On Data Publishing with Clustering Preservation

MICHAIL VLACHOS, IBM Research - Zurich
JOHANNES SCHNEIDER*, ABB Corporate Research - Baden
VASSILIOS G. VASSILIADIS, IBM Research - Zurich

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 the event of data leakage. Current right-protection technologies, however, rarely provide strong guarantees on dataset utility after the protection process. This work presents techniques that explicitly address this topic 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. Our approach considers all prevalent hierarchical clustering variants: single-, complete-, and average-linkage. We imprint the ownership in a dataset using watermarking principles, and we derive tight bounds on the expansion/contraction of distances incurred by the process. We leverage our analysis to design fast algorithms for right protection without exhaustively searching the vast design space. Finally, because the right-protection process introduces a user-tunable distortion on the dataset, we explore the possibility of using this mechanism for data obfuscation. We quantify the tradeoff between obfuscation and utility for spatiotemporal datasets and discover very favorable characteristics of the process. An additional advantage is that when one is interested in both right-protecting and obfuscating the original data values, the proposed mechanism can accomplish both tasks simultaneously.

Categories and Subject Descriptors: H.3.3 [**Information Search and Retrieval**]: Clustering; H.2.8 [**Database Management**]: Database Applications—*Data mining*; H.2.8 [**Database Management**]: Database Administration—*Security, Integrity and protection*

Additional Key Words and Phrases: Distance-Based Mining, Watermarking, Distortion Estimation, Restricted Isometry Property

ACM Reference Format:
Michail Vlachos, Johannes Schneider, and Vassilios G. Vassiliadis. 2015. On data publishing with clustering preservation. ACM Trans. Knowl. Discov. Data 9, 3, Article 23 (April 2015), 30 pages.
DOI: http://dx.doi.org/10.1145/2700403

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

*Work conducted while at IBM Research - Zurich.
The research leading to these results received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement no 259569.
Authors' addresses: M. Vlachos and V. G. Vassiliadis, IBM Research Lab, Säumerstr. 4, Rüschlikon, Switzerland; email: vva@zurich.ibm.com; J. Schneider, ABB Corporate Research, Segelhofstr. 1K, Baden, Switzerland; email: johannes.schneider@ch.abb.com.

DOI: http://dx.doi.org/10.1145/2700403

[HGP 2013], which advocated data sharing, led to "rapid scientific breakthroughs that otherwise would not have occurred."[1] Recently, even fields that viewed data sharing through a negative prism (e.g., the banking industry) have promoted the establishment of consortia to ease data exchange [Cope and Antonini 2008]. 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 [Geambasu et al. 2009; Mont et al. 2012; Faniel and Zimmerman 2011].

Data owners nonetheless need also maintain principal rights over the shared 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 twofold: 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 [Ludwig and Klenk 2001].
—Natural sciences, for the taxonomic categorization of plants or animals based on their similarity to previously categorized objects [Sickle 1997].
—Business analytics and marketing, for performing customer base segmentation and aiding the discovery of common customer profiles [Žiberna and Žabkar 2003].

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 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 HC after the right-protection is isomorphic to the one on the original data (see Figure 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 the most common HC variants: single-, complete-, and average-linkage.

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

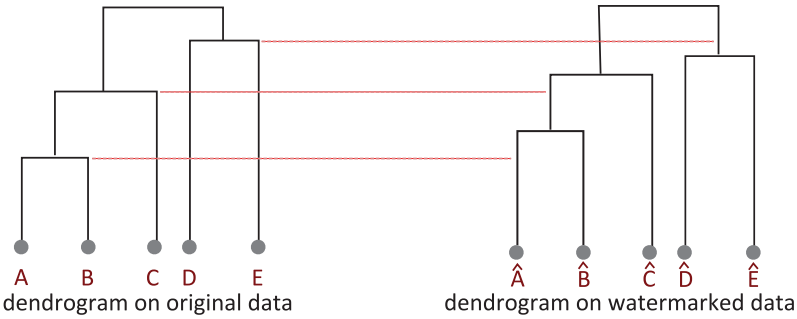

Fig. 1. Our goal is to guarantee "isomorphic" dendrograms before and after right-protection.

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 HC preservation but operate significantly faster than the exhaustive algorithms.

Our article is structured as follows: First, in Section 3.1 we describe how right-protection can be materialized via a spread-spectrum watermarking approach. We describe the threat model in Section 3.2, and we show how to detect the presence of a watermark in Section 3.3. We describe single-, complete-, and average-linkage algorithms and the necessary conditions to preserve them post-watermarking in Section 4. Subsequently, in Section 5, we study the distortion of distances due to 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.

Because watermarking introduces a tunable distortion in a dataset, we also investigate its applicability for data obfuscation (i.e., for applications where one wishes to mask the original data values (Section 6)). This could be, for example, when collecting trajectories of users (e.g., from phones or sensors), where one wishes to perform mining operations on the obfuscated user trajectories (i.e., on the data that do not reveal the actual original positions). Our methodology from conception has been designed to consider multidimensional time-series data; therefore, the applicability on spatio-temporal trajectories (among other datasets) is immediate. We evaluate the privacy-utility tradeoff curve and show the viability of the right-protection technique as a data obfuscation approach. Finally, Section 7 provides a comprehensive set of experiments that assess:

—the resilience of right-protection under common data transformations,
—the preservation of data utility using our right-protection scheme,
—the tradeoff between data utility and data distortion, and
—the computational savings introduced by our fast right-protection algorithms.

We conclude with a review of the related work and a summary of our findings.

3. RIGHT-PROTECTION THROUGH WATERMARKING

We commence by describing how watermarking techniques can embed a secret key (watermark) on a collection of objects. We demonstrate the techniques for 2D sequence data (image contours, trajectories, etc.). Subsequently, we show 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, $i^2 = -1$), $k = 1, \ldots n$. 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 that capture the perimeter coordinates of a shape.

We adapt a spread-spectrum approach [Cox et al. 1997]. 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 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 a 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. In specific:

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

and

$$\widehat{X_j} = \widehat{\delta_j} e^{\widehat{\phi_j} i} = (1 + pW_j) X_j.$$

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 (i.e., $\widehat{x} = IDFT(\widehat{X})$). An overview of the methodology described is given in Figure 2.

The robustness of the watermark embedding depends on the choice of the position of its nonzero values. We assume that the object is a member of a dataset. First, the Fourier coefficients are calculated for each object, and their magnitudes are averaged over the dataset. Then, we embed the watermark on the coefficients that exhibit some of the largest average 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 and that its utility would be diminished. Figure 3 shows the reconstruction of one object from a dataset when only some of the highest energy coefficients are used. It is apparent that the high-energy coefficients capture important characteristics of the dataset.

3.1.1. Watermark Choice. Given a 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:

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

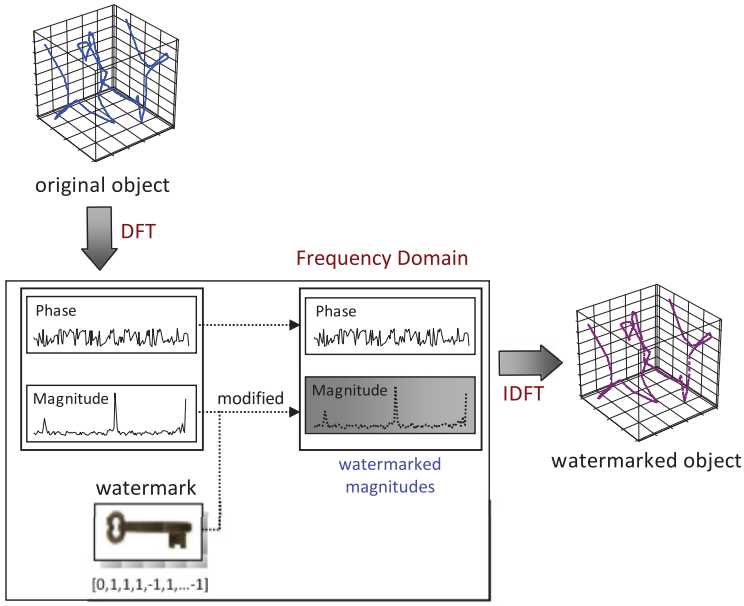

Fig. 2. Overview of the right-protection process.

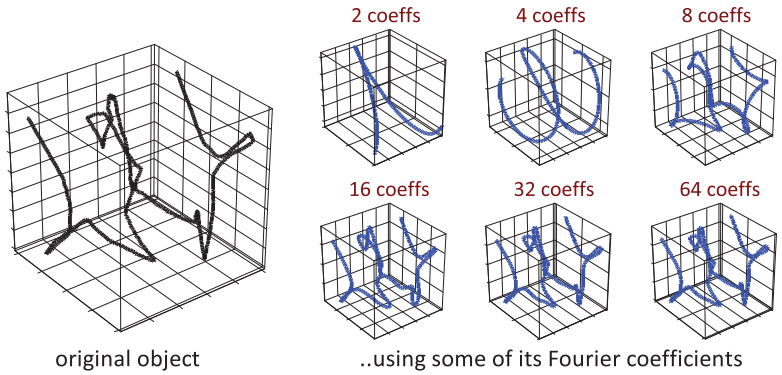

Fig. 3. Object reconstruction for different number of Fourier coefficients that contain the highest energy.

where $\mu_k(\mathcal{D}), k = 2, \ldots, n$ is the average of the kth magnitudes over the database (formally presented in Equation (2)), and $\sum_{j=1}^{n} W_j = 0$.

Note that in this 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.

3.1.2. 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. Threat Model

A potential attacker does not have access to the original data, but has access to the watermarked data and may modify them in an effort to remove the watermark. The attacker may transform the data to the extent that their utility is not severely compromised (otherwise the attack is obvious). We assume that an attacker: (a) is knowledgeable of the algorithm but not of the secret key; and (b) may distort the data using geometric transformations, noise addition (in both time and frequency domains), or data transcription (e.g., upsampling or downsampling).

3.3. 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 because 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 $\delta_j^{(k)}$ the magnitude of the kth Fourier descriptor of object $x^{(k)}$ 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

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

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:

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

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 because a malicious attacker may try to discover the embedded key by probing different

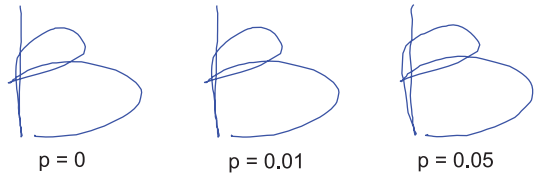

Fig. 4. One object from a handwritten dataset (hand dataset) for different embedding powers of the watermark. Left: original object ($p = 0$). Middle: object distortion for $p = 0.01$. Right: object is more visibly distorted when the watermark is embedded with $p = 0.05$, corresponding to a 5% relative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

watermarks. However, the correlation depends on the vector $\mu(\mathcal{D})$, which the attacker has no way of knowing without access to the nonwatermarked data.

Because the dataset may undergo a number of potential modifications (smoothing, sampling, etc.), the correct key may not exhibit a perfect correlation with the dataset. The watermark detector can be made more robust by using an appropriate threshold, above which the key is considered as embedded. The threshold value can be derived via a learning process by computing the correlation histograms when probing with both correct and incorrect watermark keys. These distributions are Gaussians, and the determining threshold can be set to the midpoint of these distributions. For additional details on this process, consult also Lucchese et al. [2010].

3.3.1. Determining Ownership. To determine whether a user is the owner of a dataset, both dataset and user key are required. Using the watermark detection process just detailed, the test will only be successful with the owner's original key (i.e., the embedded watermark).

4. HIERARCHICAL CLUSTERING PRESERVATION

HC builds a nested hierarchy of groups of objects according to a given distance function. This nested hierarchy is called a *dendrogram* (see Figure 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-, complete-, or average-linkage. We explain these in detail later and also show how to preserve them postwatermarking.

Our right-protection scheme operates in such a way that important object distances are preserved, leading to identical clustering structure before and after watermarking. It is important to strike a balance among 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. Figure 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.

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.

An important observation is that the Euclidean distance with respect to the watermark embedding power p is a **parabola**: $\widehat{D}_p^2(x, y) = \sum_{j=1}^{n} \|(1 + pW_j)(X_j - Y_j)\|^2$. For illustrations, see Figures 6 and 7. We capitalize on this observation to discover lower and upper bounds on the watermark and HC distortion, which will be modeled as lower or upper envelopes of the parabolas. These are explained in detail in the upcoming sections.

The next three definitions formalize the goal of HC 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$(T) \rightarrow \mathcal{D}$ is a bijection, and $l : V(T) \rightarrow \{0, \ldots, h\}$ (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 $l(u) > l(v)$). By $E(T)$ and $V(T)$, we denote the set of the edges and the set of the vertices, respectively.

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 $\phi : V(T_0) \rightarrow V(T_1)$ 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. (HC 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.

Throughout this article, we use the notion of the distance between two clusters defined in three different ways depending on the method used.

Definition 4.4. (Distance between Two Clusters) Given two sets of objects (clusters), C_1 and C_2, we define:

—$L^s(C_1, C_2) := \min_{u \in C_1, v \in C_2} D(u, v)$,
—$L^c(C_1, C_2) := \max_{u \in C_1, v \in C_2} D(u, v)$,
—$L^a(C_1, C_2) := 1/(|C_1| \cdot |C_2|) \sum_{x \in C_1, y \in C_2} D(x, y)$.

To notate the same functions after watermarking with power p, the index p is added, e.g., $L_p^a(C_1, C_2)$.

4.1. Single-Linkage Clustering

Single-linkage HC 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. Between two clusters, the distance function L^s is used. 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 [Sibson 1973].

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 nonwatermarked objects. We wish to ensure that for every feasible watermark embedding power p, the shortest

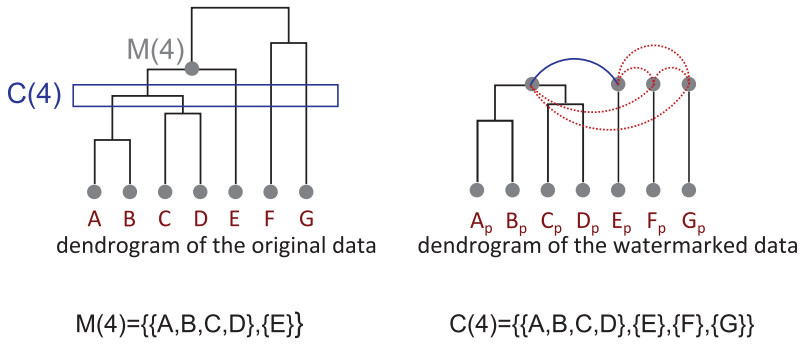

Fig. 5. All relevant distances when computing the merging M(4) of the clusters C(4). The feasible powers p are those which ensure that the blue distance is shorter than all the dotted red ones.

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** $i = 1 \rightarrow |\mathcal{D}| - 1$ **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**

edge lies 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:

$$p \text{ is feasible } \Leftrightarrow \forall i \text{ with } M(i) = \{C_{m1}, C_{m2}\}, \quad L_p^s(C_{m1}, C_{m2}) \leq \min_{\substack{C_1, C_2 \in C(i) \\ C_1 \neq C_2}} L_p^s(C_1, C_2). \tag{3}$$

To get the feasible powers p for a merger $M(i)$, defined by inequality in Equation (3), one must determine for every $p \in [p_{min}, p_{max}]$ whether the smaller distance value occurs for the two merged clusters C_{m1} and C_{m2} and not for any other two clusters. This is illustrated in an example depicted in Figure 5. Following Figure 5, we focus on $M(4)$ that merges the clusters $C_{m1} = (\{A\} \cup \{B\}) \cup (\{C\} \cup \{D\}) = \{A, B, C, D\}$ and $C_{m2} = \{E\}$ of the original data. On the righthand side of Figure 5, the distances involved are shown. Thus, a feasible power p should ensure that the "solid" (blue) distance edge (remember that this is the minimum of all the involved distances) of the clusters C_{m1} and C_{m2} is less than any "dotted" (ref) distance edge of any other combination. Here, all the lines represent the possible graph edges. In terms of inequality Equation (3), the solid edge belongs to the left-hand side and the dotted 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 (see Figure 6). For a certain p, the value of the distance function of the left side of Equation (3) is the minimum of the values of the "solid" (blue) parabola. All the "dotted" (red) parabolas correspond to edges involved in the right side of Equation (3). The feasible values of the power p are those that keep the value of the examined linkage function less than the values of the other ("dotted") relevant parabolas. All the other values of the power should be eliminated because, otherwise, the resulting dendrogram would not be isomorphic to the original one.

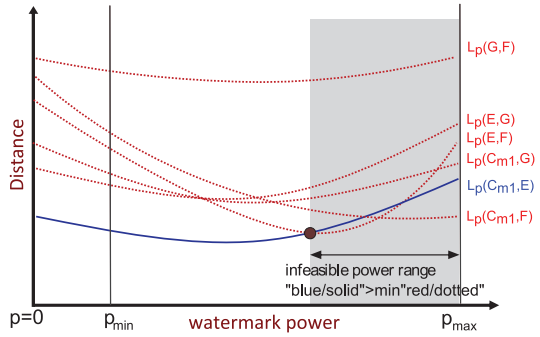

Fig. 6. The feasible range of powers guaranteeing a merger of the two clusters $\{A_p, B_p, C_p, D_p\}$ and $\{E_p\}$ in Figure 5. The edges between the merged clusters are given by solid lines. All other edges are shown as dotted.

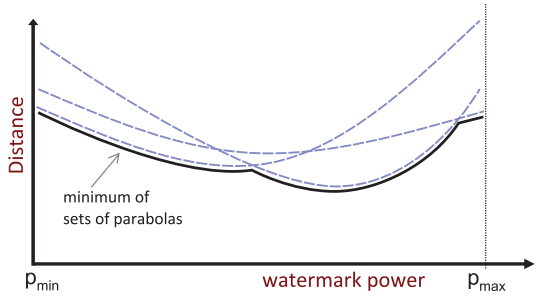

Fig. 7. Dashed line shows the distance function $\widehat{D}_p(x, y)$ for an edge $e = (x, y)$. The solid line corresponds to the lower envelope; that is, the minimum distance for every power p for a set of edges (e.g., between two clusters). For single-linkage clustering only the minimum distance is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

Analytically, this resorts to finding the lower envelope of a set of parabolas. The lower envelope EN^l (see Figure 7) corresponds to a sequence of interleaving edges e of minimum distance and intersection points of the feasible powers p; that is,

$$EN^l = (p_0 = p_{min}, e_0, p_1, e_1, p_2, e_2, \ldots e_m, p_{max}),$$

where $p_i \in [p_{min}, p_{max}]$ is an intersection point of the parabolas for 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 Equation (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 Equation (3). We can achieve that very efficiently using the algorithm presented in Devillers and Golin [1995] to compute the compound envelope.

More precisely, using Algorithm 2, 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 Devillers and Golin [1995] (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 nonwatermarked data.

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 [Sibson 1973]
4: $E := \{(u, v) | u, v \in \mathcal{D}\}$ {set of all edges}
5: **for** $i = 1 \rightarrow |\mathcal{D}| - 1$ **do**
6: $\mathcal{C} \leftarrow$ set of clusters before ith merger $M(i)$ in den
7: $\{C_{m1}, C_{m2}\} \leftarrow$ clusters merged at ith merger $M(i)$ in den
8: $E' \leftarrow \{(u, v) | u \in C_{m1}, v \in C_{m2}\}$
9: $EN^l :=$ lower envelope EN^l of edges E using Devillers and Golin [1995]
10: **for all** $e_j \in (E' \cap EN^l)$ **do** {edges between the merged clusters that are in the envelope}
11: $pRanges(M(i)) := pRanges(M(i)) \cup [p_j, p_{j+1}]$
12: **end for**
13: $E := E \setminus E'$ {Remove edges between merged clusters}
14: **end for**
15: $p^* = \max\{p \in (\bigcap_{1 \le i < |\mathcal{D}|} pRanges(M(i))) \bigcap [p_{min}, p_{max}]\}$

Complexity: Construction of the single-linkage dendrogram requires $O(|\mathcal{D}|^2)$ time [Sibson 1973]. Computation of the lower envelope using $O(|\mathcal{D}|^2)$ elements takes $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$ using Devillers and Golin [1995]. When computing the lower envelope EN^l, we are aware of which edges $e \in E'$ are added between merged clusters; that is, the intersection of edges $E' \cap EN^l$ (Line 10) does not add to the time complexity. Combining this, the outer for loop (Line 5) runs at $O(|\mathcal{D}|^3 \log |\mathcal{D}|)$. Computing the intersections of feasible powers $pRanges(M(i))$ of individual mergers costs $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$: 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

Now, we examine how to derive the maximum watermark embedding power p^* that maintains postwatermarking the result of complete-linkage HC. Our approach is similar as that use din the single-linkage case: We need to discover which watermark powers will not violate the original merging order of the clusters in the dendrogram. The only difference is that under complete-linkage two clusters are merged according to a different linkage function L^c. Under single-linkage, cluster distance depends on the minimum object distance, whereas under complete-linkage, cluster distance is evaluated as the maximum distance of their respective objects. For each step, the two clusters with the smallest maximum distance are merged. Therefore, we have a min max relationship: Between each pair of clusters we consider the maximum distance; thus, we use all edges (object distances) between them to compute an upper envelope (maximum distance parameterized by p). Out of these envelopes, we are interested in the minimum distance of any upper envelope between any pair of clusters. This is achieved by computing a lower envelope taking into account all upper envelopes. In what follows, we describe this intuition more formally.

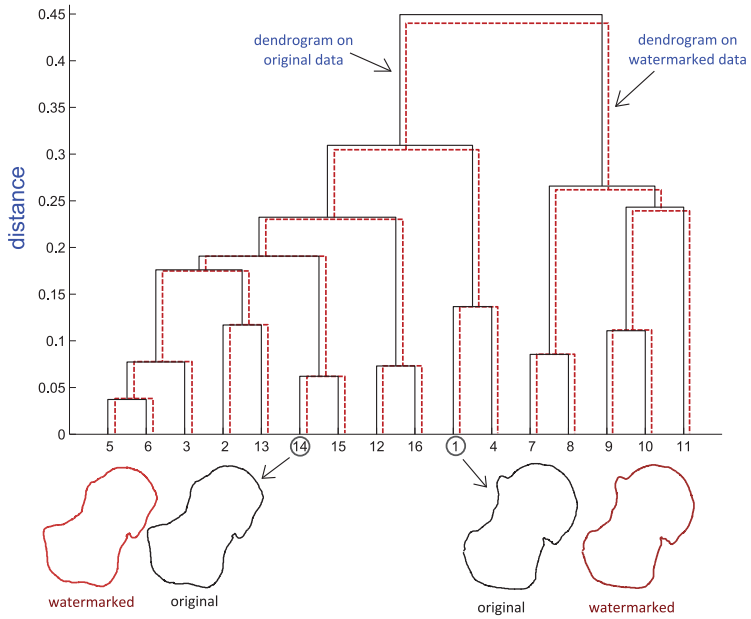

Fig. 8. Preservation of complete-linkage dendrogram by watermarking (largest possible power $p = 0.0221$).

We define $M(i)$, $\mathcal{C}(i)$, EN in the same way as in the previous section. Thus, if C_{m1} and C_{m2} are the clusters merged in the current step, then:

$$p \text{ is feasible } \Leftrightarrow \forall i \text{ with } M(i) = \{C_{m1}, C_{m2}\}, \quad L_p^c(C_{m1}, C_{m2}) \leq \min_{\substack{C_1, C_2 \in C(i) \\ C_1 \neq C_2}} L_p^c(C_1, C_2), \tag{4}$$

We have to deal with the min-max relationship due to the complete linkage criterion expressed in inequalities in Equation (4); that is, merge the two clusters with minimum maximum distance of two nodes. This requires us 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 minimum maximum distance, we compute a lower envelope EN^l for the edges in the upper envelopes.

An overview of the process is given in Algorithm 3. Initially, a clustering of non-watermarked objects is computed using the CLINK algorithm [Defays 1977]. 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 Devillers and Golin [1995] 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 EN^u up to date.

[2]For example, in Devillers and Golin [1995] 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 $F_{i+1}(x) = \max(p_{i+1}(x), F_i(x))$.

We delete all upper envelopes of pairs involving any of two merged clusters C_{m1} or C_{m2} from 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 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^*.

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 [Defays 1977]
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: $\mathcal{C} \leftarrow$ set of clusters before ith merger $M(i)$ in den
8: $\{C_{m1}, C_{m2}\} \leftarrow$ clusters merged at ith merger $M(i)$ in den
9: /*– Compute feasible powers $pRanges(M(i))$ –*/
10: Compute lower envelope EN^l consisting of all edges in the upper envelopes in EN^u using Devillers and Golin [1995]
11: $E' \leftarrow \{(u, v) | u \in C_{m1}, v \in C_{m2}\}$
12: **for all** $e_j \in (E' \cap EN^l)$ **do** {edges between the merged clusters that are in the envelope}
13: $pRanges(M(i)) := pRanges(M(i)) \cup [p_j, p_{j+1}]$
14: **end for**
15: /*– Update upper envelopes $\mathcal{EN}^u$ –*/
16: Remove $\{EN^u(C_{m1}, C), EN^u(C_{m2}, C) | C \in \mathcal{C}\}$ from EN^u
17: **for all** $C \in \mathcal{C} \setminus \{C_{m1}, C_{m2}\}$ **do**
18: Compute envelopes for newly merged cluster $C_{m1} \cup C_{m2}$ and C to $EN^u(C_{m1} \cup C_{m2}, C)$
19: **end for**
20: **end for**
21: $p^* = \max\{p \in (\bigcap_{1 \le i < |\mathcal{D}|} pRanges(M(i))) \bigcap [p_{min}, p_{max}]\}$

Complexity: Construction of the complete-linkage dendrogram requires $O(|\mathcal{D}|^2)$ time [Defays 1977]. Computation of the lower envelope EN^l using Devillers and Golin [1995] (Line 10) costs $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$. When adding an edge $e \in E'$ to envelope EN^l, we immediately know 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 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, $|C_{m1} \cup C_{m2}| \sum_{C \in \mathcal{C}} |C| \le |C_{m1} \cup C_{m2}||\mathcal{D}| \le |\mathcal{D}|^2$. The algorithm in Devillers and Golin [1995] 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.

4.3. Average Linkage Clustering

To address cluster preservation under average-linkage we consider the modifications that we have to make due to the different cluster distance function. For average-linkage, cluster distance is determined by the function L^a. By definition, the average distance L^a between two clusters is the sum of the distances of all edges divided by the number of edges. Summing up parabolas and dividing them by a (fixed) number still

yields a parabola. Therefore, the function L^a between any pair of clusters C_1 and C_2 is a parabola of the form $ap^2 + bp + c$ for values a, b, c depending on the parabolas of the edges between C_1 and C_2. To discover the maximal embedding power p between any two clusters, we compute an envelope for the parabolas (distance parameterized by p) between any pair of clusters, and there is one parabola per cluster pair.

The constraints and the definitions of $M(i)$, $\mathcal{C}(i)$ can be stated in the same way as before. To compute the feasible set of powers p for a merger $M(i) = \{C_{m1}, C_{m2}\}$, formally,

$$p \text{ is feasible } \Leftrightarrow \forall i \text{ with } M(i) = \{C_{m1}, C_{m2}\}, \quad L_p^a(C_{m1}, C_{m2}) \leq \min_{\substack{C_1, C_2 \in C(i) \\ C_1 \neq C_2}} L_p^a(C_1, C_2), \tag{5}$$

requires us to compute the intersections of $L_p^a(C_{m1}, C_{m2})$ with all the parabolic distances of the other pairs of clusters C_1, C_2.

ALGORITHM 4: Average-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 Murtagh [1984]
4: **for** $i = 1 \rightarrow (|\mathcal{D}| - 1)$ **do**
5: $\mathcal{C} \leftarrow$ set of clusters before ith merger $M(i)$ in den
6: $\{C_{m1}, C_{m2}\} \leftarrow$ clusters merged at ith merger $M(i)$ in den
7: /*– Compute feasible powers $pRanges(M(i))$ –*/
8: Compute lower envelope EN^l consisting of all edges, i.e., one for each pair (C_0, C_1) with $C_0, C_1 \in \mathcal{C}$ using Devillers and Golin [1995]
9: **for all** $e_j \in (e' = (C_{m1}, C_{m2}) \cap EN^l)$ **do** {edges between the merged clusters that are in the envelope}
10: $pRanges(M(i)) := pRanges(M(i)) \cup [p_j, p_{j+1}]$
11: **end for**
12: **end for**
13: $p^* = \max\{p \in (\bigcap_{1 \leq i < |\mathcal{D}|} pRanges(M(i))) \bigcap [p_{min}, p_{max}]\}$

Complexity: Construction of the average-linkage dendrogram requires $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$ time [Murtagh 1984]. Computing the linkage function $L_p^a(C_0, C_1)$ in terms of p between two clusters C_0 and C_1 requires time $O(|C_0||C_1|)$. Before the first merger, the computation of all pairwise distances between clusters takes time $O(|\mathcal{D}|^2)$. The time complexity for computation of the linkage function $L_p^a(C_{m1} \cup C_{m2}, C)$ between a merged cluster $C_{m1} \cup C_{m2}$ and all other clusters $C \in \mathcal{C} \setminus \{C_{m1}, C_{m2}\}$ can be bounded as follows: $|C_{m1} \cup C_{m2}| \sum_{C \in \mathcal{C} \setminus \{C_{m1}, C_{m2}\}} |C|$. Since each object is in exactly one cluster, we have $\sum_{C \in \mathcal{C}} |C| \leq |\mathcal{D}|$ and also $|C_{m1} \cup C_{m2}| \leq |\mathcal{D}|$, yielding $|\mathcal{D}|^2$ time to compute the linkage function. Overall, there are $O(|\mathcal{D}|)$ mergers. Thus, the distance computations take $O(|\mathcal{D}|^3)$ time.

Computation of the lower envelope EN^l using Devillers and Golin [1995] (Line 8) takes $O(|\mathcal{D}|^2 \log |\mathcal{D}|)$. To determine the powers between merged clusters $\{C_{m1}, C_{m2}\}$, we intersect $e' = \{C_{m1}, C_{m2}\} \cap EN^l$ (Line 9); that is, we go through the edges $e \in EN^l$ sequentially and compare each edge $e \in EN^l$ with e', which requires time at most $O(|EN^l|) \in O(|\mathcal{D}|)$ since each cluster consists only of a single parabola. In iteration i of the for-loop (Line 4) there are $|\mathcal{D}| - i$ clusters remaining. Therefore, the running time of the for-loop (Line 4) 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.

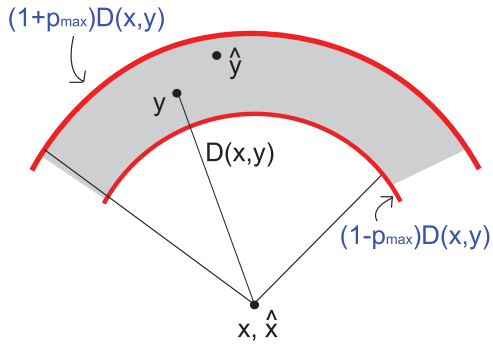

Fig. 9. Two objects cannot get arbitrarily close or far after the watermark embedding. In this figure, we represent objects as points for presentational purposes. To better illustrate the distortion bounds, objects $\widehat{x}, \widehat{y}$ are globally translated so that x and $\widehat{x}$ coincide.

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:*

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

Figure 9 illustrates the proof.

PROOF. Using Parseval's theorem, 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:

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

5.1. Tightness of Distance Bounds

The bounds are tight; that is, 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 $X = (X_0, X_1, \ldots)$ and $Y = (Y_0, Y_1, \ldots)$ such that $X_j = Y_j$ for

$W_j \in \{0, 1\}$. All remaining coordinates j with $W_j = -1$ are arbitrary. This gives:

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

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.

The inequality of Theorem 5.1 essentially hints on that fact that if we already have an index that is built on the original non-right-protected data, we can use it to speed up the search process because we can bound the watermarked distance as a parameter of the original distance (which we can get using the index) and the embedding power p.

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:

$$(1 + p_{\max})D(e') < (1 - p_{\max})D(e''), \text{ or } (1 + p_{\max})D(e'') < (1 - p_{\max})D(e');$$

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. This reduces significantly the number of quadratic inequalities to be solved. After adding an edge e' to the lower envelope in Algorithms 2, 3 and 4 [Devillers and Golin 1995] (e.g., 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'). For the example of Figure 5, after the computation of the lower envelopes (Figure 6), one can avoid solving the quadratic inequality between $L_p(G, F) = \hat{D}_p(G, F)$ and $L_p(C_{m1}, F)$ since the lower bound of the first is greater than the upper bound of the latter. An illustration of this is given in Figure 10. In the experimental section that follows, we show that the use of lower and upper bounds on the watermarked distance can lead to a significant reduction of the search space that ranges from one to three orders of magnitude.

5.2. Discussion

The foregoing analysis assumed the Euclidean distance for object comparison. However, it is easily transferable for other commonly used distance (or similarity) functions, such as correlation or cosine similarity, which can be relegated to some normalized Euclidean distance calculation (see, e.g., Qian et al. [2004] and Borgatti [2007]).

When dealing with nonlinear distance functions (e.g., Dynamic Time Warping [DTW]), the existence of a closed-form analysis of the distance distortion due to watermarking is largely unknown and represents an open topic for research. However, because there are ways to upper/lower bound DTW using Euclidean distance on a transformed space (space-bounding envelope [Vlachos et al. 2003]), such analysis may be viable but resides outside the scope of the current work.

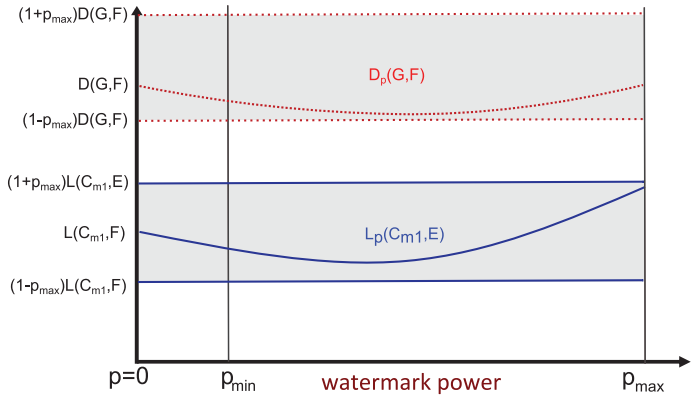

Fig. 10. An example of the pruning achieved using the lower/upper bounds. We illustrate the case where the upper bound of the edges between the merged clusters is lower than the edge of one of the other possible pairs (also refer to Figures 5 and 6).

6. WATERMARKING AS PERTURBATION MECHANISM

Embedding a watermark in the dataset alters the original values, so it also provides a data obfuscation method. Therefore, our methodology could also be applicable in cases where one is interested in masking the original values but still wants to be able to mine the resulting dataset [Xue et al. 2013]. Our technique inherently considers multidimensional sequences, so it can be used without alterations for data obfuscation in spatiotemporal datasets. Today, such datasets are collected very easily from phones, GPS devices, and the like that continuously collect the movements of individuals and vehicles. Analysis of such trajectory datasets holds great importance for many applications [Parent et al. 2013], such as traffic analysis and optimization [Duffield and Grossglauser 2001] and city planning [Song et al. 2010]. However, such datasets may need to be accordingly altered before data release, for example, in cases when we wish to mask the exact location of individuals [Wicker 2012]. Because the watermarking process adds noise to the original data trajectory, the original values are not revealed, but the trajectory still preserves to a high degree its overall pattern. So, it is still useful for further analysis. The degree of distortion is controlled by the embedding power p of the watermark.

Our methodology discovers the maximum power p^* that guarantees preservation of HC after right-protection. The power p^* is the one that maximizes obfuscation and, at the same time, offers 100% utility preservation. If one wishes to offer better obfuscation, then the watermark can be embedded with higher intensity. Then, of course, utility does not remain at the 100% level, but this is a tradeoff that one has to pay for the benefit of increased data masking. Note that a similar tradeoff also exists between utility and data privacy/anonymization [Ghinita et al. 2009], a well-understood concept in privacy-preserving data mining [Li and Li 2009; Aggarwal and Yu 2008] and privacy-preserving data publishing [Fung et al. 2010; Xue et al. 2011].

However, there is an upside to the reduction in data utility. Because the watermark is embedded with stronger intensity, it can be better detected under more destructive transformational attacks on the right-protected dataset. In the experimental section, we show that the watermark embedded with power p^* is robustly detected under a variety of attacks. Now, if the watermark is embedded with power $p > p^*$, then naturally the detection is even more pronounced, or, in other words, the watermark can be retrieved under more destructive data attacks (e.g., increased down-sampling rates, removal of more objects, and so on).

Therefore, when one is interested in both right-protecting and obfuscating a dataset, the proposed methodology can achieve both simultaneously. The advantage of this is not only that it is simpler, but, intuitively, it suggests that less noise has to be added to ensure the same degree of dataset utility. As an example, assume that an institution wishes to share its dataset. The requirements are to both alter the original values and right-protect the dataset with at most a 5% relative distortion on the dataset. When data obfuscation and watermarking are applied as separate processes, then part of this 5% noise is allocated for the obfuscation process and part for the right-protection process. Therefore, the watermark is always embedded with weaker intensity when the two processes are separated.

In summary, the advantages of using watermarking methodology as a data obfuscation process are the following:

(1) The right-protection is accomplished simultaneously without additional modification.
(2) Right-protection is even more robust when the watermark is embedded with higher intensity.

Here, we examine in more detail the metrics that we use to quantify data utility after the right-protection, as well as the metric for distortion/obfuscation. We also quantify the degree of the distortion via the parameters of the right-protection mechanism.

6.1. Utility Metric

To assess the utility of the data, we measure how well the HC clustering is preserved. We quantify the similarity between two dendrograms using the confusion matrices when forming $k = 2, \ldots |\mathcal{D}| - 1$ (i.e., when the dendrogram is cut at different hierarchical levels [Fowlkes and Mallows 1983]). Assume that, at a particular level of the dendrogram, k clusters are formed. The $k \times k$ confusion matrix at position i, j has a value m_{ij}, indicating the number of objects in common between the ith cluster of the first dendrogram and the jth cluster of the second. So, at that particular level, we define the association between the two clusterings to be:

$$B_k = T_k \sqrt{P_k Q_k}, \tag{6}$$

where

$$T_k = \sum_{i=1}^{k} \sum_{j=1}^{k} m_{ij}^2 - |\mathcal{D}|,$$

$$P_k = \sum_{i=1}^{k} \left(\sum_{j=1}^{k} m_{ij} \right) - |\mathcal{D}|,$$

$$Q_k = \sum_{j=1}^{k} \left(\sum_{i=1}^{k} m_{ij} \right) - |\mathcal{D}|.$$

B_k depends both on topology and labeling of the objects and takes a value between zero to one, the latter value when the confusion matrix has exactly k nonempty cells (i.e., the two clusterings have identical labels and topologies). In Figure 11, we provide an example of this utility metric.

We have to evaluate the clustering quality at all $|\mathcal{D}| - 2$ levels of the dendrogram (the top level consisting of one cluster is the same), so the dendrogram similarity $S(p)$ for

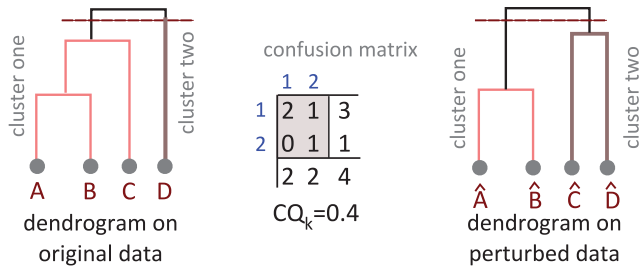

Fig. 11. Two dendrograms and their similarity when cutting the dendrogram to form $k = 2$ clusters.

a given watermark power is the average of all the measures of association B_k; that is,

$$S(p) = \sum_k B_k/(|\mathcal{D}| - 2). \tag{7}$$

We use this metric to compare the clustering using the original data $\mathcal{D}$ and the clustering obtained using the right-protected data $\hat{\mathcal{D}}$. A value of 1 suggests identical clusters of both datasets at every level of the hierarchy.

6.2. Obfuscation Metric

A watermarking scheme such as the one used by our right-protection methodology distorts (obfuscates) the original data. The obfuscation can be imperceptible (low watermark embedding powers) or perceptible (high watermark embedding powers). When we are interested primarily in data obfuscation, we can afford we embed a watermark with stronger intensity. We quantify the obfuscation using the relative data distortion r before and after watermarking:

Definition 6.1. (Relative Data Distortion) The relative data distortion r measures the average relative error of a distorted data point $\hat{x} \in \hat{\mathcal{D}}$ relative to the original data point $x \in \mathcal{D}$:

$$Dst(\mathcal{D}, \hat{\mathcal{D}}) := \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \frac{|x - \hat{x}|}{|x|} \tag{8}$$

Such a measure has also been used in other previous work on trajectories (e.g., Xue et al. [2011]; Abul et al. [2008]). Essentially, this is a measure of *information loss*. Other metrics used for location obfuscation can be found in Ardagna et al. [2007].

Note that this distortion metric could also potentially be used as a generic proxy for measuring an upper bound on data *privacy*. We mention upper bound because, to quantify data privacy, one should assume extra information about the application at hand and about the background knowledge of an attacker [Wong et al. 2011]. For example, a recent study notes that knowledge of the home and work location can almost uniquely identify an individual [Golle and Partridge 2009]. Therefore, in a dataset of spatiotemporal trajectories recording phone locations of users, the start (home) and end (work) of the trajectory are particularly important. If the obfuscation mechanism does not distort the beginning and end of the path in such a dataset, given background information, privacy can still be compromised irrespective of the amount of noise that is added. Other studies have also shown that knowledge of subparts of a location-based trajectory may similarly function as quasi-identifiers [Terrovitis and Mamoulis 2008]. Therefore, privacy really depends on the application and on the background information that an attacker holds. Popular models for data privacy include k-anonymity [Nergiz et al. 2009] and differential-privacy [Chen et al. 2011],

among others. The presented obfuscation metric may provide useful indications on the maximal and idealized degree of privacy that could be obtained using the presented right-protection technique. However, here, we do not make claims on the privacy achieved via our right-protection model, but only about its obfuscation capacity.

6.3. Obfuscation Introduced by Right-Protection

Using the previously presented *watermark embedding* (Section 3.1), we can distort the original data. The process has two parameters: the randomly chosen watermark W and the feasible power p^*, both of which are known only to the owner of the data. The scheme adds a limited amount of noise to the data used to encode a watermark W consisting of l bits distinct from 0. Understandably, the more information the watermark contains, the more distortion can be expected. The amount of information contained in a watermark depends on the number of l non-zero values $W_j \in \{-1, 1\}$.

Now, we quantify the relative distortion $r(\mathcal{D}, \hat{\mathcal{D}})$ on the data. As described in Section 3, we embed the watermark on the l high-energy frequencies. We assume that the energy on the remaining $n-l$ frequencies is negligible. This is true for real-world data, where the energy is typically sharply concentrated in the largest coefficients [Mukherjee et al. 2006]. Thus, after a renumbering of the indices,

$$\begin{aligned} ||x - \hat{x}|| &= ||X - \hat{X}|| = \left(\sum_{j=1}^{n} [X_j - \hat{X}_j]^2 \right)^{1/2} \\ &= \left(\sum_{j=1}^{n} [X_j - (1 + pW_j)X_j]^2 \right)^{1/2} = p \left(\sum_{j=1}^{n} [W_j X_j]^2 \right)^{1/2} \\ &= p \left(\sum_{j=1}^{l} X_j^2 \right)^{1/2} \simeq p||x||. \end{aligned} \tag{9}$$

Then, the relative distortion of the dataset is,

$$r(\mathcal{D}, \hat{\mathcal{D}}) = \frac{1}{|\mathcal{D}|} \sum_{k=1}^{|\mathcal{D}|} \frac{||x_k - \hat{x}_k||}{||x_k||} \simeq p. \tag{10}$$

Here, we provide an illustrative example of the obfuscation-utility curve for a spatiotemporal dataset and of the resulting distortion on one of the dataset trajectories. The data consist of taxi movements in the city of Beijing. As shown in our analysis, the obfuscation depends on the embedding power p, so we show one trajectory of the dataset for increasing p values. Note that the utility of the whole dataset remains at high levels even under larger relative distortions. We showcase obfuscation-utility curves for additional datasets in the experimental section that follows.

7. 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 of Section 4 in terms of number of operations. We report major pruning of the search space. Next, the resilience of our scheme is assessed against a broad range of potential attacks: geometric distortions, resampling, and more. Finally, we present the utility-obfuscation curves for increasing intensities of right-protection.

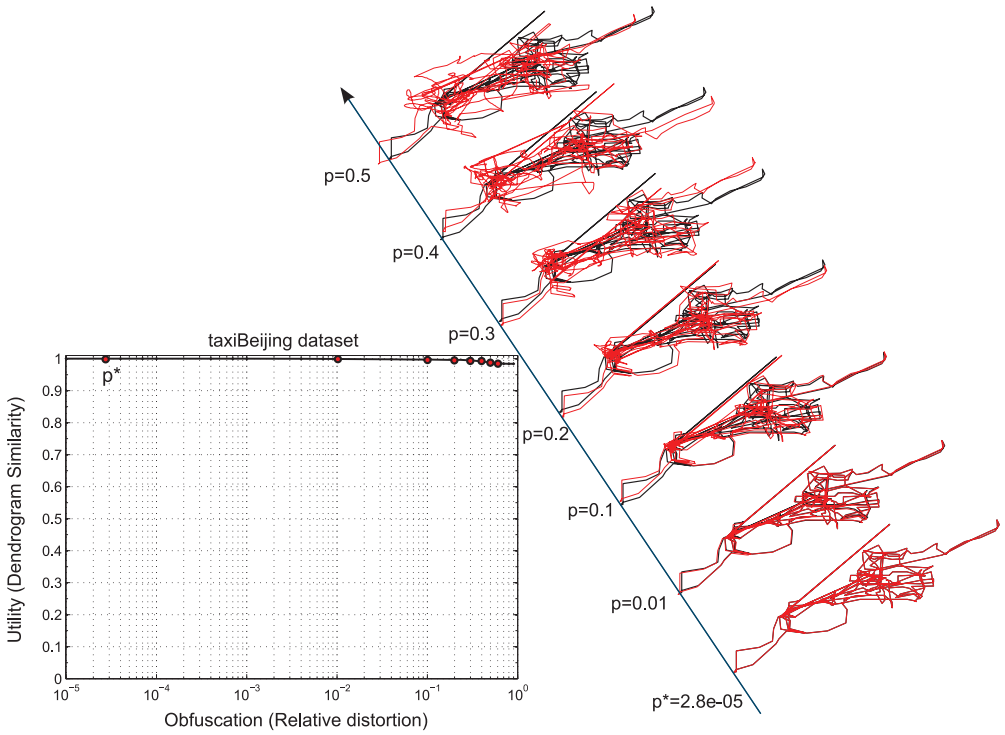

Fig. 12. Obfuscation vs. utility for a spatiotemporal dataset (taxi-Beijing). On the right, we also depict the distortion relative to the watermark embedding power p for one trajectory instance in the dataset.

Table I. Datasets Used in the Experiments

Name	Data Type	Total Data Points
nasdaq	Stock Prices	500,000
taxiSF	Taxi Traj. SF	2,580,000
taxiBeijing	Taxi Traj. Beijing	1,499,893
skulls	Image Contour	24,000
fish	Image Contour	63,232
video1	Video-Tracking	22,500
video2	Video-Tracking	11,500
hand	Handwritten	11,520

We test our methods on datasets from various application areas: mobility data (taxi trajectories in Beijing [Yuan et al. 2011, 2010] and San Francisco [Piorkowski et al. 2009]), 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 Lucchese et al. [2010]). The characteristics of our datasets are summarized in Table I and illustrated in Figures 13 and 14. All experiments have been conducted on a 2.16GHz Intel CPU with 3GB RAM.

7.1. Preservation of Distance Relations

Here, we evaluate whether the single-, complete-, and average-linkage preservation algorithms discover the correct embedding power for the watermark so that the dendrograms, both on the original and watermarked data, remain isomorphic. This means that both the tree structure and the order of the merger points $M(i)$ are the same. A

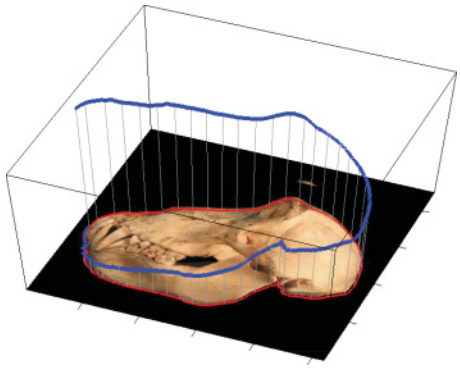

Fig. 13. Image shapes can also be treated as two-dimensional sequences by extracting the perimeter of a shape.

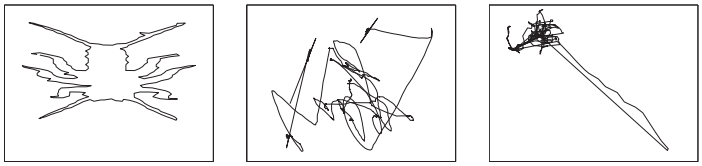

Fig. 14. Data objects from the `fish`, `video1`, and `taxiBeijing` benchmark datasets.

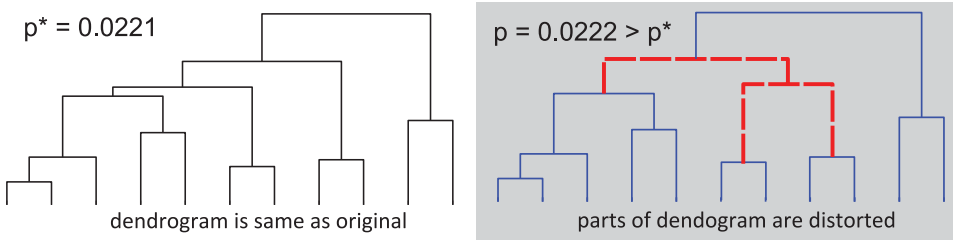

Fig. 15. Dendrogram portion for the `skulls` dataset. Left: dendrogram for maximum discovered $p^* = 0.0221$ is same as original. Right: for even slightly larger $p = 0.0222$, the dendrogram changes.

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 15, 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 that 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 II summarizes our findings.

7.2. 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 II. Maximal Watermarking Power $p^* \in [0, 0.01]$

	Single-Linkage p^*	Complete-Linkage p^*	Average-Linkage p^*
nasdaq	$0.77 \cdot 10^{-4}$	$0.243 \cdot 10^{-3}$	$0.323 \cdot 10^{-6}$
taxiSF	$0.47 \cdot 10^{-4}$	$0.97 \cdot 10^{-4}$	$0.484 \cdot 10^{-3}$
taxiBeijing	$0.4 \cdot 10^{-5}$	$0.28 \cdot 10^{-4}$	$0.18 \cdot 10^{-4}$
skulls	$0.1 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$
fish	$0.229 \cdot 10^{-3}$	$0.122 \cdot 10^{-3}$	$0.371 \cdot 10^{-3}$
video1	$0.39 \cdot 10^{-2}$	$0.1 \cdot 10^{-1}$	$0.121 \cdot 10^{-2}$
video2	$0.1 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$	$0.1 \cdot 10^{-1}$
hand	$0.2271 \cdot 10^{-2}$	$0.116 \cdot 10^{-3}$	$0.409 \cdot 10^{-3}$

The percentage of dendrogram preservation was 100% for all benchmarks.

Table III. Number of Quadratic Inequalities Solved for Different Datasets, Single Linkage

Dataset	Single-Linkage	Fast Single-Linkage	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 IV. Number of Quadratic Inequalities Solved for Different Datasets, Complete Linkage

Dataset	Complete-Linkage	Fast Complete-Linkage	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 III 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 reduction in terms of solved inequalities; up to three orders of magnitude. Tables IV and V report the results of the same experiment for the case of complete- and average-linkage preservation.

7.3. 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 the embedded watermark's detectability 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). Figure 16 depicts the ROC curves representing true- versus false-positive rates. We also report the performance of a random baseline that randomly classifies the dataset as having or not having the watermark embedded.

Table V. Number of Quadratic Inequalities Solved for Different Datasets, Average Linkage

Dataset	Average-Linkage	Fast Average-Linkage	Pruning
nasdaq	22,897,415	26,144	**875**×
taxiSF	27,880,050	6,678	**4,174**×
taxiBeijing	200,191,602	28,752	**6,962**×
skulls	665	2	**333**×
fish	2,511,250	1,157	**2,170**×
video1	546	17	**32**×
video2	2,002	22	**91**×
hand	121,396	181	**670**×

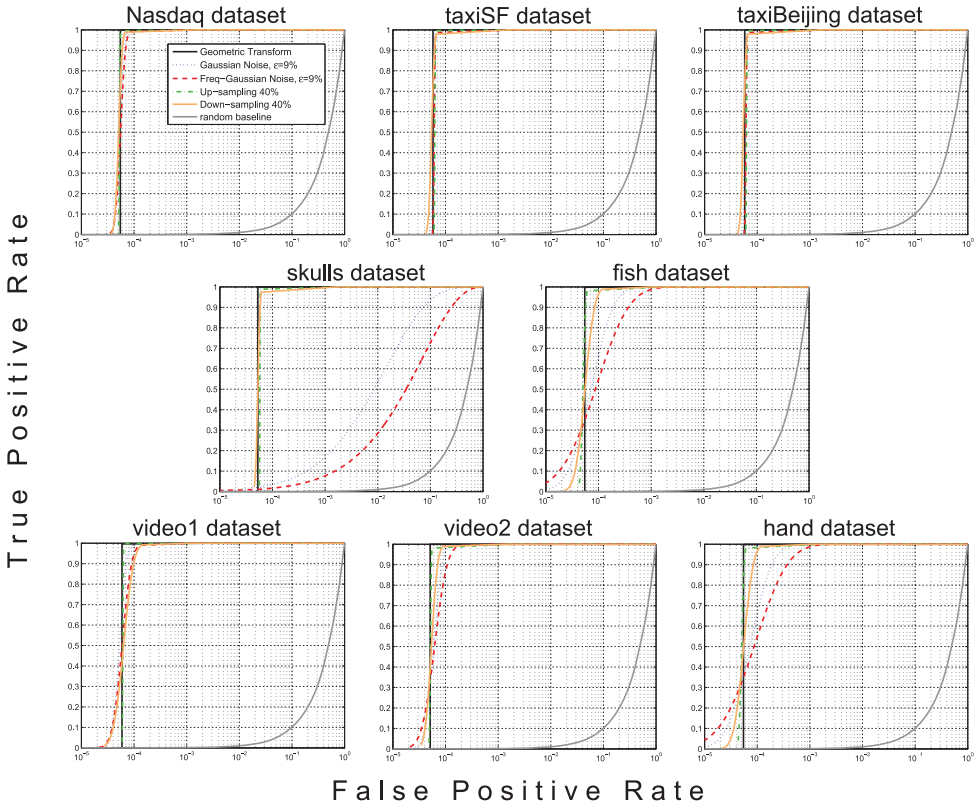

Fig. 16. ROC curves for watermark detection corresponding to geometric transformations, noise addition, noise addition in the frequency domain, up-sampling, and down-sampling on different datasets.

We observe that our detection method works very effectively. It is more than four 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.

7.4. Obfuscation vs. Utility

Now we evaluate the efficacy of our right-protection methodology as a data obfuscation method, particularly for larger embedding powers p of the watermark. In particular, we allow p to be greater than 0.01 (or 1% relative distortion) and test for values as large as $p = 1$. We compute the utility and obfuscation using the metrics of Section 6

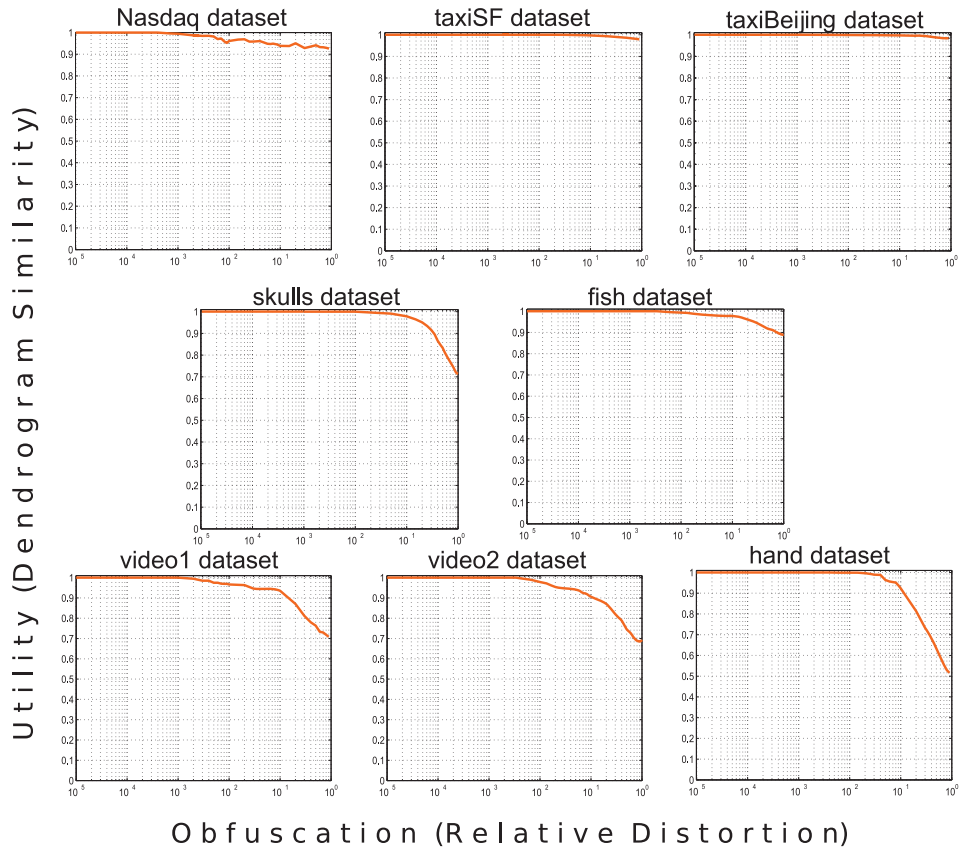

Fig. 17. Tradeoff curves between obfuscation and utility for average-linkage clustering and diverse datasets. Observe that utility (orange line) is kept at consistently high levels even for large amounts of obfuscation.

for increasing embedding powers p. We investigated the tradeoff for all HC variants (SLC, CLC and ALC). The results are similar for SLC, CLC and ALC, so we only show the plots for ALC in Figure 17.

Each experiment computes both utility (dendrogram similarity before and after the watermark embedding) and obfuscation (relative distortion) using 100 different watermarks W of length $l = 64$ chosen uniformly at random for each power p. The tradeoff curve captures the average value of all the different watermark embeddings.

The right-protection algorithm returns back the maximal embedding power of the watermark p^* that does not change the utility. Therefore, for $p \in [0, p^*]$, we achieve the maximum possible utility. Note that for all datasets there exists a range of embedding powers that achieves utility of 100% in the sense that HC remains identical before and after the right-protection process. For embedding powers greater than p^*, one can no longer guarantee 100% preservation of utility. Of course, the average relative data distortion is upper bounded by the embedding power p, as we saw in our analysis in Section 6.3. We observe that the overall dataset utility after right-protection is always kept at high levels, more than 70% for the majority of datasets, even for the maximal value of p.

What is interesting to note is that, for the larger datasets (e.g., Nasdaq and taxi movements) that contain a lot of data objects, the rate of reduction in utility is slower compared to the smaller datasets. Even for $p = 1$, or relative distortion of 100%, the utility preservation is greater than 90%. This can be easily understood because larger datasets automatically lead to bigger HC trees. Whereas the obfuscation process may change relationships between lower parts of the leafs in a tree, the higher parts of the tree (aggregate clusters) are more likely to remain the same. This is an important observation because it suggests that, for big datasets, the utility preservation is expected to remain at consistently high levels even under increased levels of obfuscation.

8. RELATED WORK

Watermarking is a steganographic technique used for establishing ownership, with many applications in multimedia datasets [Cox et al. 1997] such as images [Moulin et al. 2000], vector graphics [Xiamu Niu 2006], audio [Bassia et al. 2001; Swanson et al. 1998], and video [Simitopoulos et al. 2002; Zhu et al. 1999]. Multimedia watermarking focuses on the protection of a *single* object while 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 HC.

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 [Liu and Thuraisingham 2006; Kargupta et al. 2003], condensation [Aggarwal and Yu 2004], or data transformation [Chen and Liu 2005; Oliveira and Zaïane 2010]. Similar notions have also been used for watermarking databases [Agrawal and Kiernan 2002; Sion et al. 2004]. Contrary to these 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.

Privacy protection via *dataset partition* is achieved using horizontal or vertical data partitioning [Vaidya and Clifton 2003; Jagannathan et al. 2006; Yu et al. 2006a, 2006b]. Different portions of the data are distributed to different sites, and data exchange without leakage of private information becomes possible through cryptographic techniques (multiparty 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 [Sion et al. 2006]. In contrast to our approach, Sion et al. 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.). Approaches for watermarking categorical data, which we do not address in this work, can be found in Atallah et al. [2004] and Coatrieux et al. [2011].

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. Right-protection schemes based on watermarking principles that preserve Nearest-Neighbors (NNs) of objects using either additive or multiplicative techniques have been presented in Lucchese et al. [2010] and Zoumpoulis et al. [2014], respectively. We adopt the spread-spectrum watermarking model of these works, but here we study the more elaborate case of HC preservation.

We also examine the tradeoff between obfuscation and mining utility on watermarking. An approach for anonymization in data publishing that relies on maintaining certain patterns is proposed in Xue et al. [2011]. The key idea is to sample the solution space uniformly at random by performing a random walk. They illustrate their technique by applying it to k-means clustering. However, for HC, they do not provide any guarantees. The topic of privacy-preserving data publishing for cluster analysis has also been examined in Fung et al. [2009], where the clustering problem is tackled via a transformation to a classification problem.

In Mukherjee et al. [2006], the Fourier transform was used for privacy preservation and data reduction. Frequencies containing little energy are suppressed. One of the findings was that, for real-world datasets, it suffices to simply pick the first couple of high-energy coefficients of the Fourier transform of the entire dataset such that some high-energy coefficients are also chosen for each point. We confirm this to a great extent. To further increase privacy, these coefficients may also be permuted. The paper evaluates the scheme on k-means clustering but does not offer any utility guarantees. In Li and Li [2009], the Pareto-efficiency principle from economics is applied to study the tradeoff between privacy and utility in data publishing. Parameswaran and Blough [2005] uses NN substitution to guarantee privacy and cluster preservation. Our approach does not need to replace NN's, but it learns the maximum amount of noise that can be added, which guarantees cluster preservation. The method presented in Oliveira and Zaïane [2010] randomly rotates, scales, and/or translates the data (thus hiding them); therefore, original distances—and hence clustering—is preserved. However, this approach is very susceptible to attacks, as shown in Turgay et al. [2008]. There has also been a number of papers dealing specifically with the anonymization of trajectories. In Terrovitis and Mamoulis [2008], an attacker might know parts of a trajectory, but this knowledge should not allow him to identify other locations given a set of published trajectories. The work discusses location and time-series anonymization through value generalization to the spatiotemporal field. An extension of k-anonymity to space-time trajectories using co-location is proposed in Abul et al. [2008]. Through space translation, the exact location of an object is disguised. The LKC privacy model for trajectories is introduced in Mohammed et al. [2009]. It draws on the observation that trajectories are high dimensional and sparse, thus making k-anonymity an improper tool due to its heavy negative impact on data utility. Privacy is achieved through global suppression; that is, if a sequence of locations is suppressed in one trajectory, then it is also suppressed in all other trajectories. The notion of differential privacy has also attracted a lot of attention. In Xiao et al. [2011], the authors make use of orthonormal transforms (wavelets) to guide the process of enforcing differential privacy. Also, Jiang et al. [2013] present a mechanism for publishing trajectories of ships that is ϵ-differentially private and also provides good utility.

9. CONCLUSION

Right-protection presents an inherent tradeoff: The ownership key should be embedded with high intensity to guarantee robustness of detection, but, at the same time, the embedding should compromise to the least amount the utility of the dataset, so that it is useful for subsequent mining operations. We present 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 HC operations on the modified dataset. The fast variants that we put forward can reduce, in certain cases, the search space by more than 6,000 times compared to the exhaustive algorithms, with no sacrifice in accuracy. Our analysis is generic and delivers great promise for a broader applicability for other distance-based mining operations, such as anomaly detection, classification, and visualization.

REFERENCES

Osman Abul, Francesco Bonchi, and Micro Nanni. 2008. Never walk alone: Uncertainty for anonymity in moving objects databases. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering (ICDE'08)*. IEEE Computer Society, Washington, DC, 376–385.

Charu C. Aggarwal and Philip S. Yu. 2004. A condensation approach to privacy preserving data mining. In *Advances in Database Technology - EDBT 2004*, Elisa Bertino, Stavros Christodoulakis, Dimitris Plexousakis, Vassilis Christophides, Manolis Koubarakis, Klemens Böhm, and Elena Ferrari (Eds.). Lecture Notes in Computer Science, Vol. 2992. Springer, Berlin, 183–199.

Charu C. Aggarwal and Philip S. Yu. 2008. A general survey of privacy-preserving data mining models and algorithms. In *Privacy-Preserving Data Mining*, Charu C. Aggarwal and Philip S. Yu (Eds.). Advances in Database Systems, Vol. 34. Springer, 11–52.

Rakesh Agrawal and Jerry Kiernan. 2002. Watermarking relational databases. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB'02)*. VLDB Endowment, 155–166.

Claudio Agostino Ardagna, Marco Cremonini, Ernesto Damiani, Sabrina De Capitani di Vimercati, and Pierangela Samarati. 2007. Location privacy protection through obfuscation-based techniques. In *Data and Applications Security XXI*, Steve Barker and Gail-Joon Ahn (Eds.). Lecture Notes in Computer Science, Vol. 4602. Springer, Berlin, 47–60.

Mikhail J. Atallah, Sunil Prabhakar, Keith B. Frikken, and Radu Sion. 2004. Digital rights protection. *IEEE Data Engineering Bulletin* 27, 1 (2004), 19–25.

Paraskevi Bassia, Ioannis Pitas, and Nikos Nikolaidis. 2001. Robust audio watermarking in the time domain. *IEEE Transactions on Multimedia* 3, 2 (2001), 232–241.

Steve Borgatti. 2007. Distance and Correlation. (2007). Retrieved March 30, 2014 from http://www.analytictech.com/mb876/handouts/distance_and_correlation.htm.

Keke Chen and Ling Liu. 2005. Privacy preserving data classification with rotation perturbation. In *Proceedings of the 5th IEEE International Conference on Data Mining*. 589–592.

Rui Chen, Benjamin C. M. Fung, and Bipin C. Desai. 2011. Differentially private trajectory data publication. *CoRR* abs/1112.2020 (2011).

Gouenou Coatrieux, Emmanuel Chazard, Régis Beuscart, and Christian Roux. 2011. Lossless watermarking of categorical attributes for verifying medical data base integrity. In *Proceedings of the 33th IEEE Annual International Conference of the Engineering in Medicine and Biology Society*. 8195–8198.

Eric Cope and Gianluca Antonini. 2008. Observed correlations and dependencies among operational losses in the ORX consortium database. *Journal of Operational Risk* 3, 4 (2008), 47–76.

Ingemar J. Cox, Joe Kilian, F. Thomson Leighton, and Talal Shamoon. 1997. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing* 6, 12 (1997), 1673–1687.

Daniel Defays. 1977. An efficient algorithm for a complete link method. *Computer Journal* 20, 4 (1977), 364–366.

Olivier Devillers and Mordecai J. Golin. 1995. Incremental algorithms for finding the convex hulls of circles and the lower envelopes of parabolas. *Information Processing Letters* 56, 3 (1995), 157–164.

Nick G. Duffield and Matthias Grossglauser. 2001. Trajectory sampling for direct traffic observation. *IEEE/ACM Transactions on Networking* 9, 3 (2001), 280–292.

Ixchel M. Faniel and Ann Zimmerman. 2011. Beyond the data deluge: A research agenda for large-scale data sharing and reuse. *International Journal of Digital Curation* 6, 1 (2011), 58–69.

E. B. Fowlkes and C. L. Mallows. 1983. A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association* 78, 383 (1983), 553–569.

Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. 2010. Privacy-preserving data publishing: A survey of recent developments. *ACM Computing Surveys* 42, 4, Article 14 (June 2010), 53 pages.

Benjamin C. M. Fung, Ke Wang, Lingyu Wang, and Patrick C. K. Hung. 2009. Privacy-preserving data publishing for cluster analysis. *Data and Knowledge Engineering* 68, 6 (2009), 552–575.

Roxana Geambasu, Steven D. Gribble, and Henry M. Levy. 2009. CloudViews: Communal data sharing in public clouds. In *Proceedings of the 2009 Conference on Hot Topics in Cloud Computing (HotCloud'09)*. USENIX Association, Article 14.

Gabriel Ghinita, Panagiotis Karras, Panos Kalnis, and Nikos Mamoulis. 2009. A framework for efficient data anonymization under privacy and accuracy constraints. *ACM Transactions on Database Systems* 34, 2, Article 9 (July 2009), 47 pages.

Philippe Golle and Kurt Partridge. 2009. On the anonymity of home/work location pairs. In *Proceedings of the 7th International Conference on Pervasive Computing (Pervasive'09)*. Springer-Verlag, Berlin, 390–397.

HGP 2013. All About The Human Genome Project. Retrieved from http://www.genome.gov/10001772.

Geetha Jagannathan, Krishnan Pillaipakkamnatt, and Rebecca N. Wright. 2006. A new privacy-preserving distributed k-clustering algorithm. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. 494–498.

Kaifeng Jiang, Dongxu Shao, Stéphane Bressan, Thomas Kister, and Kian-Lee Tan. 2013. Publishing trajectories with differential privacy guarantees. In *Proceedings of the 25th International Conference on Scientific and Statistical Database Management (SSDBM'13)*. ACM, New York, NY, Article 12, 12 pages.

Hillol Kargupta, Souptik Datta, Qi Wang, and Krishnamoorthy Sivakumar. 2003. On the privacy preserving properties of random data perturbation techniques. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*. IEEE Computer Society, Washington, DC, 99–106.

Tiancheng Li and Ninghui Li. 2009. On the tradeoff between privacy and utility in data publishing. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*. ACM, New York, NY, 517–526.

Li Liu and Bhavani Thuraisingham. 2006. The applicability of the perturbation model-based privacy preserving data mining for real-world data. In *Proceedings of the 6th IEEE International Conference on Data Mining - Workshops (ICDMW'06)*. IEEE Computer Society, Washington, DC, 507–512.

Claudio Lucchese, Michail Vlachos, Deepak Rajan, and Philip S. Yu. 2010. Rights protection of trajectory datasets with nearest-neighbor preservation. *The VLDB Journal* 19, 4 (Aug. 2010), 531–556.

Wolfgang Ludwig and Hans-Peter Klenk. 2001. Overview: A phylogenetic backbone and taxonomic framework for procaryotic systematics. In *Bergey' s Manual of Systematic Bacteriology*. Springer, 49–65.

Noman Mohammed, Benjamin C. M. Fung, and Mourad Debbabi. 2009. Walking in the crowd: Anonymizing trajectory data for pattern analysis. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM'09)*. ACM, New York, NY, 1441–1444.

Marco Casassa Mont, Ilaria Matteucci, Marinella Petrocchi, and Marco Luca Sbodio. 2012. *Enabling Data Sharing in the Cloud*. Technical Report. HP Laboratories, Tech Report HPL-2012–22.

Pierre Moulin, Mehmet Kivanç Mihcak, and Gen-Iu Lin. 2000. An information-theoretic model for image watermarking and data hiding. In *Proceedings of the IEEE International Conference on Image Processing*, Vol. 3. 667–670.

Shibnath Mukherjee, Zhiyuan Chen, and Aryya Gangopadhyay. 2006. A privacy-preserving technique for Euclidean distance-based mining algorithms using Fourier-related transforms. *The VLDB Journal* 15, 4 (Nov. 2006), 293–315.

Fionn Murtagh. 1984. Complexities of hierarchic clustering algorithms: State of the art. *Computational Statistics Quarterly* 1, 2 (1984), 101–113.

Mehmet Ercan Nergiz, Maurizio Atzori, Yücel Saygin, and Baris Güç. 2009. Towards trajectory anonymization: A generalization-based approach. *Transactions on Data Privacy* 2, 1 (April 2009), 47–75.

Stanley R. M. Oliveira and Osmar R. Zaïane. 2010. Privacy preserving clustering by data transformation. *Journal of Information and Data Management* 1, 1 (2010), 37–52.

Rupa Parameswaran and D. Blough. 2005. A robust data obfuscation approach for privacy preservation of clustered data. In *Proceedings of the 2005 IEEE International Conference on Data Mining*. 18–25.

Christine Parent, Stefano Spaccapietra, Chiara Renso, Gennady Andrienko, Natalia Andrienko, Vania Bogorny, Maria Luisa Damiani, Aris Gkoulalas-Divanis, Jose Macedo, Nikos Pelekis, Yannis Theodoridis, and Zhixian Yan. 2013. Semantic trajectories modeling and analysis. *ACM Computing Surveys* 45, 4, Article 42 (Aug. 2013), 32 pages.

Michal Piorkowski, Natasa Sarafijanovic-Djukic, and Matthias Grossglauser. 2009. A parsimonious model of mobile partitioned networks with clustering. In *Proceedings of the International Conference on Communication Systems and Networks and Workshops (COMSNETS'09)*. 1–10.

Gang Qian, Shamik Sural, Yuelong Gu, and Sakti Pramanik. 2004. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In *Proceedings of the 2004 ACM Symposium on Applied Computing (SAC),* Hisham Haddad, Andrea Omicini, Roger L. Wainwright, and Lorie M. Liebrock (Eds.). ACM, 1232–1237.

Robin Sibson. 1973. SLINK: An optimally efficient algorithm for the single-link cluster method. *Computer Journal* 16, 1 (1973), 30–34.

John Van Sickle. 1997. Using mean similarity dendrograms to evaluate classifications. *Journal of Agricultural, Biological, and Environmental Statistics* (1997), 370–388.

Dimitrios Simitopoulos, Sotirios A. Tsaftaris, Nikolaos V. Boulgouris, and Michael G. Strintzis. 2002. Compressed-domain video watermarking of MPEG streams. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, Vol. 1. IEEE, 569–572.

Radu Sion, Mikhail Atallah, and Sunil Prabhakar. 2004. Rights protection for relational data. *IEEE Transactions on Knowledge and Data Engineering* 16, 12 (Dec. 2004), 1509–1525.

Radu Sion, Mikhail Atallah, and Sunil Prabhakar. 2006. Rights protection for discrete numeric streams. *IEEE Transactions on Knowledge and Data Engineering* 18, 5 (May 2006), 699–714.

Chaoming Song, Zehui Qu, Nicholas Blumm, and Albert-Lszl Barabsi. 2010. Limits of predictability in human mobility. *Science* 327, 5968 (2010), 1018–1021.

Mitchell D. Swanson, Bin Zhu, Ahmed H. Tewfik, and Laurence Boney. 1998. Robust audio watermarking using perceptual masking. *Signal Processing* 66, 3 (1998), 337–355.

Manolis Terrovitis and Nikos Mamoulis. 2008. Privacy preservation in the publication of trajectories. In *Proceedings of the the 9th International Conference on Mobile Data Management (MDM'08)*. IEEE Computer Society, Washington, DC, USA, 65–72.

E. Onur Turgay, Thomas B. Pedersen, Yücel Saygın, Erkay Savaş, and Albert Levi. 2008. Disclosure risks of distance preserving data transformations. In *Scientific and Statistical Database Management*, Bertram Ludäscher and Nikos Mamoulis (Eds.). Lecture Notes in Computer Science, Vol. 5069. Springer, Berlin, 79–94.

Jaideep Vaidya and Chris Clifton. 2003. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, New York, NY, 206–215.

Michail Vlachos, Marios Hadjieleftheriou, Dimitrios Gunopulos, and Eamonn Keogh. 2003. Indexing multidimensional time-series with support for multiple distance measures. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*. ACM, New York, NY, 216–225.

Aleš Žiberna and Vesna Žabkar. 2003. Application of end-users market segmentation using statistical methods. In *Developments in Applied Statistics*, Anuška Ferligoj and Andrej Mrvar (Eds.). metodološki zvezki - Advances in Methodology and Statistics, Vol. 19. 243–263.

Stephen B. Wicker. 2012. The loss of location privacy in the cellular age. *Commununications of the ACM* 55, 8 (Aug. 2012), 60–68.

Raymond Chi-Wing Wong, Ada Wai-Chee Fu, Ke Wang, Philip S. Yu, and Jian Pei. 2011. Can the utility of anonymized data be used for privacy breaches? *ACM Transactions on Knowledge Discovery Data* 5, 3, Article 16 (Aug. 2011), 24 pages.

Xiaotong Wang Xiamu Niu, Chengyong Shao. 2006. A survey of digital vector map watermarking. *International Journal of Innovative Computing, Information and Control* 2, 6 (2006), 1301–1316.

Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. 2011. Differential privacy via wavelet transforms. *IEEE Transactions on Knowledge and Data Engineering* 23, 8 (2011), 1200–1214.

Andy Yuan Xue, Rui Zhang, Yu Zheng, Xing Xie, Jianhui Yu, Yong Tang, Sapna Jain, and Jingren Zhou. 2013. DesTeller: A system for destination prediction based on trajectories with privacy protection. *PVLDB* 6, 12 (2013), 1198–1201.

Mingqiang Xue, Panagiotis Karras, Chedy Raïssi, and Hung Keng Pung. 2011. Utility-driven anonymization in data publishing. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management (CIKM'11)*. 2277–2280.

Hwanjo Yu, Xiaoqian Jiang, and Jaideep Vaidya. 2006a. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. In *Proceedings of the 2006 ACM Symposium on Applied Computing (SAC'06)*. 603–610.

Hwanjo Yu, Jaideep Vaidya, and Xiaoqian Jiang. 2006b. Privacy-preserving SVM classification on vertically partitioned data. In *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining (PAKDD'06)*. 647–656.

Jing Yuan, Yu Zheng, Xing Xie, and Guangzhong Sun. 2011. Driving with knowledge from the physical world. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'11)*. 316–324.

Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: Driving directions based on taxi trajectories. In *Proceedings of SIGSPATIAL International Conference on Geographic Information Systems (GIS'10)*. 99–108.

Wenwu Zhu, Zixiang Xiong, and Ya-Qin Zhang. 1999. Multiresolution watermarking for images and video. *IEEE Transactions on Circuits and Systems for Video Technology* 9, 4 (1999), 545–550.

Spyros I. Zoumpoulis, Michail Vlachos, Nikolaos M. Freris, and Claudio Lucchese. 2014. Right-protected data publishing with provable distance-based mining. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 2014–2028.

Received November 2013; revised April 2014; accepted September 2014