Adaptive Privacy-Preserving Visualization Using Parallel Coordinates

Aritra Dasgupta and Robert Kosara

Abstract—Current information visualization techniques assume unrestricted access to data. However, privacy protection is a key issue for a lot of real-world data analyses. Corporate data, medical records, etc. are rich in analytical value but cannot be shared without first going through a transformation step where explicit identifiers are removed and the data is sanitized. Researchers in the field of data mining have proposed different techniques over the years for privacy-preserving data publishing and subsequent mining techniques on such sanitized data. A well-known drawback in these methods is that for even a small guarantee of privacy, the utility of the datasets is greatly reduced. In this paper, we propose an adaptive technique for privacy preservation in parallel coordinates. Based on knowledge about the sensitivity of the data, we compute a clustered representation on the fly, which allows the user to explore the data without breaching privacy. Through the use of screen-space privacy metrics, the technique adapts to the user's screen parameters and interaction. We demonstrate our method in a case study and discuss potential attack scenarios.

Index Terms—Parallel coordinates, privacy, clustering.

1 Introduction

Visualization techniques currently have an underlying assumption that there is unrestricted access to data. In reality, access to data in many cases is restricted to protect sensitive information from being leaked. There are legal regulations like the *Health Insurance Portability and Accountability Act* (HIPAA) in the United States that regulate disclosure of private data. Privacy can be personal (e.g., medical records) or corporate (e.g., company records) [8]. The main concern with sensitive data is their misuse [3]. Such data are therefore released publicly only after removing explicit identifiers. Over the past few years, many privacy-preserving data mining (PPDM) techniques have been developed for mining and publishing of sanitized data.

Many sensitive datasets are susceptible to privacy breaches through linking to external information. To address this, PPDM distinguishes between *quasi-identifiers* and *sensitive attributes*. Sensitive attributes are those whose exact values need to be protected, so that they cannot be linked to an individual. Examples include disease names in medical databases and company-specific information in corporate databases. Quasi-identifiers are those attributes that, taken together, can identify an individual even if that person's name or complete address is not included. This is possible because the same information can be found in public databases, and enough quasi-identifiers can narrow the choices down to a single person (e.g, given a date of birth, gender, and ZIP code, it is often possible to pinpoint a single person in the U.S.).

The problem with PPDM techniques is that even for a minimal privacy guarantee, there is a significant loss of utility [5]. Moreover, the published output might still be susceptible to mining by malicious users, who might use data analysis results to breach the privacy safeguards.

In this paper, we adapt parallel coordinates for use in privacy-preserving visualization. Instead of publishing the data or just the analysis results, a privacy-preserving information visualization tool provides an interactive interface to both the data owner and outside users. The data owner can customize the tool to choose different reordered configurations of the data he wants to show to analysts without sacrificing privacy. Outside users cannot directly access the data, but only visualize the patterns in the data through the tool. Different constraints are imposed by the technique to prevent them from breaching privacy through interaction. The data is sanitized on the fly, based on the user's screen resolution and other viewing parameters.

Similar to PPDM, we assume that the data holder is aware of the sensitivity of the data attributes and the context in which privacy can be breached. In both cases the goal of a privacy-preserving technique is to minimize disclosure of sensitive information even after data analysis techniques have been used. At the same time, the access of non-sensitive information and their fidelity should be minimally affected by the sanitization process.

We summarize the contributions of our paper as follows:

1. Privacy-preserving information visualization: we propose a model for protecting information privacy using visualization techniques based on screen-space metrics. We introduce and illustrate the different aspects of our model in Section 3.

2. Implementation of the k-anonymity and l-diversity concepts in parallel coordinates: we apply clustering in parallel coordinates based on the well-established k -anonymity and l-diversity metrics (Section 4).

3. Investigation of interaction leading to attack scenarios: in the context of privacy-preservation, interaction in visualization is both an advantage and a challenge at the same time. The advantage is that, unlike conventional data mining approaches, we keep the utility high by adapting the views to user interaction (like reordering). The challenge lies in effectively handling the different interaction conditions that may lead to potential privacy breach scenarios. We discuss our application and attack scenarios in Section 5 and Section 6.

2 Related Work

Many techniques for publication and analysis of de-identified sensitive data have been developed in the field of privacy-preserving data mining [2]. The k-anonymity model [26, 27] focuses on making k records indistinguishable with respect to the quasi-identifiers so that identification through linking is prevented. The k-anonymity problem has been shown to be NP-hard [22] and therefore many approximation algorithms have been proposed [1]. We use the k-member clustering algorithm proposed by Byun et al. [7]. While we adopt the overall algorithm, we use a different criterion for seeding and a different cost function. We also apply the algorithm individually to each axis pair, rather than across all dimensions at once.

k-anonymity does not ensure sufficient diversity in sensitive attributes: even if records are indistinguishable with respect to quasi-identifiers, the malicious user can use background knowledge and

- *Aritra Dasgupta is with UNC Charlotte, E-mail: adasgupt@uncc.edu.*
- *Robert Kosara is with UNC Charlotte, E-mail: rkosara@uncc.edu.*

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

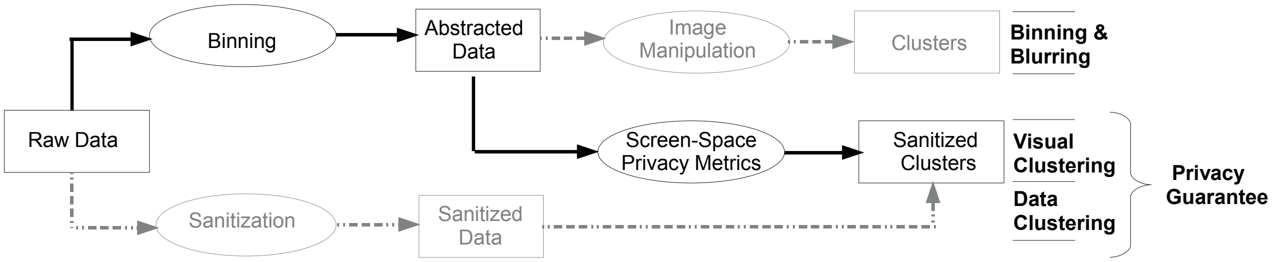
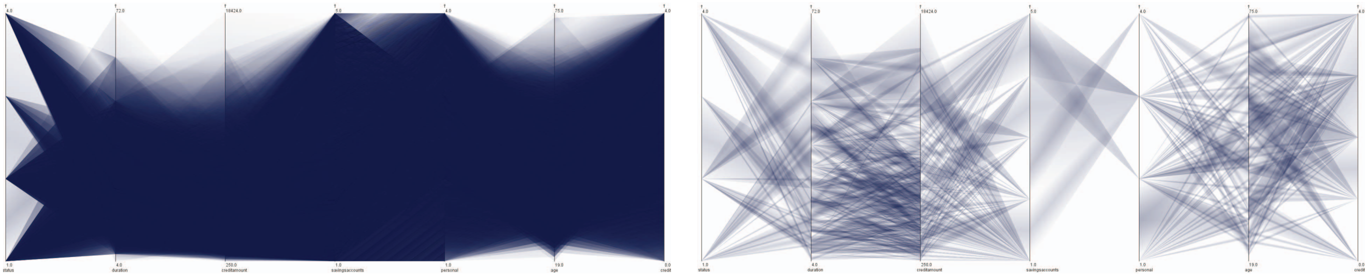


Fig. 1. Possible approaches to privacy-preservation in information visualization (Section 3.2): binning and blurring, data clustering, and visual clustering. Only the bottom two approaches guarantee a given level of privacy, in the case of data clustering, it leads to considerable loss of utility.



(a) Visualizing data that was sanitized using approaches from privacy-preserving data mining results in poor utility.

(b) Clustering by axis pair and using a different metric in the clustering, more of the visual structure can be retained.

Fig. 2. Comparing data clustering to our visual clustering approach. Both algorithms make it impossible to tell fewer than three records apart, but our approach provides higher utility.

breach the privacy. To address this, Machanavajjhala et al. proposed l-diversity [21] which ensures each group has at least l different values for the sensitive attribute. For example, in a disease dataset if a particular quasi-identifier group for people belonging to age above 50 is associated with the same sensitive value *cancer*, then a user who knows the age of a person can correctly guess the value. We use this l-diversity property as a basis for constraining the user interaction and show at least l different values for the sensitive attribute.

In the parallel coordinates [15] research literature, there have been several approaches to clustering. Much of the clustering work focuses on approaches for clutter reduction and improving the perceptual aspects of parallel coordinates plots. Zhou et al. propose geometrically deforming and grouping poly-lines to overcome edge clutter [29]. Johansson et al. look at overcoming the problem of over-plotting by using high-precision textures [16]. While their goal is to maximize within-cluster information, we aim to protect privacy of the records within a cluster. Clustering has also been applied using data space properties [14, 23]. The most critical difference between the existing work and our approach is that we guarantee a minimum cluster size for privacy-protection purposes [10].

3 Data Privacy in Visualization

In this section we discuss where visualization fits in, the potential approaches, and the architecture on which we have based our technique.

3.1 Scenarios

Typical scenarios in privacy-preserving applications are: a) a medical company wants to protect its proprietary data, yet they want to release either the sanitized data or analyzed results for scientific uses and b) two marketing companies having different attributes for a common set of individuals can share their analyses and find optimal customer targets to meet their business goals.

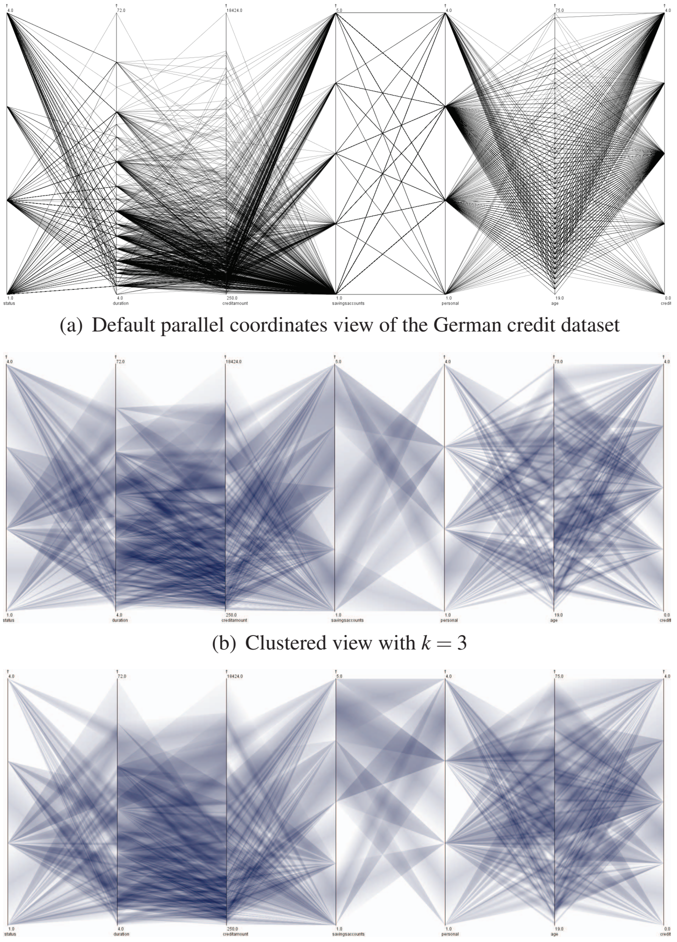
This has been addressed in privacy-preserving data mining mainly by sanitizing the data, so that key identifiers, which would be used as an input to the mining algorithm, are hidden. This approach is termed protection of *input privacy* [6]. However, the published results can still be used by attackers to extract sensitive information, which is referred to as *output privacy* and largely overlooked in the literature. Our technique fits into the realm of output privacy, where we aim to build an analytic tool on top of the data but aim to protect the sensitive information that can be gained from it.

3.2 Approaches to Privacy-Preserving Visualization

Why develop a new approach to hiding information in visualization? Naive approaches, like blurring the image or visualizing the output of a PPDM sanitization technique, turn out to be ineffective. This section discusses the shortcomings of these approaches (Figure 1).

Binning and Blurring: When a dataset is visualized, there is a natural loss of precision due to the limited resolution of the screen. To add to the information loss, the image could be blurred to hide individual records. The drawback of this approach, however, is the limited control over information loss. Single data points might exist far enough away from others so as not to be blurred together with them. There is no guarantee that each visible point contains at least a given minimum number of records.

Data-Space Sanitization: Another approach is applying the sanitization algorithms proposed in the data-mining literature and then visualizing the resulting data. While that guarantees a level of privacy, it also comes at the price of greatly reduced utility, a problem that is well-known in PPDM [5]. The resulting visualization is practically useless (Figure 2(a)), as the clusters cover most of the axes as opposed



(a) Default parallel coordinates view of the German credit dataset

(b) Clustered view with $k = 3$

(c) Clustered View with $k = 6$

Fig. 3. Illustrating k-anonymity for different values of k.

to using our approach (Figure 2(b)).

Screen-Space Sanitization: Effective visual representation is one of the key factors that lead to high utility in visualization [20]. This requires modeling the appearance of a visualization on screen, and controlling the attributes of the visualization to control the amount of information that is shown to the user. Using visual metrics for the sanitization process, it is possible to develop a clustering method that is much better suited for the purpose of visualization. This is the approach we describe in this paper.

3.3 Architecture

Our technique is based on a client-server architecture (Figure 4), where the data resides on the server and the client only fetches the clustered data and displays it on screen. This model is similar to the idea of interactive privacy described by Dwork [12], where an interface is provided by data owners to interactively filter responses to user queries and add noise where needed to preserve privacy. Similarly, in our system, the user cannot access the raw data, but can only set the interaction parameters necessary (in this case order of dimensions and number of pixels) for the server to apply the privacy-preservation algorithm (in this case, clustering) and return the sanitized clusters. Axis order is important: the server has knowledge of which dimensions are sensitive and which ones are quasi-identifiers so the output is tailored towards that configuration. The screen-space metrics, used as a starting point for clustering in our technique, are dependent on pixel-based binning. The number of pixels thus determines the appearance of the clusters.

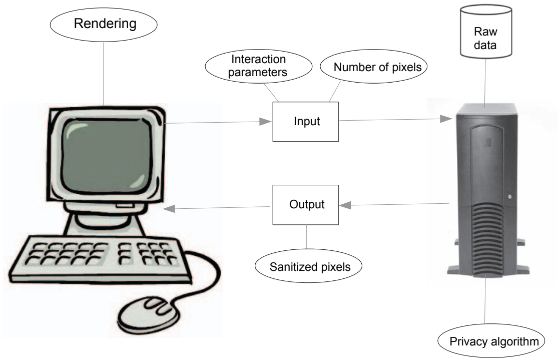


Fig. 4. Architecture for handling privacy

4 Privacy-Preserving Parallel Coordinates

Our privacy-preserving visualization model is based on controlling the information loss that occurs while mapping data points to screen-space of limited resolution. We intentionally hide information from the user by imposing de-identification constraints in screen space. To achieve this, we combine two types of information loss: the inherent information loss in parallel coordinates and additional loss from grouping together records to achieve a desired level of privacy.

In parallel coordinates, both intended and unintended information loss [30] can be observed. While there is intended information loss like binning of lines leading to over-plotting, undesirable visual artifacts like too many line crossings and clutter is an example of unintended information loss. We have proposed a set of screen-space metrics to quantify these properties [9], which we use in this technique.

4.1 Implementation of k-Anonymity

We have extended our previous work [10] and implemented a privacy-preserving variant of parallel coordinates. Based on the idea of k-anonymity, our program combines k records into one cluster, and displays it as a trapezoid instead of as individual lines (Figure 3). We have adapted an existing clustering mechanism in order to maximize the utility of the resulting clusters for visualization [7]. Our implementation is based on considering parallel coordinates as a sequence of axis pairs. This has two advantages: a) the clustering algorithm takes local properties between adjacent axis pairs into account, independent of the other axes, as a result of which cluster sizes are optimized and b) as pointed out by Li et al. [19] in parallel coordinates users are ultimately interested in finding patterns between pairs of axes. Our approach, therefore, relates directly to what we actually perceive between the axis-pairs. Sensitive attributes are usually categorical, while quasi-identifiers are generally a mix of both. Our model can handle both types of variables.

We use pixel coordinates for our model: all coordinates are first transformed into screen space and then rounded to integers. This places them in pixel-sized bins that reflect the precision of the display.

4.1.1 Seeding

The quality of the clusters depends strongly on the cluster seeds we choose. Axis-pairwise clustering enables us to look at the information loss between adjacent dimensions and select our seeds at different stages of the iteration based on that. In previous work[10], we used the degree of over-plotting as the seeding criterion. However, there are other properties of the bins, like convergence/divergence [9] that need to be taken into account: adjacent dimensions which are both categorical or both numerical with very few distinct values are likely to have more converging/diverging structures. Adjacent numerical dimensions with a more even distribution between them are likely to have more over-plotting. The seeding algorithm we use here takes these properties into account and also chooses the seeding dimension (the first one or the second one in each axis pair) and the seeding bin (the bin from which a seed record is chosen). The steps are as follows:

1. Determine if axes are numerical or categorical.
2. In case of both categorical axes use over-plotting degree as the criterion, for a numerical and categorical adjacency use the degree of convergence/divergence as the criterion.
3. Compute degree of convergence and divergence for both axes.
4. Convergence and divergence are mirrors of each other. If convergence is greater than divergence, use convergence as the basis for selecting seed, else use divergence. In case of the former, the right dimension is the seeding dimension and in case of the latter, the left dimension is the seed dimension.
5. If both are numerical dimensions check which one of convergence/divergence and over-plotting is greater. Choose that metric to pick the highest frequency bin from which the record is chosen.
6. When the values of either over-plotting degree or degree of convergence/divergence is equal to 1 for all the bins, we build a histogram hierarchically by combining bins that are in the nearest neighborhood: if b adjacent bins have a value of 1, then we put b values in a bin. We select a record from the highest frequency bin in the new histogram. Following this method, we get a coarser histogram and this enables us to avoid selecting a record from a bin which has a sparse neighborhood.

4.1.2 Clustering Algorithm

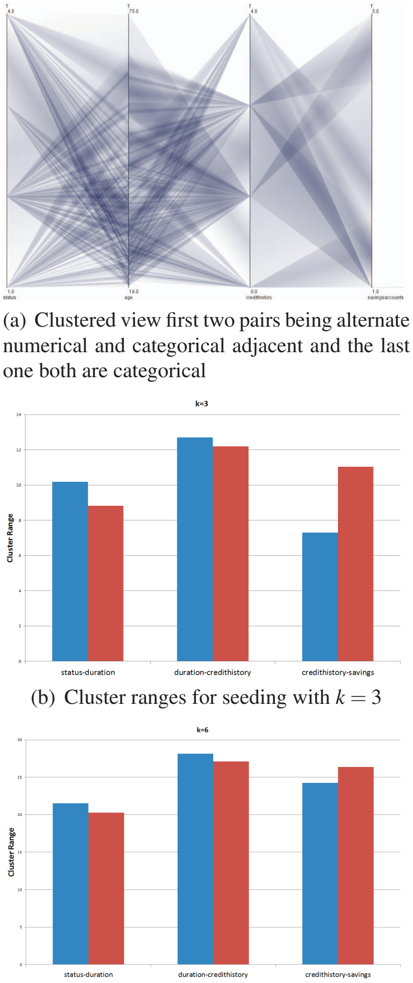
After choosing the initial seed, the clustering algorithm searches for the best record and adds it to the current cluster until the threshold value k is reached. Our method departs from the original algorithm in two ways: a) choice of a distance metric and b) locality-preserving clustering.

Choice of a distance metric: The original algorithm uses an information loss metric based on generalization hierarchy of the attributes as the distance function [7]. Earlier we have argued that a purely data-based clustering approach like this does not work well for visualization (Figure 1). Instead we use the Manhattan distance as the cost function as the goal here is to find visually similar records. The Manhattan distance metric allows us to minimize the vertical distance between the lines on the axes. Manhattan distance translates directly to what we observe as cluster size on the axes.

Locality-preserving clustering: Instead of multi-dimensional clustering, we employ axis-pairwise clustering that takes just local properties into account. This helps to retain the local features between adjacent axes leading to smaller and more discernible clusters and thus less occlusion. In the initial iteration of the clustering, records are grouped into $\lfloor \frac{n}{k} \rfloor$ clusters based on the seeds we chose earlier. After that, there are still n mod k records left. Those are added to existing clusters following the same initial steps, this time minimizing the cost function for a particular cluster and adding it to the cluster which incurs minimum cost. The process is repeated for each axis pair.

4.1.3 How Seeding Criteria Affect Clustering

We choose cluster range as a measure for visual quality of the clusters. Cluster range is measured as the sum of the number of pixels spanned by the records in a cluster on each axis. Figure 5 shows a clustered parallel coordinates configuration with three categorical axes and one numerical axis. The bar graphs show that, in case of both categorical dimensions, cluster ranges for seeding with over-plotting degree are lower than that with degree of convergence/divergence. In case of a numerical-categorical adjacency, the degree of convergence/divergence gives lower cluster ranges. We have observed that the choice of axis is critical in producing smaller clusters, and a wrong choice leads to larger cluster ranges and therefore more occlusion.



(a) Clustered view first two pairs being alternate numerical and categorical adjacent and the last one both are categorical

(b) Cluster ranges for seeding with $k = 3$

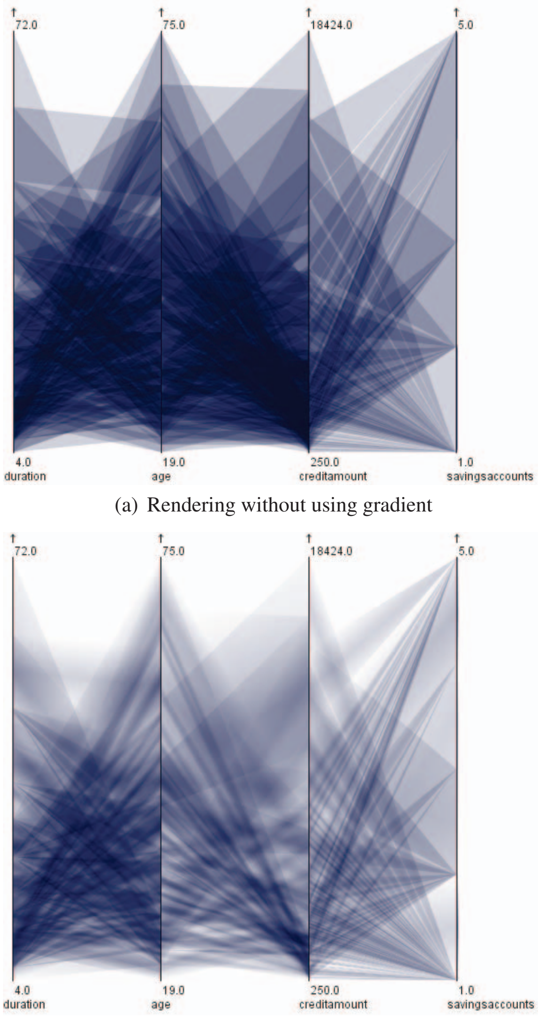
(c) Cluster ranges for seeding with $k = 6$

Fig. 5. Different seed-selection criteria like over-plotting (blue bars) and convergence/divergence (red bars) have a significant effect on the cluster ranges. Lower cluster ranges obscure less of the data and thus help perceive the different trends and patterns that exist between axes.

An issue arising in the rendering of the clusters is that they create a large number of visual artifacts. In addition to the clutter from many overlapping clusters, it is difficult to tell exactly how many clusters are overlapping at each point. Sharp edges of the clusters create a large amount of visual noise that also makes the display harder to read.

We originally used a depth-from-color effect [25], where we ordered the clusters by size and drew the largest ones first. That helps somewhat, and especially makes smaller clusters stand out (which are more relevant, because they provide more specific information). We draw the larger ones at the back and smaller ones in front. Appropriate color emphasizes the differentiation: we make a gradual transition from blue in the background to more orange in the foreground. This helps in distinctly separating the clusters according to their size, but the clutter and noise issues remain.

Based on previous ideas of how to draw clusters in parallel coordinates, in particular Fua et al.'s hierarchical parallel coordinates approach [14], we developed a different way of rendering the clusters that does not use sharp edges. Unlike Fua et al., we do not indicate the cluster centroid with a line. Rather, the color's alpha channel varies perpendicular to the cluster's main direction (the direction of the centroid), creating a fuzzy boundary. Scaling the same range of alpha values over larger clusters produces more fuzziness, while smaller clusters appear sharper (Figure 6(b)). The overall effect of many overlapping clusters is similar to splatting [28].



(a) Rendering without using gradient

(b) Rendering using gradient

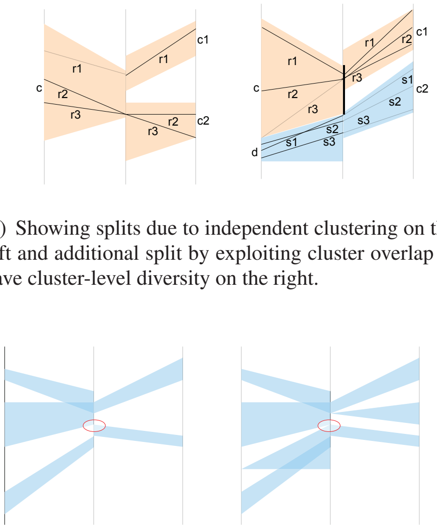
Fig. 6. Demonstrating two different types of rendering for k=5: On the top is the rendering without gradient which produces rigid edges and below is the rendering with gradient which provides better discernibility.

4.2 Cluster Diversity

k-anonymity ensures record-level privacy which is a necessary but not sufficient condition for privacy protection. The k-anonymity method is susceptible to the homogeneity problem, where a cluster based on quasi-identifiers can have the same values for the sensitive attribute and thus the value of the sensitive attributes can be guessed (see also Section 6). Therefore, we apply the concept of l-diversity [21] as a constraint for filtering the clusters that are highlighted on interaction. A privacy-preserving visualization technique differs from its data-mining counterpart because of the added challenge of efficiently handling different interaction conditions. We address this in parallel coordinates by adapting the l-diversity condition to the dynamic user interaction.

4.2.1 Cluster Splits

An artifact of independent clustering between adjacent axes is that clusters are discontinuous and they appear to get split when highlighted as shown in Figure 7(a). On the left we see that the records $r1$ and $r2$ in cluster c are contained in cluster $c1$ on the adjacent axis while the record $r3$ is contained in the cluster $c2$. When c is selected by the user, both $c1$ and $c2$ get highlighted. These splits add to the uncertainty in guessing the exact value of a record.



(a) Showing splits due to independent clustering on the left and additional split by exploiting cluster overlap to have cluster-level diversity on the right.

(b) Cluster splits for different configurations of data. On the left: Splits at the edge compromising privacy with $k = 2$: In the circled area, we know there is a distinct value at that point leading to attribute disclosure. With $k > 2$ it gets difficult to guess the value because that data point can belong to multiple clusters which overlap at that point.

Fig. 7. Demonstrating cluster splits and overlap.

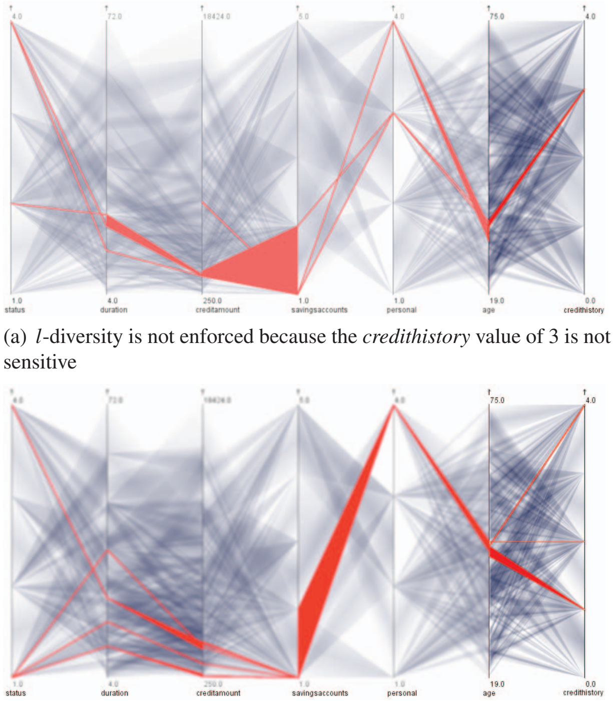
4.2.2 Added Perturbation

A cluster can also be continuous as shown on the right in Figure 7(a), where all $r1$, $r2$ and $r3$ are contained in the same cluster $c1$. In that case, there is no split. But to apply the l-diversity constraint between a quasi-identifier and sensitive dimension, we need a cluster to split into at least l different clusters on the sensitive dimension. For this we use the overlapped pixels on a particular axes (Figure 7(a)) by clusters from adjacent dimension and highlight l different clusters. In this case, $l = 2$, and the cluster $c2$, that is also highlighted on selection of c, is actually a continuity of cluster d. If there are no overlaps, we do not show the clusters on the sensitive dimension. Effectively, we add some random noise to the clusters and alter the actual data values that are perceived. Although this perturbation lowers utility, this is a necessary step to protect the sensitivity of the data values.

4.2.3 Adaptive l-diversity

A dimension, on the whole, might not be sensitive, but some of the values can be. In the German Credit dataset [13], we have four different values for the sensitive dimension, of which only the value 4 (bad credit) is deemed sensitive by the data owner. The l-diversity constraint is only applied in case of a cluster that has a record with that data value (Figure 8). Another example of this scenario would be disease datasets, where cold and flu might not be considered sensitive by a data owner or an individual, but diseases like cancer are sensitive and need privacy-preservation.

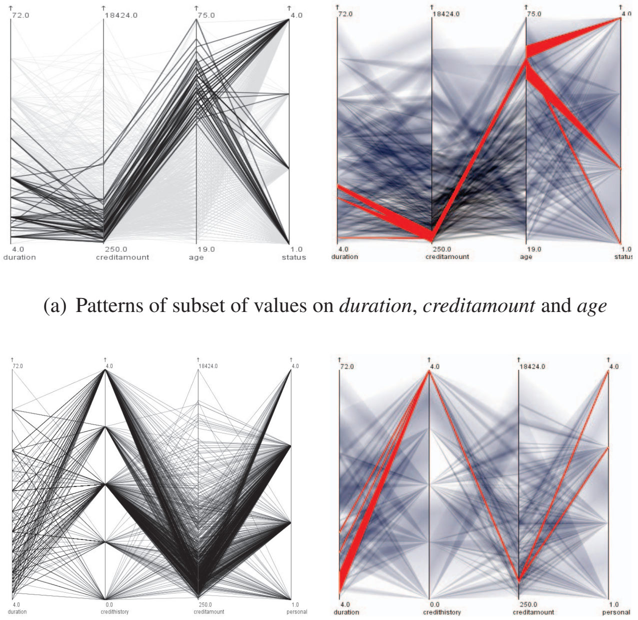
A couple of special cases arise when i) a sensitive dimension has only two values, for example a class variable with a binary yes or no; so in that case a cluster can only be 2-diverse and ii) there are n different values on the sensitive dimension and a cluster has to be $n-$*diverse*. In these cases we do not show any of the highlighted clusters between the quasi-identifier and sensitive dimension. This reduces the utility of the resulting visualization, but imposing that restriction is critical from a privacy-preserving perspective.



(a) *l*-diversity is not enforced because the *credithistory* value of 3 is not sensitive

(b) *l*-diversity is enforced because the *credithistory* value is 4 which is *sensitive*

Fig. 8. l-diversity demonstrated in the rightmost axis pair. Top: l-diversity is not applied between *age* and *credithistory* because the value 0 is not sensitive as determined by the data owner. Bottom: l-diversity is applied because we want to protect clusters which have records with a sensitive 4 value. In this case, *l* is set to be 3, so we cannot tell apart among 3 different values of *credithistory* for the selected cluster.



(a) Patterns of subset of values on *duration*, *creditamount* and *age*

(b) Patterns between *credithistory*, *creditamount* and *personal*

Fig. 9. Illustrating utility of clustered view with respect to different reordering configurations of the raw data.

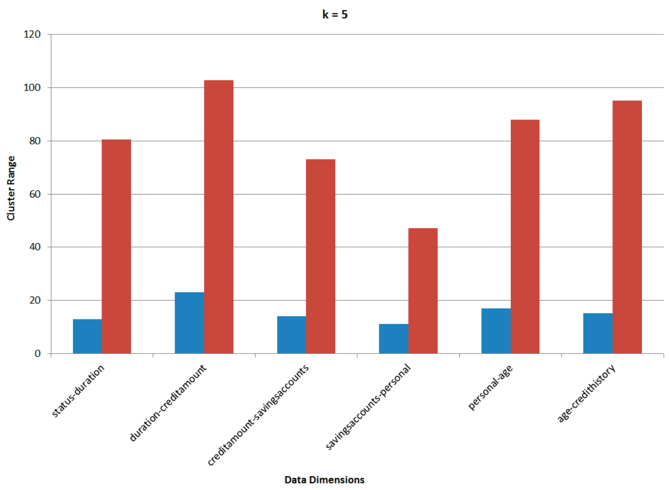


Fig. 10. Multi-dimensional clustering (red bars), as opposed to axis-pairwise clustering (blue bars), leads to high cluster ranges that cause occlusion as shown in Figure 2.

5 Case Study

The German Credit dataset has 1000 instances which classify bank account holders into credit classes *Good* or *Bad*. Each data object is described by 20 attributes that include 13 categorical and 7 numerical attributes. In our experiments we consider the *credithistory* as the sensitive attribute, however we assume that good credit history is not a sensitive value, but bad credit is, so we want to protect the value 4 for *credithistory*. For the other attributes we use a subset of the original attributes. We choose a mix of numerical and categorical attributes among the ones which show maximum information gain and are deemed selectable [24]. Those are: existing checking account *status*, *duration of loan*, *credithistory*, *savings accounts status*, *credit amount*, *personal* status (depends upon gender and marital status).

5.1 Utility of Clustered View

Compared to the conventional data-based approach of multi-dimensional clustering, axis-pairwise clustering produces much more discernible clusters as we had shown in Figure 2. This fact is demonstrated by the graph in Figure 10 where cluster ranges are much smaller in axis-pairwise clustering used in our technique than the multi-dimensional clustering, thereby helping in clutter reduction.

By applying the *k*-members clustering algorithm, we protect the privacy of records, so that the user can only visualize cluster-level information. We cannot show individual record values in the form of lines, but the overall multivariate distribution among the different dimensions can still be visualized. In Figure 3 we see that for different values of *k* most relationships in the raw data are discernible in the clustered view.

We also demonstrate by showing different configurations of axis reordering that a user can still see the different trends and patterns. Figure 9(a) and Figure 9(b) show two different configurations of raw data on the left and the corresponding clustered configurations on the right. Patterns can also be seen between: a) *duration*, *creditamount* and *age*: low duration values corresponding to low credit amount and higher values of age are visible on mouse-over interaction in the clustered view; and b) *credithistory*, *creditamount* and *personal* exhibit a band of clusters that are seen in both views.

5.2 Privacy protection

Cluster splits and overlaps are an artifact of axis-pairwise clustering. As mentioned in Section 4.2, the splits lead to uncertainty. Splits are especially pronounced in the case of numerical axes and higher values of *k*. Even if we know the exact values of the quasi-identifiers, there are splits among the different axes which introduce uncertainty for guessing the exact value for a particular record (Figure 8). For the

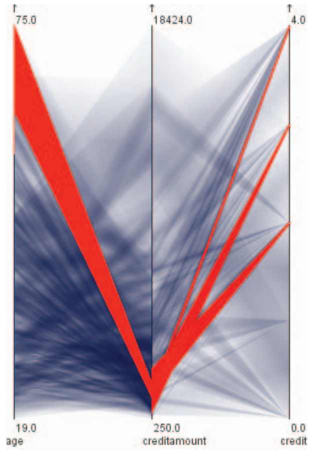


Fig. 11. High mutual information may give away too much information and it becomes easier to narrow down the guesses.

adjacency condition of a quasi-identifier and a sensitive attribute, we selectively enforce l-diversity. In Figure 8(a) we retain the highlighting of clusters based on just k-anonymity because the data owner has determined that only persons with a poor credit (in this case with a *credithistory* value of 4) should be protected. In Figure 8(b) we show that l-diversity is applied and we cannot tell three different values of *credithistory* apart.

In case of reordering, if the sensitive dimension is not the last axis, then we have cluster splits on both sides of that dimension. But this reduces the utility to a great extent because we have the diversity on either side of the sensitive dimension leading to a lot of cluster splits. This reduces the meaningfulness of such a view. In the *German credit* dataset there is a single sensitive attribute, but our model can easily handle the case of multiple sensitive attributes by ensuring sufficient diversity in the corresponding axis pairs.

6 Privacy Attacks

The clustering mechanism in our system is similar compared to the k-members clustering commonly used in data mining. From a privacy point-of-view it is beneficial that an unauthorized user can only read the data in terms of pixel coordinates and not in a tabular form; the loss of precision makes it difficult to guess the data values and gain knowledge from the visualization. The cluster splits further make it difficult to breach privacy. On the interaction side, we add sufficient noise so that finding the sensitive values is very difficult. In the following sections we describe some of the attack scenarios, how they are handled and potential pitfalls of our approach.

6.1 Attack Conditions

The different attack conditions and disclosure risk scenarios are described as follows:

Re-identification risks: Two types of re-identification types have been identified in the literature [11, 18]: a) identity disclosure: this occurs if the intruder is able to assign a particular identity to any record in the data and b) attribute disclosure: this occurs when an intruder learns something new about the sensitive attribute values that can be linked to an individual, without knowing which specific record belongs to that individual. For example, if all 30-year-old male patients in the disclosed database who live in a particular area had a prescription for diabetes, then if the intruder knows that John is 30 years old and lives in that particular area, he or she will learn that John is a diabetic, even if the particular record belonging to John is unknown. Identity disclosure typically needs external information like quasi-identifiers, so medical databases try to guard against this type of disclosure. Attribute disclosures are likely to occur in corporate data, which are not generally associated with external information.

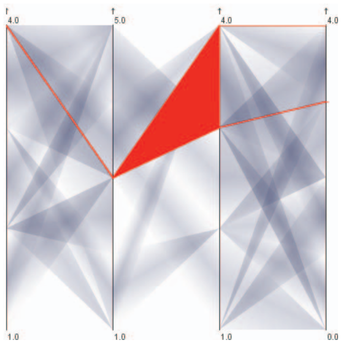


Fig. 12. A configuration like this is avoided because categorical dimensions tend to produce clusters with splits at the edges, which when placed adjacent to a sensitive dimension, may lead to disclosure.

Potential attack scenarios: Two ways to break the privacy of a sanitized dataset suggested in the literature [17] also apply in case of visualization. The first one is called *prosecutor re-identification scenario*, where an intruder (e.g., a prosecutor) knows that a particular individual (e.g., a defendant) exists in an anonymized database and wishes to find out which record belongs to that individual. In the second one, known as the *journalist re-identification scenario*, an attacker tries to re-identify an arbitrary individual. The intruder does not care which individual is being re-identified, but is only interested in being able to claim that privacy breach is possible.

We describe the following possible cases of attack with respect to these disclosure risks and attack scenarios:

6.2 Split Clusters

Clusters splitting at the edges are vulnerable to disclosure, especially for smaller values of k. As shown in Figure 7(b), on the left the clusters split at the edges and clearly there is a single record with those two border values in the record. This relates to the journalist attack scenario, where it is not necessary for an attacker to know if an individual exists in the database. They can still exploit this kind of configuration leading to attribute disclosure. When clusters get larger, guessing the precise values at splitting points becomes harder, because the value at the edge might come from a different cluster. Larger k leads to more splits and thereby more uncertainty, but reduces the utility: a trade-off that is difficult to model.

6.3 Different Reordering Configurations

The common ways to interact with parallel coordinates are to: a) hover over different lines to trace their path across the different dimensions, b) reorder the axes to see the patterns for different configurations of adjacent axes, and c) brush over different records on one axis to see the patterns of the subset of the data on other axes. Since the clusters already represent aggregated values, we ignore brushing for this paper and focus on the first two aspects.

High mutual information between two data dimensions A and B imply that the uncertainty about A is highly reduced in presence of B. Mutual information has been shown as an effective screen-space metric for parallel coordinates [9]. In the privacy context, we use lack of mutual information as a measure of high uncertainty [4]. If there is high mutual information between two axes like *creditamount* and *credithistory* (Figure 11), there might be a skewed distribution which makes it easier for an intruder to breach the privacy if he has some background knowledge about the person. This relates to attribute disclosure and is susceptible to both attack scenarios. When reordering the axes, we enforce the constraint that the quasi-identifier attribute with the least mutual information should be adjacent to the sensitive attribute. This ensures that even with interaction, the attacker cannot exploit the strong correlative effect between the two dimensions.

In case of a mix of categorical and numerical attributes, our technique puts numerical and categorical axes alternately, adjacent to each other. We avoid a configuration like the one shown in Figure 12 for two reasons. Firstly, cluster ranges can be very high between two categorical dimensions reducing the utility. Secondly, placing a categorical attribute adjacent to a sensitive dimension can reduce the intended privacy because the cluster edges represent actual data values and most values for a categorical cluster may lie on its edge and give away information. This relates to the prosecutor identification scenario, where an attacker knows an individual exists in the database and can select the appropriate cluster to gain knowledge about the attributes that describe the individual.

6.4 Cluster Attacks

Given the client-server nature of the system, an attacker could pretend to be a visualization client and repeatedly retrieve clusters with different settings. The potential information gain from such an attack is small, however, and there are simple precautions we can take to make them even less productive.

Forced Reclustering. By requesting clusters for different resolutions, an attacker can collect and analyze differences in the clusters in order to gain more information. Due to the seeding strategy employed and the pixel-based binning, some clusters will be different given different numbers of vertical pixels. These differences are comparatively small however, and only lead to a limited number of additional values being accessible as cluster boundaries. The values are also rounded to the nearest pixel coordinate, so exact values cannot be recovered.

Resolution Limits. Rounding to pixel values is only effective as long as the number of pixels is relatively small. If the client can specify an arbitrarily large number as the number of pixels, the rounding is effectively circumvented (though the clusters do not significantly change). A simple precaution therefore is to limit acceptable vertical resolutions to a reasonably small number, such as 500. Requests for more vertical pixels will simply be handled as if the maximum had been requested.

Step Size Limits. To limit the potential information gain from forced rounding errors, the data owner can choose to only allow a minimal step size in the number of pixels. Rather than being able to request 501, 502, 503, etc. pixels, only multiples of 50 or 100 would be possible this way. This limits the usefulness somewhat because only certain client window sizes will be supported well, but it also reduces the potential knowledge gain for an attacker.

7 Conclusions and Future Work

In the work reported here, we have presented a privacy-preserving visualization technique based on the k-anonymity and l-diversity approaches. Conventional PPDM techniques do not work well in the visualization context, leading to visualizations that are of very little use. Screen-space sanitization takes the properties of the visual representation into account, and thus retains much more of the visual structure in the data. Our technique is adaptable to user interactions and takes potential attack scenarios into account.

As a next step we will conduct a comprehensive user study to test our system and accommodate any unforeseen attack scenarios which we are currently not addressing in our technique. Moreover, we are currently working on developing information-theoretic measures to quantify privacy and utility in the visualization space that would help us address the optimization problem involving these two factors. Finally, our goal is to extend our privacy-preserving approach based on screen-space metrics and apply it to other visualization techniques.

References

[1] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation algorithms for k-anonymity. In *Journal of Privacy Technology*, 2005.

[2] R. Agrawal and R. Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.

[3] E. Bertino, D. Lin, and W. Jiang. A Survey of Quantification of Privacy Preserving Data Mining Algorithms. *Privacy-Preserving Data Mining*, pages 183–205, 2008.

[4] M. Bezzi. An entropy based method for measuring anonymity. In *Third International Conference on Security and Privacy in Communications Networks and the Workshops, 2007.*, pages 28–32. IEEE, 2008.

[5] J. Brickell and V. Shmatikov. The cost of privacy. *Knowledge discovery and data mining*, 2008.

[6] S. Bu, L. V. Lakshmanan, R. T. Ng, and G. Ramesh. Preservation Of Patterns and Input-Output Privacy. *23rd International Conference on Data Engineering*, pages 696–705, Apr. 2007.

[7] J. Byun, A. Kamra, E. Bertino, and N. Li. Efficient k-anonymization using clustering techniques. In *Proceedings Database Systems for Advanced Applications*, pages 188–200. Springer, 2007.

[8] C. Clifton, M. Kantarcioglu, and J. Vaidya. Defining privacy for data mining. In *NSF Workshop on Next Generation Data Mining*, pages 126–133, 2002.

[9] A. Dasgupta and R. Kosara. Pargnostics: screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1017–26, 2010.

[10] A. Dasgupta and R. Kosara. Privacy-preserving data visualization using parallel coordinates. In *Proceedings Visualization and Data Analysis (VDA)*, pages 78680O–1–78680O–12, 2011.

[11] G. T. Duncan and D. Lambert. Disclosure-limited data dissemination. *Journal of the American Statistical Assn.*, 81(393):pp. 10–18, 1986.

[12] C. Dwork. Differential privacy. In *ICALP*, pages 1–12. Springer, 2006.

[13] A. Frank and A. Asuncion. UCI machine learning repository. http://archive.ics.uci.edu/ml, 2010.

[14] Y.-H. Fua, M. O. Ward, and E. A. Rundensteiner. Hierarchical parallel coordinates for exploration of large datasets. In *Proceedings Visualization*, pages 43–50. IEEE CS Press, 1999.

[15] A. Inselberg and B. Dimsdale. Parallel coordinates: A tool for visualizing multi-dimensional geometry. In *IEEE Visualization*, pages 361–378. IEEE CS Press, 1990.

[16] J. Johansson, P. Ljung, M. Jern, and M. Cooper. Revealing structure within clustered parallel coordinates displays. In *Proceedings Information Visualization*, pages 125–132, 2005.

[17] F. D. K. El Emam. Protecting privacy using k-anonymity. *Journal of the American Medical Informatics Association*, 15:627–637, 2008.

[18] D. Lambert. Measures of disclosure risk and harm. *Journal of Official Statistics*, 9:313–331, 1993.

[19] J. Li, J.-B. Martens, and J. J. van Wijk. Judging correlation from scatterplots and parallel coordinate plots. *Information Visualization*, 9(1):13–30, 2010.

[20] P. Luzzardi, C. Freitas, R. Cava, G. Duarte, and M. Vasconcelos. An Extended Set of Ergonomic Criteria for Information Visualization Techniques. In *Proceedings Computer Graphics And Imaging*, pages 236–241, 2004.

[21] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. l-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):3, 2007.

[22] A. Meyerson and R. Williams. On the complexity of optimal k-anonymity. In *Proceedings Principles of Database Systems*, pages 223–228. ACM, 2004.

[23] M. Novotny and H. Hauser. Outlier-preserving focus+context visualization in parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):893–900, 2006.

[24] P. O'Dea, J. Griffith, and C. O'Riordan. Combining feature selection and neural networkds for solving classification problems. *Irish Conference on Artifical Intelligence & Cognitive Science*, pages 157–166, 2001.

[25] H. Piringer, R. Kosara, and H. Hauser. Interactive focus+context visualization with linked 2d/3d scatterplots. In *Coordinated and Multiple Views in Exploratory Visualization*, pages 49–60, 2004.

[26] L. Sweeney. k-Anonymity: A Model for Protecting Privacy. *IEEE Security And Privacy*, 10(5):1–14, 2002.

[27] S. F. V. Ciriani, S. De Capitani di Vimercati and P. Samarati. k-anonymous data mining: A survey. In *Privacy-Preserving Data Mining: Models and Algorithms*, pages 105–136. Springer-Verlag, 2007.

[28] H. Zhou, W. Cui, H. Qu, Y. Wu, X. Yuan, and W. Zhuo. Splatting the Lines in Parallel Coordinates. *Computer Graphics Forum*, 28(3):759–766, 2009.

[29] H. Zhou, X. Yuan, H. Qu, W. Cui, and B. Chen. Visual clustering in parallel coordinates. *Computer Graphics Forum*, 27(3):1047–1054, 2008.

[30] C. Ziemkiewicz and R. Kosara. Embedding Information Visualization Within Visual Representation. *Advances in Information and Intelligent Systems*, pages 307–326, 2010.