added between merged clusters, i.e. the intersection edges $E' \cap EN^l$ (line 10) does not add to the time complexity. Combining this, the outer for loop (line 5) takes time $O(|\mathcal{D}|^3 \log |\mathcal{D}|)$. Computing the intersections of feasible powers $pRanges(M(i))$ of individual mergers takes time $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$: There are in total (for all $M(i)$ together) at most as many edges as intervals of feasible powers $[p_j, p_{j+1}]$, i.e., $O(|\mathcal{D}|^2)$. Sorting the power intervals in each $pRanges(M(i))$ according to the left end point and merging the different sorted intervals $pRanges(M(i))$ yields the time complexity.

4.2 Complete Linkage Clustering

Just like single-linkage, complete-linkage hierarchical clustering merges two clusters according to a different linkage function L. The distance L between two clusters of nodes

C_0 and C_1 is given by the maximal distance of a pair of nodes in distinct clusters, i.e. $L(C_0, C_1) := \max_{x \in C_0, y \in C_1} D(x, y)$. Therefore, we are not interested in the edges of smallest distance between clusters but in edges of maximum distance. This changes the (number of) relevant constraints significantly. Instead of a single constraint for each merger, a constraint for every pair of clusters is required. We define $M(i)$, $\mathcal{C}(i)$, EN in the same way as in the previous section.

$$p \text{ is feasible} \quad \Leftrightarrow \quad \forall i \text{ with } M(i) = \{C_{m1}, C_{m2}\} : \quad (4)$$
$$\forall C_1, C_2 \in \mathcal{C}(i), C_1 \neq C_2 :$$
$$\max_{\substack{u \in C_{m1} \\ v \in C_{m2}}} \widehat{D}_p(u, v) \leq \max_{\substack{r \in C_1 \\ s \in C_2}} \widehat{D}_p(r, s)$$

Algorithm 3 for complete-linkage clustering preservation is more involved than Algorithm 2 for single-linkage. We have to deal with the min-max relationship due to the complete linkage criterion expressed in Inequalities (4), i.e., merge the two clusters with minimum maximum distance of two nodes. This requires to maintain a set of upper envelopes $\mathcal{EN}^u$, i.e. for each pair of clusters $C_1, C_2 \in \mathcal{C}$ there is an envelope $EN^u(C_1, C_2) \in \mathcal{EN}^u$. To get the minimal maximum distance, we compute a lower envelope EN^l for the edges in the upper envelopes.

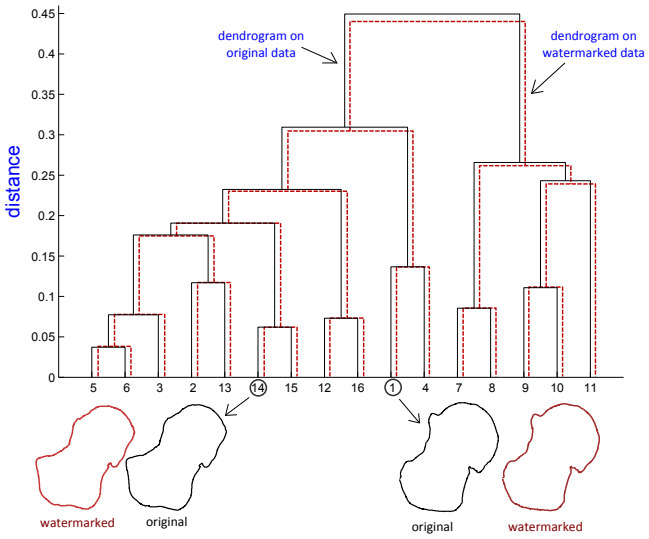

Figure 8: Preservation of complete-linkage dendrogram by watermarking (largest possible power $p = 0.0221$).

An overview of the process is given in Algorithm 3. Initially, a clustering of non-watermarked objects is computed using the CLINK algorithm [8]. We maintain a set of upper envelopes for any pair of clusters, i.e., $\mathcal{EN}^u := \{EN^u(C_1, C_2) | C_1, C_2 \in \mathcal{C}\}$. Computing the upper and lower envelope are equivalent problems, thus we can use [9] for lower envelopes with some adjustments[2]. Originally, when each object is a cluster, the upper envelope between two clusters is given by the edge between the two objects (line 4 of Algorithm 3). To get all feasible powers for merger $M(i) = \{C_{m1}, C_{m2}\}$ we compute a lower envelope EN^l consisting of the union of all edges contained in any upper envelopes $EN^u(C_1, C_2)$ of any pair $C_1, C_2 \in \mathcal{C}$. For any edge e_j between the merged clusters $\{C_{m1}, C_{m2}\}$ that is also in the lower envelope EN^l (line 12) we add the power range $[p_j, p_{j+1}]$ for which edge e_j is smallest to the feasible powers $pRanges(M(i))$. After a merger $M(i) = \{C_{m1}, C_{m2}\}$ all upper envelopes between pairs containing either C_{m1} or C_{m2} must be adjusted to keep $\mathcal{EN}^u$ up-to-date. We delete all upper envelopes of pairs involving any of two merged clusters C_{m1} or C_{m2} from $\mathcal{EN}^u$. Then a new upper envelope $EN^u(C_{m1} \cup C_{m2}, C)$ for each $C \in \mathcal{C}(i) \setminus \{C_{m1}, C_{m2}\}$ is added to $\mathcal{EN}^u$. After having gone through all mergers i in ascending order, we finally compute the intersection of all $pRanges(M(i))$ giving the maximum power p^*.

[2]e.g. in [9] Section 4, replace the lower envelope function $F(x) = \min_{i \leq n} p_i(x)$ by $F(x) = \max_{i \leq n} p_i(x)$. In Lemma 5 use $F_{i+1}(x) = \max(p_{i+1}(x), F_i(x))$

Algorithm 3 Complete-Linkage Preservation Algorithm

1: **INPUTS: dataset $\mathcal{D}$, watermark** $W \in \mathcal{W}(D), p_{min}, p_{max}$
2: **OUTPUT:** p^*
3: $den \leftarrow$ dendrogram on original data $\mathcal{D}$, e.g. using CLINK[8]
4: $EN^u(u, v) := \{p_{min}, e = (u, v), p_{max}\}$ {upper envelope of one edge}
5: $\mathcal{EN}^u := \{EN^u(u, v) | u, v \in \mathcal{D}\}$ {set of upper envelopes}
6: **for** $i = 1 \rightarrow (|\mathcal{D}| - 1)$ **do**
7: $\quad \mathcal{C} \leftarrow$ set of clusters before ith merger $M(i)$ in den
8: $\quad \{C_{m1}, C_{m2}\} \leftarrow$ clusters merged at ith merger $M(i)$ in den
9: $\quad$ /*– Compute feasible powers $pRanges(M(i))$ –*/
10: $\quad$ Compute lower envelope EN^l consisting of all edges in the upper envelopes in $\mathcal{EN}^u$ using [9]
11: $\quad E' \leftarrow \{(u, v) | u \in C_{m1}, v \in C_{m2}\}$
12: $\quad$ **for all** $e_j \in (E' \cap EN^l)$ **do** {edges between the merged clusters that are in the envelope}
13: $\quad\quad pRanges(M(i)) := pRanges(M(i)) \cup [p_j, p_{j+1}]$
14: $\quad$ **end for**
15: $\quad$ /*– Update upper envelopes $\mathcal{EN}^u$ –*/
16: $\quad$ Remove $\{EN^u(C_{m1}, C), EN^u(C_{m2}, C) | C \in \mathcal{C}\}$ from $\mathcal{EN}^u$
17: $\quad$ **for all** $C \in \mathcal{C} \setminus \{C_{m1}, C_{m2}\}$ **do**
18: $\quad\quad$ Compute envelopes for newly merged cluster $C_{m1} \cup C_{m2}$ and C to $EN^u(C_{m1} \cup C_{m2}, C)$
19: $\quad$ **end for**
20: **end for**
21: $p^* = \max\{p \in \left(\bigcap_{1 \leq i < |\mathcal{D}|} pRanges(M(i))\right) \bigcap [p_{min}, p_{max}]\}$

Complexity: Construction of the Complete-Linkage Dendrogram requires $O(|\mathcal{D}|^2)$ time [8]. Computation of the lower envelope EN^l using [9] (line 10) takes $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$. When adding an edge $e \in E'$ to envelope EN^l, we become immediately aware whether the edge becomes part of EN^l (or is too large for all powers). Thus determining the edges between merged clusters that are part of the lower envelope, i.e. $E' \cap EN^l$, (line 12) causes no costs. In iteration i of the for-loop (line 6) there are $|\mathcal{D}| - i$ clusters remaining. Computing the upper envelopes $\mathcal{EN}^u$ requires deleting $O(|\mathcal{D}|)$ envelopes and adding at most $O(|\mathcal{D}|)$ upper envelopes. More precisely, the number of edges within upper envelope $EN^u(C_{m1} \cup C_{m2}, C)$ with $C \in \mathcal{C} \setminus \{C_{m1}, C_{m2}\}$ is given by $|C_{m1} \cup C_{m2}||C|$. Thus the total number is given by $|C_{m1} \cup C_{m2}| \sum_{C \in \mathcal{C}} |C|$. Each object is in exactly one cluster, i.e. $\sum_{C \in \mathcal{C}} |C| = |\mathcal{D}|$. Therefore, $|C_{m1} \cup C_{m2}| \sum_{C \in \mathcal{C}} |C| \leq |C_{m1} \cup C_{m2}||\mathcal{D}| \leq |\mathcal{D}|^2$. The algorithm in [9] runs in $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$. Therefore the running time of the for-loop (line 6) is bounded by $O(|\mathcal{D}|^3 \log |\mathcal{D}|)$. Computing the intersections of feasible powers $pRanges(M(i))$ of individual mergers takes time $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$ as for Algorithm 2.

5. FAST ALGORITHMS

Here we derive faster variants of the previous HC-preservation algorithms. They are based on a study of the distance distortion due to the multiplicative watermarking.

THEOREM 5.1. *For any two watermarked objects $\widehat{x}, \widehat{y} \in \widehat{\mathcal{D}}$, their Euclidean distance denoted as $D_p(\widehat{x}, \widehat{y})$ is lower and upper bounded by the Euclidean distance of the non-watermarked objects $x, y \in \mathcal{D}$ as follows:*

$$(1-p)D(x,y) \leq D_p(\widehat{x},\widehat{y}) \leq (1+p)D(x,y)$$

The proof can be found in the Appendix. Fig. 9 illustrates it.

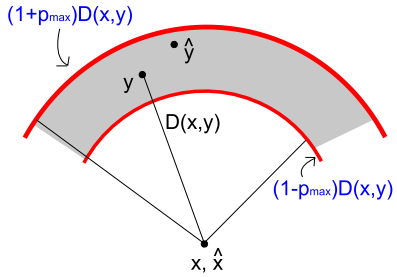

Figure 9: Two objects cannot get arbitrarily close or arbitrarily far after the watermark embedding. In this figure we represent objects as points for presentational purposes. To better illustrate the distortion bounds, objects $\widehat{x}, \widehat{y}$ are globally translated so that x and $\widehat{x}$ coincide.

We can exploit Theorem 5.1 to prune the search space while preserving the clustering structure. Namely, if for two edges e' and e'' we know that the upper bound for one of them is lower than the lower bound for the other, then those two edges will not intersect. Therefore, if for edges e' and e'' holds that:

$$(1+p_{\max})D(e') < (1-p_{\max})D(e'')$$

or

$$(1+p_{\max})D(e'') < (1-p_{\max})D(e')$$

then using Theorem 5.1 we can be sure that edge e' will be shorter than e'' (or e'' shorter than e', respectively) for any feasible power $p \in [0, p_{max}]$. Thus, there is no need to search for their intersection. After adding an edge e' to the lower envelope in Algorithm [9] (line 9 of Algorithm 2) we can therefore avoid computing the intersection of edges e' and e'' to update the envelope, i.e. figure out which edges to remove from the envelope due to the addition of e'. Indeed, the algorithm can avoid the computation for e' and e'' if:

$$\frac{1-p_{\max}}{1+p_{\max}} \leq \frac{D(e'')}{D(e')} \leq \frac{1+p_{\max}}{1-p_{\max}}$$

thus affecting significantly the number of quadratic inequalities to be solved.

In the experimental section that follows, we show that the use of lower- and upper-bounds on the watermarked distance can lead to reduction of the search space that ranges from one (1) to three (3) orders of magnitude.

6. EXPERIMENTAL EVALUATION

In this section, we evaluate the presented schemes. First, we verify that our right-protection methodology discovers the maximum watermarking power that correctly preserves the original dendrogram. Then, we compare the fast algorithms proposed in Section 5 to their exhaustive counterparts cf. Section 4, in terms of number of operations. We report major speed-ups. Finally, we assess the resilience of our scheme against a broad range of potential attacks: geometric distortions, resampling, etc.

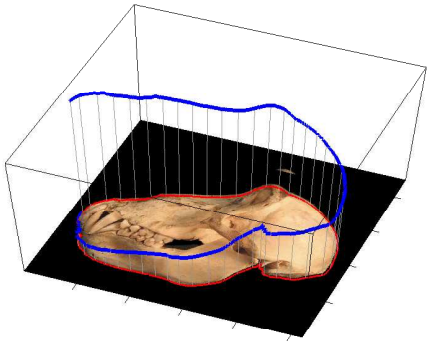

Figure 10: Image shapes can also be treated as two-dimensional sequences by extracting the perimeter of a shape.

We test our methods on datasets from various application areas: mobility data (taxi trajectories in Beijing [32, 33] and San Francisco [21]), financial data (stock prices in the NASDAQ stock market), video-tracking data, handwritten data, and image contour data from anthropology and natural sciences (the latter datasets were obtained from [15]). The characteristics of our datasets are summarized in Table 1. All experiments have been executed on a 2.16GHz Intel CPU with 3GB RAM.

Name	Data Type	Points/Object	Objects
nasdaq	Stock Prices	1000	500
taxiSF	Taxi Traj. SF	5000	516
taxiBeijing	Taxi Traj. Beijing	1411	1063
skulls	Image Contour	1500	16
fish	Image Contour	256	247
video1	Video-Tracking	1500	15
video2	Video-Tracking	500	23
hand	Handwritten	128	90

Table 1: Datasets used in the experiments.

6.1 Preservation of Distance Relations

Here we evaluate whether the single- and complete-linkage preservation algorithms discover the correct embedding power for the watermark so the dendrograms both on the original and the watermarked data remain isomorphic. This means that both the tree structure and the order of the merger points $M(i)$ are the same. A sample comparison of a dendrogram resulting from the complete linkage procedure of the original data and of the watermarked data was presented in Figure 8. In Figure 11 we show the distortion of the dendrogram that occurs when a watermark is embedded with power $p > p^*$. Indeed, even a slightly increased value of the power results in alterations in the resulting dendrogram. We report the maximal power p^* that the algorithms returned which resulted in all cases in total dendrogram preservation. The same maximal embedding

power was returned by both exhaustive and fast variants of the algorithms. Table 2 summarizes our findings.

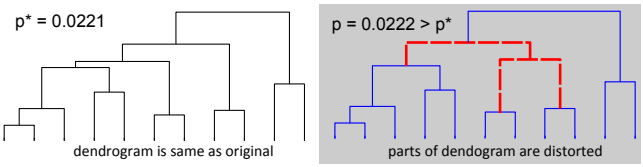

Figure 11: Dendrogram portion for the skulls dataset. Left: dendrogram for maximum discovered $p^* = 0.0221$ is same as original. Right: for even slightly larger $p = 0.0222$ the dendrogram changes.

Dataset	Single linkage p^*	Pres.	Complete linkage p^*	Pres.
nasdaq	$0.77 \cdot 10^{-4}$	100%	$0.243 \cdot 10^{-3}$	100%
taxiSF	$0.47 \cdot 10^{-4}$	100%	$0.97 \cdot 10^{-4}$	100%
taxiBeijing	$0.4 \cdot 10^{-5}$	100%	$0.28 \cdot 10^{-4}$	100%
skulls	$0.1 \cdot 10^{-1}$	100%	$0.1 \cdot 10^{-1}$	100%
fish	$0.229 \cdot 10^{-3}$	100%	$0.122 \cdot 10^{-3}$	100%
video1	$0.39 \cdot 10^{-2}$	100%	$0.1 \cdot 10^{-1}$	100%
video2	$0.1 \cdot 10^{-1}$	100%	$0.1 \cdot 10^{-1}$	100%
hand	$0.2271 \cdot 10^{-2}$	100%	$0.116 \cdot 10^{-3}$	100%

Table 2: Maximal watermarking power p^* and the percentage of dendrogram preservation.

6.2 Comparison of Algorithms

Now we compare the efficiency of the fast dendrogram preservation algorithms. With the use of the lower- and upper-bounds on the distance distortion, the fast variants can eliminate many pairs of objects from examination. This reduction leads to solving fewer quadratic inequalities in the progress of the algorithm. We record exactly how many quadratic inequalities we need to solve with each algorithm. Note, that this is also a CPU-agnostic measure and therefore does not depend on any runtime optimization.

Table 3 summarizes the results of single-linkage preservation. The pruning efficiency is reported as the ratio of the number of inequalities solved by the exhaustive algorithms, compared to the fast algorithms. The use of the bounds on the distance distortion (Section 5) results in considerable speedup in terms of solved inequalities; up to 3 orders of magnitude. Table 4 reports the results of the same experiment for the case of complete-linkage preservation.

Dataset	Single-L	Fast Single-L	Pruning
nasdaq	22,039,601	16,854	**1,308**×
taxiSF	23,032,154	5,993	**3,843**×
taxiBeijing	214,304,616	161,305	**1,329**×
skulls	770	16	**48**×
fish	2,895,694	8,425	**344**×
video1	637	23	**28**×
video2	2,311	40	**58**×
hand	129,377	976	**133**×

Table 3: Number of quadratic inequalities solved for different datasets, single linkage ($p_{min} = 0$, $p_{max} = 0.01$).

Dataset	Complete-L	Fast Complete-L	Pruning
nasdaq	21,187,434	36,280	**584**×
taxiSF	28,667,907	91,039	**315**×
taxiBeijing	322,436,544	393,970	**818**×
skulls	770	14	**55**×
fish	2,618,789	9,552	**274**×
video1	646	29	**22**×
video2	2,260	54	**42**×
hand	126,953	1,113	**114**×

Table 4: Number of quadratic inequalities solved for different datasets, complete linkage ($p_{min} = 0$, $p_{max} = 0.01$).

6.3 Resilience to Attacks

Here, we test the resiliency to potential attacks of the right-protection scheme . We right protect the dataset by inserting a watermark with the maximum allowed embedding power that preserves the dendrogram (we test on the power that preserves single-linkage clustering). We measure how detectable is the embedded watermark under a series of attacks: addition of Gaussian noise in the space and in the frequency domain, up- and down-sampling, and geometric transformations (rotation, translation, scaling). Fig. 12 depicts the ROC curves representing true- versus false-positives rates. We also report the performance of a random baseline, that randomly classifies the dataset as having or not having the watermark embedded. Due to lack of space, we only report ROC curves for four datasets. However, the results for the remaining ones were very similar.

We observe that our detection method works very effectively. It is more than 4 orders of magnitude more effective than the random baseline. In addition, the graphs exhibit a large area under-the-curve, suggesting high detectability and therefore high resilience to attacks.

7. RELATED WORK

Watermarking is a steganographic technique used for establishing the ownership, with many applications in multimedia datasets [7], such as images [18], vector graphics [19], audio [4, 27] and video [24, 34]. Multimedia watermarking focuses on the protection of a *single* object whilst minimizing visual/audible distortions of the data. In contrast, our setting operates on a collection of objects and at the same time accounts for preservation of distance relations between objects. More importantly, our scenario incorporates additional constraints in the form of guaranteeing identical outputs, after watermarking, for a class of mining and learning algorithms based on hierarchical clustering.

Privacy-preserving techniques are also related to our work, because they also alter data but enforce different constraints. To achieve privacy-preservation two research paths are typically followed: a) protection through data alteration or masking, and b) protection through dataset partition. *Data alteration* can be achieved via noise addition [14, 13], condensation [2] or data transformation [5, 20]. Similar notions have also been used for watermarking databases [3, 25]. Contrary to the above approaches, we do not attempt to reconstruct the original data distribution but work *directly* on the perturbed data, while guaranteeing preservation of distance properties on them.

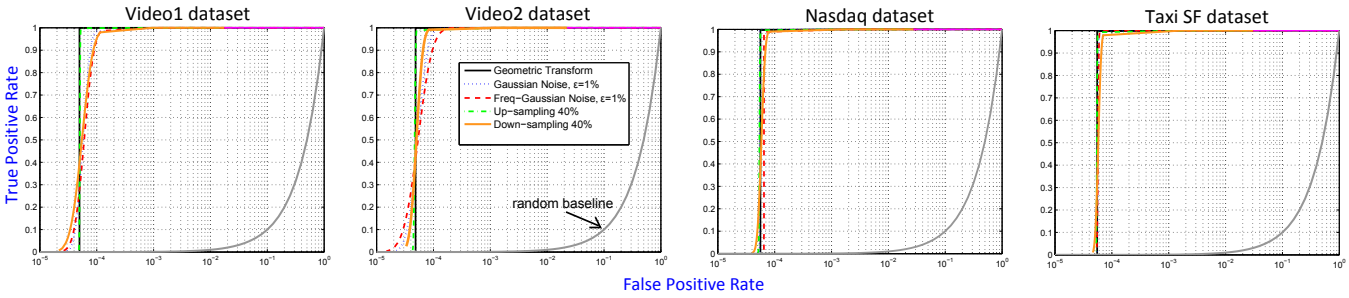

Figure 12: ROC curves for watermark detection corresponding to geometric transformations, noise addition, noise addition in the frequency domain, upsampling, and downsampling on different datasets.

Privacy-protection via *dataset partition* is achieved using horizontal or vertical data partitioning [29, 12, 30, 31]. Different portions of the data are distributed to different sites, and data exchange without leakage of private information becomes possible through cryptographic techniques (multi-party computation). The techniques in our approach are fundamentally different; the dataset is not dissected in portions, but is distributed as a whole.

Of relevance is also the work on watermarking streaming time-series [26]. Differently from our approach, they examine watermarking on a *single* numerical sequence, as opposed to considering a collection of sequences. We also aim at maintaining the original pairwise relationships and we consider resilience even under geometric data transformations (rotations, etc).

In summary, our setting presents additional challenges compared to traditional watermarking or privacy preservation techniques, because not only do we work directly on the perturbed data, but more importantly, we provide *provable guarantees* on preservation of distance properties. A right-protection scheme based on watermarking principles that preserved the Nearest Neighbor of objects was presented in [15]. We adopt the watermarking model of that work, but here we study the more elaborate case of hierarchical clustering preservation. In addition, we provide a thorough theoretical analysis on the distance distortion under a multiplicative watermarking process. We leverage the theoretical analysis, first presented in this work, to obtain fast versions of the exhaustive algorithms that we put forward, and we demonstrate substantial speed-up in all experiments, without any sacrifice in accuracy.

8. CONCLUSION

Right protection of a dataset presents an inherent engineering tradeoff: the protection should be strong enough to offer robustness of detection, but at the same time not so dominant to destroy the utility of the dataset for subsequent mining operations. We propose a watermarking technique that identifies an optimal compromise between the two conflicting factors. We design algorithms that find the maximum embedding power that guarantees preservation of hierarchical clustering operations on the modified dataset. The fast variants that we put forward can reduce the search space by more than 3000 times compared to the exhaustive algorithms, with no sacrifice in accuracy. Our analysis is generic and delivers great promise for a broader applicability for an extended class of distance-based mining operations, such as anomaly detection, classification and visualization.

9. REFERENCES

[1] The Human Genome Project. In *U.S. National Library of Medicine*, 2012.

[2] C. C. Aggarwal and P. S. Yu. A condensation approach to privacy preserving data mining. In *International Conference on Extending Database Technology*, pages 183–199, 2004.

[3] R. Agrawal and J. Kiernan. Watermarking relational databases. In *28th International Conference on Very Large Databases*, pages 155–166, 2002.

[4] P. Bassia and I. Pitas. Robust audio watermarking in the time domain. In *9th European Signal Processing Conference*, pages 25–28, 1998.

[5] K. Chen and L. Liu. Privacy preserving data classification with rotation perturbation. In *International Conference on Data Mining*, pages 589–592, 2005.

[6] E. Cope and G. Antonini. Observed correlations and dependencies among operational losses in the ORX consortium database. In *Journal of Operational Risk*, 2008.

[7] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.

[8] D. Defays. An efficient algorithm for a complete link method. *Comput. J.*, 20(4):364–366, 1977.

[9] O. Devillers and M. J. Golin. Incremental algorithms for finding the convex hulls of circles and the lower envelopes of parabolas. *Inf. Process. Lett.*, 56(3):157–164, 1995.

[10] I. M. Faniel and A. Zimmerman. Beyond the Data Deluge: A Research Agenda for Large-Scale Data Sharing and Reuse. In *Proc. of 6th International Digital Curation Conference*, 2010.

[11] R. Geambasu, S. D. Gribble, and H. M. Levy. CloudViews: Communal Data Sharing in Public Clouds. In *Proc. of HotCloud*, 2009.

[12] G. Jagannathan, K. Pillaipakkamnatt, and R. N. Wright. A new privacy-preserving distributed k-clustering algorithm. In *SIAM International Conference on Data Mining*, 2006.

[13] H. Kargupta, S. Datta, Q. Wang, and K. Sivakumar. On the privacy preserving properties of random data perturbation techniques. In *3rd IEEE International Conference on Data Mining*, pages 99–106, 2003.

[14] L. Liu, M. Kantarcioglu, and B. Thuraisingham. The

applicability of the perturbation model-based privacy preserving data mining for real-world data. In *6th IEEE International Conference on Data Mining*, pages 507–512, 2006.

[15] C. Lucchese, M. Vlachos, D. Rajan, and P. S. Yu. Rights protection of trajectory datasets with nearest-neighbor preservation. *The VLDB Journal*, 19(4):531–556, 2010.

[16] W. Ludwig and H.-P. Klenk. Overview: A phylogenetic backbone and taxonomic framework for prokaryotic systematics. In *Manual of Systematic Bacteriology*, pages 49–65, 2001.

[17] M. C. Mont, I. Matteucci, M. Petrocchi, and M. L. Sbodio. Enabling Data Sharing in the Cloud. In *HP Laboratories, Tech Report HPL-2012-22*, 2012.

[18] P. Moulin, M. E. Mihcak, and G.-I. Lin. An information-theoretic model for image watermarking and data hiding. In *IEEE International Conference on Image Processing*, pages 667–670, 2000.

[19] X. Niu, C. Shao, and X. Wang. A survey of digital vector map watermarking. *International Journal of Innovative Computing, Information and Control*, 2(6):1301–1316, 2006.

[20] S. Oliveira and O. Zaiane. Privacy preserving clustering by data transformation. In *18th Brazilian Symposium on Databases*, pages 304–318, 2003.

[21] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser. A Parsimonious Model of Mobile Partitioned Networks with Clustering. In *The First International Conference on COMmunication Systems and NETworkS (COMSNETS)*, January 2009.

[22] R. Sibson. Slink: An optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16(1):30–34, 1973.

[23] J. V. Sickle. Using Mean Similarity Dendrograms to Evaluate Classifications. In *Journal of Agricultural, Biological and Environmental Statistics*, pages 370–384, 2001.

[24] D. Simitopoulos, S. A. Tsaftaris, N. V. Boulgouris, and M. G. Strintzis. Compressed-domain video watermarking of MPEG streams. In *IEEE International Conference on Multimedia and Expo*, volume 1, pages 569–572, 2002.

[25] R. Sion, M. Atallah, and S. Prabhakar. Rights protection for relational data. *IEEE Transactions on Knowledge and Data Engineering*, 16(12):1509–1525, 2004.

[26] R. Sion, M. J. Atallah, and S. Prabhakar. Rights Protection for Discrete Numeric Streams. *IEEE Transactions on Knowledge and Data Engineering*, 18(5):699–714, 2006.

[27] M. D. Swanson, B. Zhu, A. H. Tewfik, and L. Boney. Robust audio watermarking using perceptual masking. *Signal Processing*, 66(3):337–355, 1998.

[28] A. Z. V. Zabkar. Application of End-Users Market Segmentation using Statistical Methods. In *Advances in Methodology and Statistics (19)*, 2003.

[29] J. Vaidya and C. Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 206–215, 2003.

[30] H. Yu, X. Jiang, and J. Vaidya. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. In *ACM Symposium on Applied Computing*, pages 603–610, 2006.

[31] H. Yu, J. Vaidya, and X. Jiang. Privacy-preserving SVM classification on vertically partitioned data. In *10th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 647–656, 2006.

[32] J. Yuan, Y. Zheng, X. Xie, and G. Sun. Driving with knowledge from the physical world. In *KDD*, pages 316–324, 2011.

[33] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie, G. Sun, and Y. Huang. T-drive: driving directions based on taxi trajectories. In *GIS*, pages 99–108, 2010.

[34] W. Zhu, Z. Xiong, and Y.-Q. Zhang. Multiresolution watermarking for images and video. *IEEE Transactions on Circuits and Systems for Video Technology*, 9(4):545–550, 1999.

Appendix

Proof of Theorem 5.1: The squared Euclidean distance can be expressed in the time or frequency domain as: $D^2(x,y) = \|x-y\|^2 = \|X-Y\|^2$. The same objects x and y after watermarking with power p have distance:

$$\begin{aligned}\widehat{D}_p^2(x,y) &= \|\widehat{x}-\widehat{y}\|^2 = \|\widehat{X}-\widehat{Y}\|^2 = \sum_{j=1}^{n}\|\widehat{X_j}-\widehat{Y_j}\|^2 \\ &= \sum_{j=1}^{n}\|(1+pW_j)X_j-(1+pW_j)Y_j\|^2 \\ &= \sum_{j=1}^{n}\|(1+pW_j)(X_j-Y_j)\|^2\end{aligned}$$

However, because $W_j \in \{0,1,-1\}$, we get the following bounds:

$$\begin{aligned}\sum_{j=1}^{n}\|(1-p)(X_j-Y_j)\|^2 &\le \widehat{D}_p^2(x,y) \le \sum_{j=1}^{n}\|(1+p)(X_j-Y_j)\|^2 \\ \Leftrightarrow (1-p)^2\|X-Y\|^2 &\le \widehat{D}_p^2(x,y) \le (1+p)^2\|X-Y\|^2 \\ \Leftrightarrow (1-p)^2D^2(x,y) &\le \widehat{D}_p^2(x,y) \le (1+p)^2D^2(x,y)\end{aligned}$$

□

Tightness of Distance Bounds: The bounds are tight, i.e. there exist distinct data points x and y such that $(1-p)^2D^2(x,y) = \widehat{D}_p^2(x,y)$ and points x', y' such that $(1+p)^2D^2(x',y') = \widehat{D}_p^2(x',y')$. First, we show how to construct the subspace containing points X, Y to match the lower bound. Consider two points $X = (X_0, X_1, ...)$ and $Y = (Y_0, Y_1, ...)$ such that $X_j = Y_j$ for $W_j \in \{0,1\}$. All remaining coordinates j with $W_j = -1$ are arbitrary. This gives:

$$\begin{aligned}\widehat{D}_p^2(x,y) &= \sum_{j=1}^{n}\|(1+pW_j)(X_j-Y_j)\|^2 \\ &= \sum_{j=1,W_j\in\{0,1\}}^{n}\|(1+pW_j)(X_j-Y_j)\|^2 \\ &\quad + \sum_{j=1,W_j=-1}^{n}\|(1+pW_j)(X_j-Y_j)\|^2 \\ &= (1-p)^2\sum_{j=1}^{n}\|(X_j-Y_j)\|^2 = (1-p)^2D^2(x,y)\end{aligned}$$

The points X', Y' to reach the upper bound are constructed analogously: $X'_j = Y'_j$ for $W_j \in \{0,-1\}$ and arbitrary coordinates X_j, Y_j otherwise.