Cartification: A Neighborhood Preserving Transformation for Mining High Dimensional Data

Emin Aksehirli[•], Bart Goethals[•], Emmanuel Müller[•○] and Jilles Vreeken[•]
[•]University of Antwerp, Belgium Email: firstname.lastname@uantwerpen.be
[○]Karlsruhe Institute of Technology, Germany Email: emmanuel.mueller@kit.edu

***Abstract*—The analysis of high dimensional data comes with many intrinsic challenges. In particular, cluster structures become increasingly hard to detect when the data includes dimensions irrelevant to the individual clusters. With increasing dimensionality, distances between pairs of objects become very similar, and hence, meaningless for knowledge discovery.**

In this paper we propose *Cartification*, a new transformation to circumvent this problem. We transform each object into an itemset, which represents the neighborhood of the object. We do this for multiple views on the data, resulting in multiple neighborhoods per object. This transformation enables us to preserve the essential pairwise-similarities of objects over multiple views, and hence, to improve knowledge discovery in high dimensional data. Our experiments show that frequent itemset mining on the cartified data outperforms competing clustering approaches on the original data space, including traditional clustering, random projections, principle component analysis, subspace clustering, and clustering ensemble.

***Keywords*—transformation; high dimensional data; clustering; frequent itemset mining; subspace projections**

I. Introduction

Many knowledge discovery tasks rely on some notion of similarity between objects. For example, in clustering, the goal is to group similar objects into a cluster, while separating dissimilar objects into different clusters. For many real world applications, such as in customer segmentation, gene expression analysis, and health surveillance, we observe that their inherently high dimensional data spaces hinder the application of traditional clustering algorithms. In particular, one of the main problems with high dimensional data is that the distances between pairs of objects, measured over the full data space, rapidly grow more and more alike for higher numbers of dimensions—which essentially renders the notion of neighborhoods meaningless in the high dimensional space [4]. This effect is well-known as the *curse of dimensionality*, and affects all methods that consider the full data space.

Loosely speaking, on the one hand we find existing approaches that aim to alleviate this problem by finding subspaces over which to calculate similarities. Examples include unsupervised feature selection [7], dimensionality reduction [14], meta-distance measures [13], or subspace search [5]. Each of these proposes a fixed notion of similarity and is not flexible in adapting to the hidden cluster structure, or, local neighborhoods, of the data. On the other hand we find subspace clustering, which proposes to detect individual projections for each relevant local neighborhood [24], [22]. Unfortunately, however, finding the relevant dimension sets is a combinatorial problem that adds to the intrinsic complexity of the cluster analysis itself. Overall, existing subspace approaches lack efficient computation [2], [15], [20], are proven to be NP-hard [21], or lack local subspace projections [5], [16], [23].

In this paper we propose a new transformation to circumvent these problems. In particular, we tackle the dual challenge of (1) considering multiple local subspaces and (2) exponential complexity of the search space. We propose to not directly consider the original data, but to transform it such that all local neighborhoods are preserved, allowing for example, clustering data in multiple projections of the data. Intuitively, we consider multiple views on the original data space (i.e., a high dimensional database) and transform these views into a novel itemset space. We call our transformation *cartification*, following the illustrative idea of building shopping carts, and storing the bought products in the transactional data scheme known from frequent itemset mining.

In a nutshell, we transform each object into a *cart* containing all k nearest neighbors of that object. For each object, we exploit multiple views on the original data space by using different (dis)similarity measures for our neighborhood assessment. By using multiple views on the data, neighborhood information, i.e. similarity, is preserved with increasing number of dimensions in at least some of these views. This leads to a robust preservation of local neighborhoods in subspace projections of the original data space. Applying frequent itemset mining then reveals hidden cluster structures. Frequency of items, and itemsets, implicitly measures the similarity of co-occurring objects, and hence, automatically excludes irrelevant dimensions with noisy data distributions. That is, in our transformed data, an itemset with a high frequency means that the corresponding objects share a local neighborhood. By checking which transactions support the itemset, we can easily infer the subspace in which these objects are clustered. Moreover, as neither noisy objects nor irrelevant dimensions will add significantly to the supports, they do not interfere with the detectability of subspace clusters.

In our experiments we focus on clustering as the underlying data mining task and show that our transformation is more stable for cluster detection than using the original data space. We are able to detect clusters and their individual subspace projections. For each cluster, our method excludes locally irrelevant dimensions, and hence, is robust w.r.t. increasing numbers of dimensions. Overall, we show that our transformation outperforms traditional clustering, random projections, principle component analysis, subspace clustering methods, and clustering ensemble on data projections.

II. Related Work

In this paper we consider clustering as an application domain for our transformation. Traditional clustering methods [12] have been shown to be affected by increasing numbers of dimensions in real databases. Clustering models that measure similarities over all given dimensions (i.e., the full data space) are hindered by irrelevant dimensions and loss of contrast in high dimensional data spaces [4]. In particular, traditional distances such as Euclidean or L_p norms are affected by the concentration of norms with increasing number of dimensions [17]. We address this general problem and propose a data transformation for more robust cluster detection in subspaces of high dimensional data.

Dimensionality reduction techniques [14], [17] or unsupervised feature selection [7], [25] are general techniques for projecting a database to single lower-dimensional projections. Such projections, however, incur information loss. Moreover, these methods are limited to single views of the data. That is, clusters are detected only in a single projection although other projections might reveal additional cluster structures and novel knowledge as well. We propose a different processing, which captures multiple views on the data, and hence, allows for cluster detection in arbitrary subspace projections.

Subspace clustering [24], [22] searches for clusters in subsets of dimensions. However, every clustering model needs special adaptation. For example, the well-known K-Means algorithm has been extended to subspace projections, namely Proclus [1]. Other clustering notions such as DBSCAN need a totally different processing [15]. Recently, more general approaches have been proposed for subspace search [5], [16], [23], leaving the discovery of clusters or outliers to specialized algorithms. However, all of these methods are computationally expensive as they search in an exponential set of subspaces. With our method we follow the idea of mining multiple subspace projections. However, in contrast to the existing approaches we aim at a general transformation of the problem into a single itemset space that preserves neighborhood similarity over all subspaces.

As we incorporate multiple views into one, consensus methods are related. A naive approach is to cluster each of the views, and then use clustering ensemble methods [9], [26], [8] to derive a consensus solution. We include this as a baseline in our experiments. Such clustering ensemble techniques, however, focus on the combination of different results and do not address the high dimensionality problem systematically. In contrast to the consensus approach, we start at one-dimensional projections and systematically search for the correlated dimensions by itemset mining on the transformed data space. This allows us to find multiple overlapping clusters in different subspaces, in contrast to the single clustering obtained by consensus techniques.

III. Cartification

In this section we explain our transformation from a high dimensional numeric space into a transaction database. In particular we show how this transformation preserves neighborhood information by co-occurrence of objects.

A. Basic Notions

Our transformation turns the neighborhood information of a *high dimensional data space* into a *transaction database*.

INPUT: HIGH DIMENSIONAL NUMERIC DATABASE
Let $\mathcal{A} = \{A_1, \dots, A_d\}$ be a set of d dimensions. A data object, o, is defined as a tuple of values over $\mathcal{A}$. A d-dimensional database, $\mathbf{D}$, is a collection of n data objects such that every object in $\mathbf{D}$ is represented by a d-dimensional vector $\mathbf{o} \in \mathbb{R}^d$.

The curse of dimensionality [4] intuitively states that if the number of dimensions increase, distances between objects grow more and more alike. Thus, local neighborhood becomes meaningless for a large number of dimensions. Our transformation tries to preserve the local neighborhoods inside clusters that show high similarity. Therefore, we use multiple views on the data by using different similarity measures for each view.

We use the notation of $\mathcal{M}$ as a set of different measures to induce all of these views. An individual measure m is simply a function that represents the similarity relation between pairs of objects. For example, we use m_S for a measure over arbitrary attribute set $S \subseteq \mathcal{A}$. Since we exploit the relative similarities of objects, a measure can be a dissimilarity measure (e.g. Euclidean distance), or a similarity measure (e.g. Jaccard similarity), as long as the objects can be ordered by their pairwise similarity. Intuitively, one can think of m as the projection scheme, which, obviously, can model any arbitrary transformation of the data.

OUTPUT: TRANSACTION DATA REPRESENTATION
For a set of items $\mathcal{I}$, an *itemset* X is defined as $X \subseteq \mathcal{I}$. A transaction $t = (tid, X)$ is a pair of unique transaction id and an itemset. A transaction database $\mathcal{T}$ is a set of transactions over $\mathcal{I}$. The *support* of an itemset X in database $\mathcal{T}$ is the number of transactions in $\mathcal{T}$ in which X occurs, i.e.,

$$supp_{\mathcal{T}}(X) = |\{t \in \mathcal{T} \mid X \text{ occurs in } t\}|.$$

An itemset is called *frequent* if its support is larger than some user-defined threshold called the *minimum support* or $minsup$.

B. Formal Transformation

Cartification transforms the original dataset into a coalition of different views on the data, such that each of them contributes with the neighborhood information of a given similarity measure.

Definition 1 (View of an object): Let $\mathbf{o} \in \mathbf{D}$, $\mathbf{p}_i \in \mathbf{D}$, $|\mathbf{D}| = n$ the number of objects in $\mathbf{D}$, and m a dissimilarity measure. The cart $C_m(\mathbf{o})$ of object $\mathbf{o}$ is a tuple of its neighbors, ordered descending by their similarity to $\mathbf{o}$. Formally,

$$C_m(\mathbf{o}) = (\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_n)$$

with, $m(\mathbf{o}, \mathbf{p}_i) \leq m(\mathbf{o}, \mathbf{p}_j) \iff i < j$.

Neighborhood information on objects $\mathbf{p}$ that are far away from $\mathbf{o}$, contributes very little towards the discernibility of local cluster structures, therefore, far away neighbors can be excluded from the carts. That is, carts can be trimmed to include just the *top-k Nearest Neighbors* (*kNN*) of object $\mathbf{o}$ w.r.t a similarity measure m.

Definition 2 (View of a measure): A local view C_m is a reflection of neighborhood information from the view of a single similarity measure m:

$$\mathcal{C}_m = \bigcup_{\mathbf{o} \in \mathbf{D}} C_m(\mathbf{o})$$

Since using the transformation of only one measure would be too restrictive, we employ multiple views to collect information about neighborhoods.

Definition 3 (Transformed database): Let $\mathcal{M}$ be the set of measures, and $\mathcal{C}_m$ a *local view* of the similarity measure m, then a cartified database $\mathcal{C}$ is defined as

$$\mathcal{C} = \bigcup_{m \in \mathcal{M}} \mathcal{C}_m$$

After the transformation, neighbors of each object for each measure become clearly visible on their corresponding carts. Aggregating this neighborhood information from a selection of measures will reveal the cluster structures. We utilize the intrinsic property of neighborhoods, that is, the strong dependence of objects with high similarity inside a cluster is reflected in their co-occurrence in both their own neighborhoods, as well as in the neighborhoods of objects close-by.

C. Co-occurrence of objects

Frequent occurrence of an object set X in $\mathcal{C}$ indicates that these objects are often spotted in the same neighborhood together in the original space. That is, they are likely to be related. Every measure reflects a different set of relations in the data and cartification extracts the neighborhood information for each similarity measure separately. As such, extra measures add information; the accuracy of individual measures is not disturbed by irrelevant views.

Definition 4 (co-occurring object set): The *occurrence* of an object set X is the number of carts $C \in \mathcal{C}$ that are a superset of X. Formally,

$$\mathit{occurrence}(X) = |\{C \mid X \subset C, C \in \mathcal{C}\}|$$

A cluster X is then defined as a set of co-occurring objects w.r.t. parameter *minsup*:

$$\mathit{occurrence}(X) \geq \mathit{minsup}$$

We measure the occurrence of an object set X by simply counting the co-occurrences of these objects in any cart in our cartified database $\mathcal{C}$. This can be considered as an implicit similarity assessment on the objects in X. Clustered objects have to share a large amount of objects in their respective neighborhoods. Hence they show high mutual similarity to each other. Please note, that this clustering criterion can be considered as a generalization of Shared Nearest Neighborhood (SNN) clustering [13]. In contrast to SNN, we use multiple local views, which in turn better grasp the structures that exist only in data projections. Furthermore, we keep the order of the neighbors, that can be used to improve the following analysis steps after our data transformation.

Formally, clustered objects $X = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ will be clearly separated from residual objects $\mathbf{q} \notin \mathbf{D} \setminus X$ in at least some measures of $\mathcal{M}$:

$$\mathit{occurrence}(X) \geq \mathit{minsup} \overset{\text{(Def. 4)}}{\Longleftrightarrow}$$
$$\exists V \subseteq \mathbf{D} \times \mathcal{M} \wedge |V| \geq \mathit{minsup}$$
$$\forall (\mathbf{o}, m) \in V : X \subset C_m(\mathbf{o}) \overset{\text{(Def. 1)}}{\Longleftrightarrow}$$
$$\exists V \subseteq \mathbf{D} \times \mathcal{M} \wedge |V| \geq \mathit{minsup}$$
$$\forall (\mathbf{o}, m) \in V : m(\mathbf{o}, \mathbf{p}_j) < m(\mathbf{o}, \mathbf{q}) \quad \forall \mathbf{p}_j \in X \wedge \mathbf{q} \notin C_m(\mathbf{o})$$

Thus, there exists a set of carts that reflect the similarity of all clustered objects X in some of the views $\mathcal{M}$. The similarity of a cluster is preserved in these views, i.e., objects $\mathbf{p}_j$ are clustered together in the neighborhood of a central object $\mathbf{o}$, while other objects $\mathbf{q}$ are clearly separated from this central object. Please note that there might be other contradicting views, however, these do not have a negative effect on the number of co-occurrences of X. Therefore, similarity and neighborhoods are preserved. In contrast to this, the original data space is effected by irrelevant dimensions, and thus, does not reveal any cluster structures.

D. Instantiation: One-Dimensional Data Projections

In order to preserve neighborhoods and circumvent the curse of dimensionality [4], we follow the idea of lower-dimensional projections as meaningful instantiation of different measures. Lower-dimensional projections have shown to be effective for query processing [11] and subspaces clustering [24] in high dimensional databases. They capture neighborhood information w.r.t. different views on the data and provide us the required similarity measures on the database. However, in our case we do not need complex analysis and exponential overhead of subspace clustering [2], [15], [20]. As minimum requirement the similarity measures have to identify neighborhoods for each individual dimension only. Then possible correlation in higher-dimensional projections is assessed by our co-occurrence.

We simply use $\mathcal{M} = \{m_1, \ldots, m_d\}$ as initial set of measures. Each measure m_i represents a one-dimensional projection of the data w.r.t. attribute A_i. It is simply computed by the Euclidean distance in this dimension. Using these measures, we allow for similarity assessment in any projection of the data. A co-occurring object set X can show mutual similarity in an arbitrary subspace $S \subseteq \mathcal{A}$:

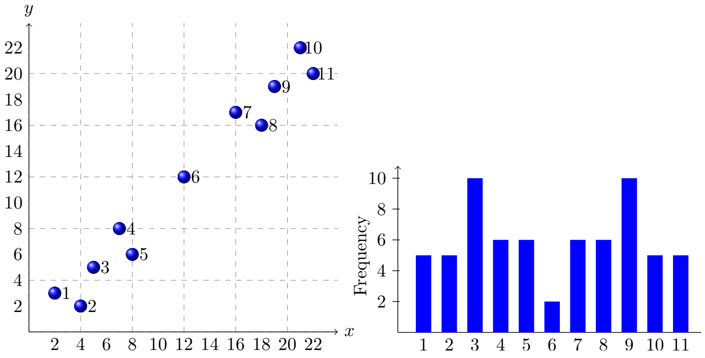
$$\exists A_i \in S \;:\; X \subset C_{m_i}(\mathbf{o}) \quad \forall \mathbf{o} \in X \quad \Longleftrightarrow$$
$$|o_i - p_i| < |o_i - q_i| \quad \forall \mathbf{o}, \mathbf{p} \in X \wedge \mathbf{q} \notin X$$

Please note that we might consider more complex similarity measures as alternative to the current one-dimensional instantiation, e.g. by using correlation analysis or subspace search [5], [16], [23]. Our current solution abstracts from such instantiations and provides the general processing for any set of similarity measures $\mathcal{M}$.

Let us now give an example of how we can identify central elements of a cluster. Assume we are given the two-dimensional dataset with $\mathcal{A} = \{x, y\}$ and data objects as shown in Figure 1a. Using Euclidian distance on separate dimensions as dissimilarity measures, the CARTIFY algorithm

Table I: Cartification of the data in Figure 1a, for $k = 3$.

x	cart	y	cart
1	$\{1, 2, 3\}$	1	$\{1, 2, 3\}$
2	$\{1, 2, 3\}$	2	$\{1, 2, 3\}$
3	$\{2, 3, 4\}$	3	$\{1, 3, 5\}$
4	$\{3, 4, 5\}$	4	$\{3, 4, 5\}$
5	$\{3, 4, 5\}$	5	$\{3, 4, 5\}$
6	$\{5, 6, 7\}$	6	$\{4, 6, 8\}$
7	$\{7, 8, 9\}$	7	$\{7, 8, 9\}$
8	$\{7, 8, 9\}$	8	$\{7, 8, 9\}$
9	$\{8, 9, 10\}$	9	$\{7, 9, 11\}$
10	$\{9, 10, 11\}$	10	$\{9, 10, 11\}$
11	$\{9, 10, 11\}$	11	$\{9, 10, 11\}$



(a) Example dataset

(b) Item Frequencies for $k = 3$

Figure 1: Cartification Example

(cf. Section III-E) creates a database where each cart is composed of the items ordered by one-dimensional distances to each item.

For this example, instead of taking into account detailed ordering in the carts, we truncate each transaction to the same size, k, disregard the inner order, and treat each transaction as a set. This process converts the cartified database to a regular transactional database where each transaction is the k-nearest neighbors of each item on each dimension. For example, the resulting database for $k = 3$ is shown in Table I. The first two columns show the cartification for the x-dimension, and the second two columns for the y-dimension. Every row corresponds to the cart generated from one of the data points. For instance, for point 3, the three nearest neighbors in the x-dimension are points $2, 3$, and 4, while for the y-dimension these are $1, 3$, and 5.

In Figure 1b, we plot the frequencies of all items (points) in this cartified database. The plot shows that there are two points, 3 and 9, with a high frequency. As Figure 1a shows, these points are in the centers of the clusters $\{1, 2, 3, 4, 5\}$ and $\{7, 8, 9, 10, 11\}$ respectively. Additionally, point 6 has a very low frequency which corresponds to the point being an outlier w.r.t. all other points in the figure.

E. Algorithm and Complexity Analysis

Suppose we are given a database $\mathbf{D}$ over attributes $\mathcal{A}$ and a set of similarity measures $\mathcal{M}$, defined over $S \subseteq \mathcal{A}$. The pseudo-code is given as Algorithm 1.

Algorithm 1 The CARTIFY Algorithm

```
Input: A database $\mathbf{D}$, set of similarity measures $\mathcal{M}$
Output: The cartified database $\mathcal{C}$ of $\mathbf{D}$ over $\mathcal{M}$
$\mathcal{C} \leftarrow \emptyset$
for each $m \in \mathcal{M}$ do
    $\mathcal{C}_m \leftarrow \emptyset$
    for each $\mathbf{o} \in \mathbf{D}$ do
        $C \leftarrow$ sort $\mathbf{p} \in \mathbf{D}$ according to $m(\mathbf{o}, \mathbf{p})$
        $\mathcal{C}_m \leftarrow \mathcal{C}_m \cup C$
    $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_m$
return $\mathcal{C}$
```

Theoretically, space requirement of CARTIFY algorithm is $n \times n \times |\mathcal{M}|$, where n is the number of items and $|\mathcal{M}|$ is the number of measures. However, for practical applications, it is not required to create the full-length carts. Instead, carts can be trimmed to a constant value to limit the size of the database.

Worst case time complexity of CARTIFY is $\mathrm{O}(n^2 \log(n)|\mathcal{M}|)$. For some measures, e.g., 1D projections, sorting can be done per measure, in the outer loop, which results in a time complexity of $\mathrm{O}(n \log(n)|\mathcal{M}|)$. Moreover the processing of the cartification can easily be parallelized over shared memory or distributed systems.

IV. AN APPLICATION OF CARTIFICATION TO SUBSPACE CLUSTERING

As an example of how the information in the transformed data can be used, in this section we introduce CARTICLUS. CARTICLUS, short for cartification-based subspace cluster detection, is an algorithm for mining subspace clusters by means of a straightforward itemset mining based approach. Even though relatively simple method, the experiments in Section V show that, it outperforms competing approaches on high dimensional data.

Measures that are used for transformation of the original datasets are Euclidean distances on one-dimensional projections. After the transformation, carts are trimmed to a user-specified length k. To detect subspace clusters we then stochastically generate n maximal frequent itemsets. After a simple post-processing, we report the sufficiently frequent object sets as subspace clusters.

We give the pseudo-code of CARTICLUS as Algorithm 2. It has two input parameters: k, the size of each cart and $minsup$, the minimum number of co-occurrences of an object set in the cartified database $\mathcal{C}$. Note that, since a sample of the maximal frequent itemsets (MFIs) is enough, CARTICLUS efficiently mines a sample of the MFIs instead of generating all of them (Lines 3–12). Our experiments show that MFIs tend to identify *subsets* of clusters. In order to be able to report the key cluster structure without redundancy, CARTICLUS merges MFIs that are highly similar to each other, lines 13–20. The dimensions that an object set has non-zero support are reported as the subspaces of the corresponding cluster (line 19–20).

CARTICLUS takes cartified database as input. Therefore, additional analysis tasks, e.g., using different parameters or even different mining algorithms, can be performed on the same cartified database, with a minimal data processing.

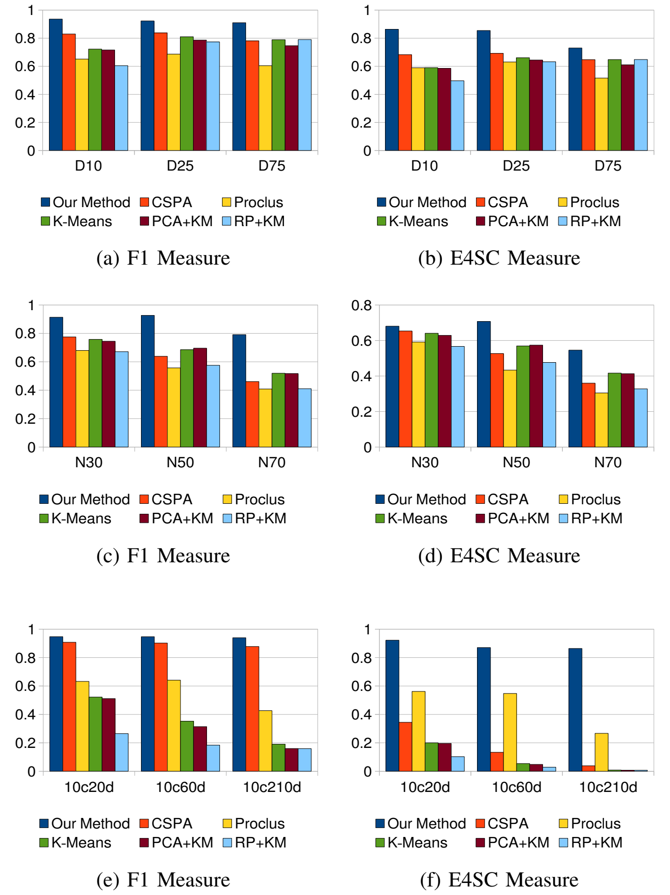
```
Algorithm 2 CartiClus Algorithm
Input: Cartified database $\mathcal{C}$, minimum support $minsup$, minimum length $minlen$, number of requested object sets $n$
Output: Set of subspace clusters $\mathcal{X}$ as pairs of object and subspace sets
$\mathcal{X} \leftarrow \emptyset$  // MFIs
$\Theta \leftarrow$ all objects in $\mathcal{C}$
repeat
    $X \leftarrow \{\mathbf{o}\}$  // randomly select a frequent object $\mathbf{o} \in \Theta$
    $P \leftarrow \{\mathbf{o} \mid supp_{\mathcal{C}}(X \cup \{\mathbf{o}\}) > minsup, \mathbf{o} \in \Theta\}$
    repeat
        $X \leftarrow X \cup \{\mathbf{p}\}$  // randomly select an object $\mathbf{p} \in P$
        $P \leftarrow \{\mathbf{o} \mid supp_{\mathcal{C}}(X \cup \{\mathbf{o}\}) > minsup, \mathbf{o} \in P \setminus X\}$
    until $P = \emptyset$
    if $|X| \geq minlen$ then
        $\mathcal{X} \leftarrow \mathcal{X} \cup \{X\}$
until $|\mathcal{X}| \geq n$ or maximum iterations are reached
$\mathcal{X}' \leftarrow \emptyset$  // output clusters
for each $X \in \mathcal{X}$ do
    for each $X' \in \mathcal{X}'$ do
        if $|X' \cap X| > minlen$ then
            $X' \leftarrow X' \cup X$
            continue with next $X$
    $\mathcal{X}' \leftarrow \mathcal{X}' \cup (X', \{m \mid supp_{\mathcal{C}_m}(X) > 0\})$
return $\mathcal{X}'$
```

V. Experiments

To explore (1) the influence of dimensionality on performance, (2) noise awareness, and (3) relevant subspace detection capabilities, we compare CartiClus against five well-known clustering algorithms: (1) *Proclus* [1] is a projected clustering algorithm, which is one of the best perfoming subspace clustering algorithms according to a recent evaluation study [22]. (2) Traditional *K-Means* [18] on all dimensions (full space), (3) *RP-KM* as K-Means on a set of random projections, and (4) *PCA-KM* as K-Means on a transformed dataspace by Principal Component Analysis (PCA). (5) *CSPA* [26] is an ensemble clustering algorithm using 10 runs of K-Means on separate dimensions as input. To improve the stability of the clusters, we use k-means++ [3] instead of the original K-Means.

We run each setup for 10 times and report the average results. For all competitors, we try to optimize the respective parameter settings by supplying the true number of clusters and dimensions. For our approach, we set $minlen$ parameter approximately to the half of k. Since it is just a minimum, this setting increased the robustness without limiting the ability of finding the whole clusters. The parameter n is set to 100 for all of the experiments. For quality assessment we use the ground truth of synthetic data. We use *F1* measure (harmonic mean of *precision* and *recall*) and *E4SC* (which includes quality assessment of detected subspaces) as used in other publications [20], [21], [10].

We use synthetic data as also used in the evaluation study on subspace clustering [22]. We provide all data sets together with algorithms (implemented in Java) on our project website.[1] All of the experiments are run on a standard PC and each run of each algorithm is completed under 2 minutes.



(a) F1 Measure

(b) E4SC Measure

(c) F1 Measure

(d) E4SC Measure

(e) F1 Measure

(f) E4SC Measure

Figure 2: Increasing the number of (1) overall dimensions, (2) noise objects, and (3) irrelevant dimensions

A. Dimensionality

To assess the subspace cluster detection capabilities of the methods w.r.t. increasing number of dimensions, we conduct experiments on datasets where the number of objects and the average ratio of relevant dimensions per cluster stay constant while the number of dimensions increases. Datasets D10, D25 and D75 have 10, 25 and 75 dimensions, respectively.

The results for these experiments are shown on the first row of Figure 2. The quality of the clusters found by CartiClus is not affected by the number of dimensions. This is expected, because more dimensions only add more views for our method without corrupting the existing views. Therefore, our transformation truly preserves the neighborhood information and its cluster detection capability does not degrade with an increase in the dimensionality.

B. Noise Awareness

Evaluation of noise awareness is done using a set of datasets that contain a fixed number of clusters and dimensions but include different ratios of noise. Datasets N30, N50 and N70 have 30%, 50% and 70% of noise objects, respectively.

[1] http://adrem.uantwerpen.be/cartification

Results on second row of Figure 2 show that CARTICLUS effectively detects high quality clusters, even if they exist in only 30% of the data. Since random objects are not nearest neighbors of most clustered objects in many views of the data, they are not expected to be in many carts, therefore, the support of object sets that they are part of are low. While CARTICLUS detects structure it ignores objects with low supports, i.e., noise. In contrast to this robust behavior, we observe K-Means, Proclus and CSPA to degenerate with increasing noise.

C. Subspace Detection

To evaluate the correct detection of subspaces, we generate datasets with 10 hidden clusters having 2 to 7 relevant dimensions and add additional random dimensions. The datasets 10c20d, 10c60d and 10c210d have 10, 50 and 200 irrelevant dimensions. All of these datasets include additional 5% random objects as noise.

Results on the third row of Figure 2 clearly show the benefits of aggregating information from multiple views of the data. CARTICLUS and CSPA can extract and efficiently use the information from the subspaces that contain structures to overwhelm the negative effects of irrelevant dimensions. Thanks to stricter cluster definition of CARTICLUS, i.e., *co-occurrence* instead of pair similarities, and noise detection capabilities, it can detect better clusters than CSPA. Proclus ignores the irrelevant dimensions up to some level. However, it degenerates with increasing number of irrelevant dimensions. Since feature selection techniques treat all dimensions equally, increasing the number of irrelevant dimensions renders them useless. Note that, since *CSPA* cannot report subspaces for clusters, its *E4SC* score decrease with the number of dimensions.

VI. CONCLUSIONS

The main contribution of this paper is *Cartification*, a novel and highly flexible framework for transforming high dimensional data into the well-studied domain of transaction data, while preserving all local neighborhood information in the respective lower-dimensional projections. Advantages include freedom in the choice of similarity measures. Our experiments show that the cartified data space maintains neighborhood information, allowing a simple itemset mining approach to outperform variety of established dedicated clustering algorithms. Moreover, the literature on mining transaction data is rich in tools to find unexpected/non-redundant relations [19] as well as very unlikely combinations [6], [27], which in turn can be used to find subspace clusters of various interests. In short, our transformation opens a wide area of applications for cartified data, in particular for those tasks that rely on neighborhood information on high dimensional data.

ACKNOWLEDGEMENTS

Emmanuel Müller and Jilles Vreeken are supported by Post-Doctoral Fellowships of the Research Foundation – Flanders (FWO). Further, this work is supported by the Young Investigator Group program of KIT as part of the German Excellence Initiative.

REFERENCES

[1] C. Aggarwal, J. Wolf, P. Yu, C. Procopiuc, and J. Park, "Fast algorithms for projected clustering," in *SIGMOD*, 1999, pp. 61–72.

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *SIGMOD*, 1998, pp. 94–105.

[3] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA*, 2007, pp. 1027–1035.

[4] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?" in *ICDT*. Springer, 1999, pp. 217–235.

[5] C.-H. Cheng, A. W. Fu, and Y. Zhang, "Entropy-based subspace clustering for mining numerical data," in *KDD*, 1999, pp. 84–93.

[6] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *KDD*. ACM, 1999, pp. 43–52.

[7] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *J. Mach. Learn. Res.*, vol. 5, no. 8, pp. 909–921, 2004.

[8] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in *ICML '04*. ACM, 2004, pp. 36–.

[9] A. Fred and A. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE TPAMI*, vol. 27, no. 6, pp. 835–850, 2005.

[10] S. Günnemann, I. Färber, E. Müller, I. Assent, and T. Seidl, "External evaluation measures for subspace clustering," in *CIKM*. ACM, 2011, pp. 1363–1372.

[11] A. Hinneburg, C. C. Aggarwal, and D. A. Keim, "What is the nearest neighbor in high dimensional spaces?" in *VLDB*, 2000, pp. 506–515.

[12] A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM CSUR*, vol. 31, no. 3, pp. 264–323, 1999.

[13] R. A. Jarvis and E. A. Patrick, "Clustering using a similarity measure based on shared near neighbors," *IEEE TC*, vol. 22, no. 11, pp. 1025–1034, 1973.

[14] I. Joliffe, *Principal Component Analysis*. Springer, New York, 1986.

[15] K. Kailing, H.-P. Kriegel, and P. Kröger, "Density-connected subspace clustering for high-dimensional data," in *SDM*, 2004, pp. 246–257.

[16] F. Keller, E. Müller, and K. Böhm, "HiCS: High contrast subspaces for density-based outlier ranking," in *ICDE*, 2012.

[17] J. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer, New York, 2007.

[18] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. Symp. Math. Stat. and Prob*, 1967, pp. 281–297.

[19] M. Mampaey, N. Tatti, and J. Vreeken, "Summarizing data succinctly with the most informative itemsets," *ACM TKDD*, pp. 1–44, 2012.

[20] G. Moise, J. Sander, and M. Ester, "P3C: A robust projected clustering algorithm," in *ICDM*, 2006, pp. 414–425.

[21] E. Müller, I. Assent, S. Günnemann, R. Krieger, and T. Seidl, "Relevant subspace clustering: Mining the most interesting non-redundant concepts in high dimensional data," in *ICDM*, 2009, pp. 377–386.

[22] E. Müller, S. Günnemann, I. Assent, and T. Seidl, "Evaluating clustering in subspace projections of high dimensional data," in *VLDB*, 2009, pp. 1270–1281.

[23] H. V. Nguyen, E. Müller, J. Vreeken, F. Keller, and K. Böhm, "CMI: An information-theoretic contrast measure for enhancing subspace cluster and outlier detection," in *SDM*, 2013, pp. 198–206.

[24] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: a review," *SIGKDD Explor. Newsl.*, vol. 6, no. 1, pp. 90–105, 2004.

[25] V. Roth and T. Lange, "Feature selection in clustering problems," in *NIPS*, 2003, pp. 141–152.

[26] A. Strehl and J. Ghosh, "Cluster ensembles - a knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.

[27] G. I. Webb, "Self-sufficient itemsets: An approach to screening potentially interesting associations between items," *ACM TKDD*, vol. 4, no. 1, pp. 3:1–3:20, Jan. 2010.