Data Min Knowl Disc (2012) 24:387–410
DOI 10.1007/s10618-011-0237-7

Tracing Evolving Subspace Clusters in Temporal Climate Data

Stephan Günnemann · Hardy Kremer · Charlotte Laufkötter · Thomas Seidl

Received: 14 January 2011 / Accepted: 6 September 2011 / Published online: 20 September 2011

Abstract Analysis of temporal climate data is an active research area. Advanced data mining methods designed especially for these temporal data support the domain expert's pursuit to understand phenomena as the climate change, which is crucial for a sustainable world. Important solutions for mining temporal data are cluster tracing approaches, which are used to mine temporal evolutions of clusters. Generally, clusters represent groups of objects with similar values. In a temporal context like tracing, similar values correspond to similar behavior in one snapshot in time. Each cluster can be interpreted as a *behavior type* and cluster tracing corresponds to tracking similar behaviors over time. Existing tracing approaches are for datasets satisfying two specific conditions: The clusters appear in all attributes, i.e., *fullspace clusters*, and the data objects have unique identifiers. These identifiers are used for tracking clusters by measuring the number of objects two clusters have in common, i.e. clusters are traced based on *similar object sets*. These conditions, however, are strict: First, in complex data, clusters are often hidden in individual *subsets of the dimensions*. Second, mapping clusters based on similar objects sets does not reflect the idea of tracing similar

Responsible editor: Katharina Morik, Kanishka Bhaduri and Hillol Kargupta.

S. Günnemann · H. Kremer (✉) · T. Seidl
Data Management and Data Exploration Group, RWTH Aachen University, Aachen, Germany
e-mail: kremer@cs.rwth-aachen.de

S. Günnemann
e-mail: guennemann@cs.rwth-aachen.de

T. Seidl
e-mail: seidl@cs.rwth-aachen.de

C. Laufkötter
Institute of Biogeochemistry and Pollutant Dynamics, ETH Zürich, Zürich, Switzerland
e-mail: charlotte.laufkoetter@env.ethz.ch

behavior types over time, because similar behavior can even be represented by clusters having no objects in common. A tracing method based on *similar object values* is needed. In this paper, we introduce a novel approach that traces *subspace clusters* based on *object value similarity*. Neither subspace tracing nor tracing by object value similarity has been done before.

1 Introduction

A key factor for a sustainable world is to weaken or even stop the ongoing climate change, which can be observed in scientific domains including hydrology (Huntington 2006) and oceanography (Barnett et al. 2001; Brodeur et al. 1999; Hoegh-Guldberg 1999). For understanding the occurring phenomena, a detection and detailed characterization of the occurring changes is crucial. To achieve this, the analysis of temporal climate data, containing climate indices as precipitation, biomass, or ocean temperatures, is a reasonable and reproducible method. For detecting climate change, we need to find patterns that show a long-term evolution over time. Analyzing such temporal properties of patterns is an active research area (Böttcher et al. 2008). In this article, we present a general solution for detecting phenomena in temporal data that is applicable in climate settings.

Clusters are a well known type of pattern: they correspond to similarity-based groupings of the data. A set of clusters, called clustering, is thus a concise description of a dataset. Cluster analysis in comparison to global approaches for data analysis, e.g. Principal Component Analysis, has the benefit that it can detect more local phenomena that only occur in a subset of the data. For analyzing the temporal behavior of clusters, i.e. their evolutions, cluster tracing algorithms have been introduced (Kalnis et al. 2005; Rosswog and Ghose 2008; Spiliopoulou et al. 2006). They find mappings between clusterings of consecutive time steps corresponding to similar clusters. These mappings describe the cluster evolutions over time. Understanding of cluster evolution in climate data can be used in the development of methods to prevent further climate change.

Cluster tracing algorithms go beyond time series clustering (Fu 2011; Liao 2005) by allowing that objects are assigned to different clusters over time. Nevertheless, the existing algorithms for cluster tracing have a severe limitation: Clusters are mapped if the corresponding object sets are similar, i.e., the algorithms check whether the possibly matching clusters have a certain fraction of objects in common; they are unable to map clusters of different objects, even if the objects have similar attribute values. In climate data, however, we are only interested in clusters representing general phenomena and not in clusters of specific individual objects. Therefore, and in contrast to the existing methods, our novel method maps clusters *only* if their corresponding object values are similar, independently of object identities. That is, we trace similar behavior types, which is a fundamentally different concept. This is a relevant scenario, as the following two examples illustrate.

Consider the relationship between the attributes temperature and a specific biomass in the oceans, a relationship which can be represented by clusters. For example that a specific temperature occurs together with a specific amount of biomass. The analyzed data are captured by sensors that are positioned at fixed grid cells in the ocean or the data are the result of complex simulations. It is obvious that the detected clusters representing the relationship evolve and that the observed values are recorded at different grid cells as time progresses. Clusters found by time series clustering approaches or sensor-identity-based tracing cannot express the evolutions we want to analyze.

Another example is scientific data of the earth's surface with the attributes temperature and smoke degree. The latter correlates with forest fire probability. The attribute values are recorded over several months. In this dataset, at some point in time a high smoke degree and high temperatures occur in the northern hemisphere; sixth months later the same phenomenon occurs in the southern hemisphere, as the seasons on the hemispheres are shifted half-yearly to each other.

A cluster tracing algorithm should detect the presented phenomena. However, existing methods do not, since the observed populations, i.e., the sensors and the environment respectively, stay at the same place, and thus there are no shared objects between clusters—only the behavior migrates.

With today's complex data, patterns are often hidden in different subsets of the dimensions; for detecting these clusters with locally relevant dimensions, subspace clustering was introduced (Kriegel et al. 2009; Parsons et al. 2004). However, despite that many temporal data sets are of this kind, e.g., gridded scientific data, *subspace clustering has never been used in a cluster tracing scenario*. The existing cluster tracing methods can only cope with fullspace clusters, and thus cannot exploit the information mined by subspace clustering algorithms. Our novel tracing method measures the subspace similarity of clusters and thus handles subspace clusters by design.

Summarized, we introduce a method for tracing behavior types in temporal data; the types are represented by clusters. The decision, which clusters of consecutive time steps are mapped is based on a novel distance function that tackles the challenges of object value similarity and subspace similarity. Our approach can handle the following developments: emerging or disappearing behavior as well as distinct behaviors that converge into uniform behavior and uniform behavior that diverges into distinct behaviors. By using subspaces, we enable the following evolutions: Behavior can gain or lose characteristics; i.e., the representing subspace clusters can gain or lose dimensions over time, and clusters that have different relevant dimensions can be similar. Varying behavior can be detected; that is, to some extent the values of the representing clusters can change.

Figure 1 exemplifies the evolution of temperature and biomass measurements for three consecutive time steps. The upper part of the figure shows the objects; the lower part abstracts from the objects and illustrates possible clusterings of the databases and tracings between the corresponding clusters. The three time steps do not share objects, i.e., each time step corresponds to a different database; to illustrate this, we used varying object symbols. A cell-based clustering paradigm is assumed, i.e., clusters are defined by lower and upper bounds in each of the two dimensions, and in case of subspace clusters, bounds are only given for the relevant dimensions. For example, the cluster $C_{1,2}$ at time step $t = 1$ is a fullspace cluster in which the temperature and biomass measurements of the cluster's objects are constrained to specific intervals;

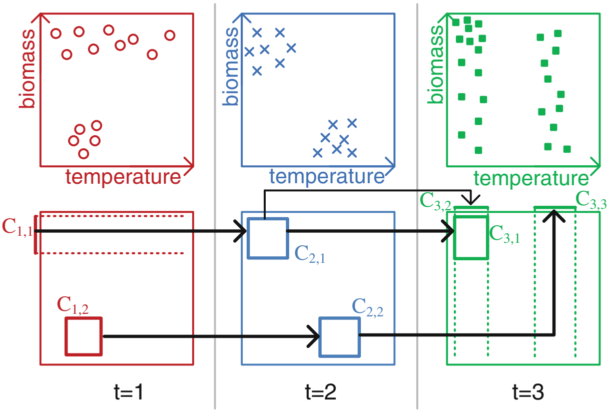


Fig. 1 Top: three consecutive time steps, with each time step corresponding to a different database. (We used varying object symbols to illustrate that the time steps do not share objects.) Bottom: possible clusterings w.r.t. to a cell-based clustering paradigm (lower and upper bounds in relevant dimensions) and exemplary cluster tracings

for the subspace cluster $C_{1,1}$, only the temperature values are constrained to an interval, while the biomass measures are scattered over the whole dimensional extent. An example for behavior that gains characteristics is the mapping of cluster $C_{1,1}$ to $C_{2,1}$: at time step $t = 1$ only the biomass measurements are positioned in a specific interval, while at $t = 2$ also the temperature measurements can be constrained by an interval, i.e., this cluster gains one dimension. Varying behavior is illustrated by the mapping from $C_{1,2}$ to $C_{2,2}$; the values of the cluster have changed. If the two dimensions of the databases were spatial, this could be interpreted as a movement. A behavior divergence can be seen from time step $t = 2$ to $t = 3$: the single cluster $C_{2,1}$ is mapped to the two clusters $C_{3,1}$ and $C_{3,2}$.

Summarized, our contributions are:

- We introduce a novel tracing approach for evolving subspace clusters in high dimensional temporal data, which often occurs in the climate domain. Subspace clusters correspond to behavior types and are traced based on similar object values and not based on object sets.
- We explicitly distinguish several kinds of behavior development for our tracing approach, i.e., clusters can emerge, disappear, diverge, or converge.
- We measure the degree of evolution between two subspace clusters with our novel distance function based on subspace similarity and value similarity.
- We propose a method for information transfer between time steps, to avoid unstable clusterings and to achieve higher quality clusterings. Tracing effectiveness is therefore improved, as it depends on the input clusterings.

This article is structured as follows: Section 2 discusses the related work. Section 3 introduces our new model for tracing subspace clusters. The effectiveness is shown in Sects. 4 and 5 concludes the article.

2 Related work

Several temporal aspects of data are regarded in the literature (Böttcher et al. 2008). In stream clustering scenarios, clusters are adapted to reflect changes in the observed data, i.e., the distribution of incoming objects changes (Aggarwal et al. 2003; Cao et al. 2006). A special case of stream clustering is for moving objects (Jensen et al. 2007; Li et al. 2004), focusing on spatial attributes. Stream clustering in general, however, gives no information about the actual cluster evolution over time (Böttcher et al. 2008). For this, cluster tracing algorithms were introduced (Kalnis et al. 2005; Kremer et al. 2010; Rosswog and Ghose 2008; Spiliopoulou et al. 2006); they rely on mapping clusters of consecutive time steps. As already mentioned, these methods map clusters if the corresponding object sets are similar, i.e. they are based on shared objects. We, in contrast, map clusters only if their corresponding object values are similar, independently of shared objects.

Time series clustering (Boriah et al. 2008; Fu 2011; Liao 2005; Steinbach et al. 2003) or trajectory clustering (Gaffney and Smyth 1999; Vlachos et al. 2002) can be seen as even more limited variants of similar-object-set-based cluster tracing, since the obtained clusters have constant object sets that do not change over time. Accordingly, these methods search for groups of objects that have a similar behavior over the whole time extent. There is no possibility of detecting that a behavior reflected by one time series cluster is occurring in a different cluster after some time.

Approaches for analyzing climate patterns in multivariate data such as (Hoffman et al. 2005) do not actually track the clusters; they cluster the time series of all points in time altogether, neglecting the temporal information. Afterwards, the resulting clusters are remapped to their temporal extension. By this, however, highly evolving patterns get lost, because they are not considered as one cluster.

The work in Aggarwal (2005) analyzes multidimensional temporal data based on dense regions that can be interpreted as clusters. The approach is designed to detect substantial changes of dense regions; however, tracing of evolving clusters that slightly change their position or subspace is not possible.

The research area of comparing clusterings, e.g., for evaluating how good an obtained clustering reflects a given ground truth clustering (Kremer et al. 2011; Zhou et al. 2005), is related to cluster tracing since clusters of two given clusterings need to be mapped. These approaches are designed for another purpose and thus there is no consideration of several time steps and more importantly, no consideration of evolutions as emerging or disappearing clusters.

A further limitation of existing cluster tracing algorithms is that they can only cope with full space clusters. Full-space clustering models use all dimensions in the data space (Ester et al. 1996). Due to the effects of high dimensionality (Hinneburg et al. 2000), i.e., irrelevant dimensions obfuscate the clustering structure and distances between objects grow alike, these approaches do not find meaningful clusters. Global dimensionality reduction approaches like PCA tend to mitigate these effects, but locally relevant clusters are missed. Therefore, for finding clusters with locally relevant dimensions, subspace clustering was introduced (Agrawal et al. 1998). An overview of different subspace clustering approaches can be found in Kriegel et al. (2009) and Parsons et al. (2004). In Müller et al. (2009), the differences between recent subspace clustering approaches are analyzed and thoroughly evaluated. A conclusion of this evaluation is that cell-based approaches (Procopiuc et al. 2002; Yiu and Mamoulis 2003) have shown to be very efficient and subspace clusterings of high quality are generated. Until now, subspace clusters were only applied in streaming scenarios (Aggarwal et al. 2004), but never in cluster tracing scenarios; deciding whether subspace clusters of varying dimensionalities are similar is a challenging issue. Our algorithm is designed for this purpose.

3 A novel tracing model

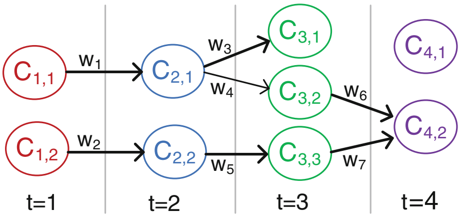
Our main objective is to trace behavior types and their developments over time. This is formalized in the following section. First, some basic notations: For each time step $t \in \{1, \ldots, T\}$ of our temporal data we have a D-dimensional database $DB_t \subseteq \mathbb{R}^D$. We assume the data to be normalized between [0, 1]. A subspace cluster $C_{t,i} = (O_{t,i}, S_{t,i})$ at time step t is a set of objects $O_{t,i} \subseteq DB_t$ along with a set of relevant dimensions $S_{t,i} \subseteq \{1, \ldots, D\}$. The objects are similar within these relevant dimensions. The set of all subspace clusters $\{C_{t,1}, \ldots, C_{t,k}\}$ of the same time step t is denoted as subspace clustering $Clus_t$. Our approach is independent of a specific clustering model. For illustration, we assume a cell-based clustering paradigm in the following, i.e. clusters are described by intervals (lower and upper bounds) in the relevant dimensions.

Each subspace cluster of our clusterings represents a behavior type. In our cell-based clustering paradigm, this could for example be a set of sensor readings expressing the relationship between biomass and temperature: the temperatures and biomass measured by many sensors are contained in a specific interval. The terms behavior, behavior type, and cluster are used interchangeably. In Sect. 3.1 we introduce how to trace a behavior over time and which temporal developments are possible. For tracing it is necessary to measure the similarity between behavior types; the formalization of this step is presented in Sect. 3.2. In Sect. 3.3 the clustering process in the single time steps is improved by incorporating temporal information, avoiding unstable clusterings and achieving higher quality clusterings. We conclude in Sect. 3.4 with a complexity analysis of our method.

3.1 Tracing of behavior types

In this section, we are interested in whether a typical behavior in time step t continues in $t + 1$. In ocean data, for example, a cluster detected in a set of sensor readings at one time step can be rediscovered (i.e., having similar values) in the next time step in a different set of sensor readings, and these sensor readings can be obtained by a different set of sensors. Other kinds of temporal developments are the disappearance of a behavior or a split-up into different behaviors. We have to identify these temporal developments for reasonable tracing of behaviors. Formally, we need a *mapping function* that maps each cluster at a given time step to a set of clusters in the next time step; we denote these successors as temporal continuations. Two clusters $C_{t,i}$ and $C_{t+1,j}$ are mapped if they are identified as similar behaviors. We use a distance function for

Fig. 2 Example for a mapping graph with edge weights. The first 3 time steps correspond to the clusters and mappings illustrated in Fig. 1



clusters to measure these similarities. If the distance is small enough the mapping is performed.

Definition 1 Mapping function. Given a distance function *dist* for two subspace clusters, the mapping function $M_t : Clus_t \rightarrow \mathcal{P}(Clus_{t+1})$ that maps a cluster to its temporal continuations is defined by

$$M_t(C_{t,i}) = \{C_{t+1,j} \mid dist(C_{t,i}, C_{t+1,j}) < \tau\}$$

A cluster can be mapped to zero, one, or several clusters (1:n). We can map several clusters to the same cluster (m:1). We do not enforce (1:1) mappings. These properties are needed, so that disappearance or convergence of behaviors can be detected. We describe all pairs of mapped clusters between two consecutive time steps by a binary relation:

$$R_t = \{(C_{t,i}, C_{t+1,j}) \mid C_{t+1,j} \in M_t(C_{t,i})\} \subseteq Clus_t \times Clus_{t+1}$$

Each tuple corresponds to one cluster mapping, i.e., for a behavior type in t we have identified a similar one in the next time step $t + 1$. These mappings as well as the corresponding clusters can be represented by a *mapping graph*. Reconsider that it is possible to map a behavior to several behaviors in the next time step (cf. Fig. 1, $t = 2 \rightarrow t = 3$). All these behaviors, however, are not equally similar to the original behavior. We represent this by using edge weights within the mapping graph; the weights indicate the strength of the temporal continuation. We measure similarity based on distances, and therefore small weights denote a strong continuation while high weights reflect a weaker continuation.

Definition 2 Mapping graph. A mapping graph $G = (V, E, w)$ is a directed and weighted graph with the following properties:

- Nodes represent clusters, i.e., $V = \bigcup_{i=1}^{T} Clus_t$
- Edges represent cluster mappings, i.e., $E = \bigcup_{i=1}^{T-1} R_t$
- Edge weights indicate the strength of the temporal continuations, i.e., $\forall (C_i, C_j) \in E : \ w(C_i, C_j) = dist(C_i, C_j)$

In Fig. 2 an exemplary mapping graph with edge weights is illustrated. A mapping graph allows us to categorize different kinds of temporal developments.

Definition 3 Kinds of temporal developments. Given a mapping graph $G = (V, E, w)$, the behaviors represented by clusters $C \in V$ can be categorized:

- a behavior *disappears*, if $outdegree(C) = 0$
- a behavior *emerges*, if $indegree(C) = 0$
- a behavior *diverges*, if $outdegree(C) > 1$
- different behaviors *converge* to a single behavior, if $indegree(C) > 1$

In Fig. 2 for example, the cluster $C_{3,1}$ corresponds to a disappearing behavior and the cluster $C_{4,1}$ is an emerging one. While the cluster $C_{2,1}$ diverges to several different behaviors, the cluster $C_{4,2}$ results from converging behaviors.

The kinds of temporal developments show whether a behavior appears in similar ways in the subsequent time step. However, it is also important to trace a behavior over several time steps. It should be noted that the characteristics of a behavior can naturally change over this time period. Thus, we denote the tracing of a single behavior over a specific period as an *evolving cluster*. Formally, an evolving cluster is described by a single path through the mapping graph. Based on the mapping graph in Fig. 2 we are able to trace the evolving cluster $C_{1,1} \rightarrow C_{2,1} \rightarrow C_{3,2} \rightarrow C_{4,2}$.

To ensure that the evolving clusters are correctly identified, we have to account for several evolution criteria to be included in our distance function. These criteria are presented in the following section.

3.2 Cluster distance measure

Our objective is to identify similar behaviors. Technically, a distance measure is needed to formally determine the similarity of two given clusters. Keep in mind, that measuring the similarity based on the fraction of shared objects is not meaningful in our approach. Even totally different populations can show up with a similar behavior in consecutive time steps.

We have to distinguish two kinds of evolution: First, a cluster can gain or lose characteristics, i.e., the relevant dimensions of a subspace cluster can evolve. Second, within the relevant dimensions the values can change over time. Both aspects have to be considered by our distance function for effective similarity measurement of evolving clusters.

Similarity based on subspaces. Each cluster represents a behavior type, and because we are considering subspace clusters, the characteristics of a behavior are restricted to a subset of the dimensions. If a behavior remains stable over time, its subspace remains also unchanged. The relevant dimensions of the underlying clusters are identical. Let us consider the clusters $C_{t,i} = (O_{t,i}, S_{t,i})$ and $C_{t+1,j} = (O_{t+1,j}, S_{t+1,j})$ of the time steps t and $t+1$. The represented behaviors are very similar if the dimensions $S_{t,i}$ are also included in $S_{t+1,j}$.

On the other hand, it is possible that a behavior loses some of its characteristics over time. In Fig. 1, for example, the attribute biomass is no longer relevant in time step $t = 3$ for the behavior depicted on the bottom. Accordingly, a distance measure is meaningful if behavior types are rated as similar, even if they lose some relevant dimensions. That is, the smaller the term $1 - \frac{|S_{t,i} \cap S_{t+1,j}|}{|S_{t,i}|}$, the more similar are the clusters.

This formula alone, however, would prevent an information gain: If a cluster $C_{t,i}$ evolves to $C_{t+1,j}$ by spanning more relevant dimensions, this would not be assessed positively. We would get the same distance for a cluster with the same shared dimensions like $C_{t,i}$, but without additional relevant dimensions like $C_{t+1,j}$. Since more dimensions mean more information, we do consider this. Consequently, the smaller the term $1 - \frac{|S_{t+1,j} \backslash S_{t,i}|}{|S_{t+1,j}|}$, the more new information is obtained.

Usually it is more important for tracing that we retain relevant dimensions. Few shared dimensions and many new ones normally do not indicate similar behavior. Thus, we need a trade-off between retained dimensions and new (gained) dimensions. This is achieved by a linear combination of the two introduced terms:

Definition 4 Distance w.r.t. subspaces. The similarity w.r.t. to subspaces between two clusters $C_{t,i} = (O_{t,i}, S_{t,i})$ and $C_{t+1,j} = (O_{t+1,j}, S_{t+1,j})$ is defined by

$$S(C_{t,i}, C_{t+1,j}) = \alpha \cdot \left(1 - \frac{|S_{t,i} \cap S_{t+1,j}|}{|S_{t,i}|}\right) + (1-\alpha) \cdot \left(1 - \frac{|S_{t+1,j} \backslash S_{t,i}|}{|S_{t+1,j}|}\right)$$

with trade-off factor $\alpha \in [0, 1]$. In this definition, only the sets of relevant dimensions $S_{t,i}$ are compared, ignoring the object sets $O_{t,i}$.

By choosing $\alpha \gg 1 - \alpha$ we achieve that the similarity between two behaviors is primarily rated based on their shared dimensions. In the best case $C_{t+1,j}$ retains all dimensions and covers many additional ones (distance of 0). In the worst case we have nearly no shared dimensions and no additional ones (distance of 1).

Similarity based on statistical characteristics. Besides the subspace similarity, the actual values within these dimensions are important. For example, solely because two clusters share a dimension like 'temperature', their values can differ extremely (high vs. low temperature); these behaviors should not be mapped. A small change in the values, however, is possible for evolving behaviors. Considering a spatial dimension, this change corresponds to a slight cluster movement.

Given a cluster $C = (O, S)$, we denote the set of values in dimension d with $v(C, d) = \{o[d] \mid o \in O\}$. The similarity between two clusters $C_{t,i} = (O_{t,i}, S_{t,i})$ and $C_{t+1,j} = (O_{t+1,j}, S_{t+1,j})$ is thus achieved by analyzing the corresponding sets $v(C_{t,i}, d)$ and $v(C_{t+1,j}, d)$. In many applications, normal distributions are well suited to model the values a cluster follows; this is often exploited, for example by clustering based on maximizing the data's likelihood assuming a mixture of normal distributions (Dempster et al. 1977). Thus, without losing much information, we can represent the sets $v(C_{t,i}, d)$ and $v(C_{t+1,j}, d)$ by two normal distributions X_d and Y_d with means μ_x, μ_y and variances σ_x, σ_y. The similarity can now be measured by comparing these distributions. We use the information theoretic Kullback-Leibler divergence (KL). Informally, we calculate the expected number of bits required to encode a new distribution of values at time step $t + 1$ (Y_d) given the original distribution of the values at time step t (X_d). By using our cluster approximations $\mu_x, \mu_y, \sigma_x, \sigma_y$, we can calculate the KL via a closed-form expression:

$$\mathrm{KL}(Y_d \| X_d) = ln(\frac{\sigma_x}{\sigma_y}) + \frac{\sigma_y^2 + (\mu_y - \mu_x)^2}{2\sigma_x^2} - \frac{1}{2} =: \mathrm{KL}(C_{t,i}, C_{t+1,j}, d)$$

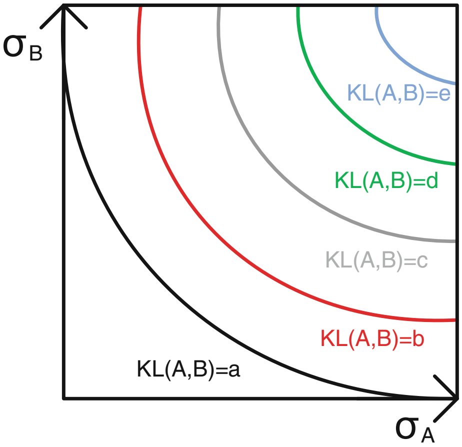


Fig. 3 Exemplary contour lines for the KL distance between normal distributions A, B with constant $\mu_A - \mu_B$; the corresponding variances are plotted on the x- and y-axis; it holds $a > b > c > d > e$

The KL is not a symmetric measure; therefore it is suitable for a temporal tracing model, in which time progresses in one direction. By using the KL, we do not just account for the absolute deviation of the means, but we have also the advantage of including the variances. A behavior with a high variance in a single dimension allows a higher evolution of the means for successive similar behaviors. A small variance of the values, however, only permits a smaller deviation of the means. This is illustrated in Fig. 3.

We use the KL to calculate the similarity per dimension, and the overall similarity is attained by cumulating over several ones. Apparently, we just have to use dimensions that are in the intersection of both clusters. The remaining dimensions are non-relevant for at least one cluster and hence are already penalized by our subspace distance function. Our first approach for computing the similarity based on statistical characteristics is

$$V(C_{t,i}, C_{t+1,j}, I) = \frac{\sum_{d \in I} \mathrm{KL}(C_{t,i}, C_{t+1,j}, d)}{|I|} \tag{1}$$

with $I = S_{t,i} \cap S_{t+1,j}$ for averaging.

In a perfect scenario this distance is a good way to trace behaviors. In practice, however, we face the following problem: Consider the example in Fig. 4 (note the 7-dim. space). With our clustering we identify the cluster $C_{1,2}$ at time step $t = 1$ and the cluster $C_{2,2}$ with the same relevant dimensions in $t = 2$. However, $C_{2,2}$ is shifted in dimensions d_1 and d_2; the distance function proposed above (Eq. 1) would determine a very high value and hence the behaviors would not be mapped. A large part $\{d_3, ..., d_7\}$ of the shared relevant dimensions $\{d_1, ..., d_7\}$, however, show nearly the same characteristics in both clusters. The *core of the behaviors* is completely identical, and therefore a mapping is reasonable; this is illustrated by the mapping of $C_{1,2}$ to $C_{2,2}$ in the lower part of Fig. 4. Consider another example: The core of two clusters detected in the oceans is identical, e.g., their biomass and temperature are similar. However, the clusters are located on different hemisphere so that their ocean currents

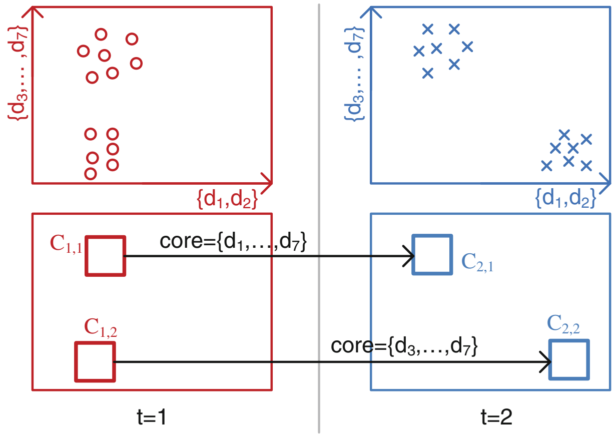


Fig. 4 Example for the concept of core dimensions in a 7-dimensional space

are different. These additional, *non-core*, dimensions provide us with further informations about the single clusters at their current time step. They are mainly induced by the individual populations and are technically resulted by the method of subspace clustering. For the continuation of the behavior, however, these dimensions are not important. Note that non-core dimensions are a different concept than non-relevant ones; non-core dimensions are shared relevant ones with differing values.

An effective distance function between clusters has to identify the core of the behaviors and incorporate it into the distance. We accomplish this by using a subset $Core \subseteq S_{t,i} \cap S_{t+1,j}$ for comparing the values in Eq. 1 instead of the whole intersection $S_{t,i} \cap S_{t+1,j}$. Unfortunately, this subset is not known in advance, and it is not reasonable to choose a fixed threshold that excludes some dimensions from the distance calculation if the corresponding dissimilarity is too large. Thus, we develop a variant that automatically determines the core. The idea is to choose the 'best' core among all possible cores for the given two clusters. That is, for each possible core we determine the distance w.r.t. their value distributions, and we additionally penalize dimensions that are not included in the core. The core with the smallest overall distance is selected, i.e. we trade off the size of the core against the value $V(C_{t,i}, C_{t+1,j}, Core)$:

Definition 5 Core-based distance function for values. The core-based distance function w.r.t. values for two clusters $C_{t,i} = (O_{t,i}, S_{t,i})$ and $C_{t+1,j} = (O_{t+1,j}, S_{t+1,j})$ is defined by

$$V(C_{t,i}, C_{t+1,j}) = \min_{\substack{Core \subseteq S_{t,i} \cap S_{t+1,j} \\ \wedge |Core|>0}} \left\{ \beta \cdot \frac{|NonCore|}{|S_{t,i} \cap S_{t+1,j}|} + (1-\beta) \cdot V(C_{t,i}, C_{t+1,j}, Core) \right\}$$

with the penalty factor $\beta \in [0, 1]$ for the non-core dimensions $NonCore = (S_{t,i} \cap S_{t+1,j}) \backslash Core$.

By selecting a smaller core, the first part of the distance formula gets larger. The second part, however, gets the possibility of determining a smaller value. The core must comprise at least one dimension; otherwise, we could map two clusters even if they have no dimensions with similar characteristics.

Overall distance function. To correctly identify the evolving clusters in our temporal data we have to consider evolutions in the relevant dimensions as well as in the value distributions. Thus, we have to use both distance measures simultaneously. Again, we require that two potentially mapped clusters share at least one dimension; otherwise, these clusters cannot represent similar behaviors.

Definition 6 Overall distance function. The overall distance function, comprising subspace and core-based value similarity, for two clusters $C_{t,i} = (O_{t,i}, S_{t,i})$ and $C_{t+1,j} = (O_{t+1,j}, S_{t+1,j})$ with $|S_{t,i} \cap S_{t+1,j}| > 0$ is defined by

$$dist(C_{t,i}, C_{t+1,j}) = \gamma \cdot V(C_{t,i}, C_{t+1,j}) + (1-\gamma) \cdot S(C_{t,i}, C_{t+1,j})$$

with $\gamma \in [0, 1]$. In the case of $|S_{t,i} \cap S_{t+1,j}| = 0$, the distance is set to ∞.

3.3 Clustering for improved tracing

In the previous sections we assume a given clustering per time step such that we can determine the distances and the mapping graph. In general, our tracing model is independent of the underlying clustering method; it can flexible be chosen based on the current application. However, since there are temporal relations between consecutive time steps, we develop a clustering method whose accuracy is improved by these relations and that avoids unstable clusterings (i.e., totally different clusterings in consecutive time steps). Our subspace clustering definition adapts the cell-based clustering paradigm (Procopiuc et al. 2002; Yiu and Mamoulis 2003), because approaches from this paradigm show high quality results and are efficiently computable (Müller et al. 2009). A benefit of cell-based algorithms is that the representing cells can be positioned flexible in space, i.e. there is no fixed grid as in other approaches. Basically, we approximate clusters by hypercubes in the data space. The extent of a hypercube is restricted to w in the relevant dimensions of a cluster and unrestricted in the non-relevant ones. Thus, the values of the objects vary to at most w within the relevant dimensions and hence we have identified a meaningful grouping. Additionally, we require that a valid cluster summarizes at least $minSup$ many objects.

Definition 7 Hypercube and valid subspace cluster. A hypercube H_S with the relevant dimensions S is defined by lower and upper bounds

$$H_S = [low_1, up_1] \times [low_2, up_2] \times \ldots \times [low_D, up_D]$$

with $up_i - low_i \leq w \; \forall i \in S$ and $low_i = -\infty, up_i = \infty \; \forall i \notin S$. The mean of H_S is called m_{H_S}. The hypercube H_S represents all objects Obj(H_S) $\subseteq DB$ with $o \in Obj(H_S) \Leftrightarrow \forall d \in \{1, \ldots, D\} : low_d \leq o[d] \leq up_d$. A subspace cluster $C = (O, S)$ is valid iff there exists a hypercube H_S with $Obj(H_S) = O$ and $|Obj(H_S)| \geq minSup$.

In the next paragraphs we introduce how temporal relations between time steps can be exploited to improve the tracing accuracy and to avoid unstable clusterings.

Predecessor information. We assume that the initial clustering at time step $t = 1$ is known. (We discuss this later.) Caused by the temporal aspect of our data, clusters at a time step t occur with high probability also in the next time step—not identical, but similar. Consequently, given a cluster and the corresponding hypercube H_S at time step t, we try to find a cluster at the subsequent time step in a *similar region*. This is achieved by a Monte Carlo approach, i.e., we draw a random point $m_{t+1} \in \mathbb{R}^D$ that represents the initiator of a new hypercube and that is located nearly to the mean m_{H_S} of H_S.

Definition 8 Initiator of a hypercube. A point $p \in \mathbb{R}^D$, called initiator, together with a width w and a subspace S induces a hypercube $H_S^w(p)$ that is defined by $\forall d \in S : low_d = p[d] - \frac{w}{2}, up_d = p[d] + \frac{w}{2}$ and $\forall i \notin S : low_i = -\infty, up_i = \infty$.

Accordingly, by using this initiator concept, we apply the cell-based clustering paradigm. After inducing a hypercube by an initiator, we check if the corresponding cluster is valid. Formally, the initiator m_{t+1} is drawn from the region $H_S^{2w}(m_{H_S})$; that is, we permit a change of the cluster. The new hypercube is then induced by m_{t+1}, i.e., the generated cluster is $H_S^w(m_{t+1})$.

With this method we detect changes in the values; however, also the relevant dimensions of a cluster can change: The initiator m_{t+1} can induce different hypercubes for different relevant dimensions S. For example, all or just one dimension of the hypercube could be restricted to the maximal extent w. Therefore, beside the initiator m_{t+1}, we additionally have to determine the relevant subspace of the new cluster. We discuss both issues in the following.

Determining the best cluster. A first possible approach is to use a quality function (Procopiuc et al. 2002; Yiu and Mamoulis 2003; Günnemann et al. 2010): $\mu(H_S) = Obj(H_S) \cdot k^{|S|}$. The more objects or the more relevant dimensions are covered by the cluster, the higher is its quality. These objectives are contrary; therefore a trade-off is realized with the constant parameter k. In time step $t + 1$ we could simply choose the subspace S such that the hypercube $H_S^w(m_{t+1})$ maximizes $\mu(H_S^w(m_{t+1}))$.

This method, however, optimizes the quality of each single cluster; it is not intended to find good tracings. Possibly, the distance between each cluster from the previous clustering $Clus_t$ and our new cluster is large, and we would find no similar behaviors. Thus, we directly integrate the distance function $dist$ into the quality function, i.e. we want to prefer clusters leading also to small mapping distances. Consequently, we choose the subspace S such that the hypercube $H_S^w(m_{t+1})$ maximizes our novel distance based quality function.

Definition 9 Distance based quality function. Given the hypercube H_S in subspace S and a clustering $Clus_t$, the distance based quality function is

$$q(H_S) = \mu(H_S) \cdot (1 - \min_{C_t \in Clus_t} \{dist(C_t, C_S)\})$$

where C_S indicates the induced subspace cluster of the hypercube H_S.

We enhance the quality of the clustering by selecting a set of possible initiators M from the specified region; this is also important as the direction of a cluster change is not known in advance. From the resulting set of potential clusters, we select the one that has the highest quality.

Overall we realize that for each cluster $C \in Clus_t$ a potential temporal continuation is identified in time step $t+1$. Nonetheless it is also possible that our method identifies no valid hypercube for a single cluster $C \in Clus_t$, e.g., because too few objects are located around the selected initiator. This indicates that a behavior type has disappeared in the current time step.

```
method: $main$(databases $DB_1, \dots, DB_T$)
$G = (V, E, w)$                                      // mapping graph, empty at beginning
$Clus_0 = \emptyset, \dots, Clus_T = \emptyset$                  // no clusters determined
for $t = 1, \dots, T$ do
    $Remain = DB_t$                                   // all objects unclustered
    for $C = (O, S) \in Clus_{t-1}$ do                 // predecessor information
        determine hypercube $H_S$ of $C$ and its mean $m_{H_S}$
        randomly draw a set $M$ of initiators with $m \in M \Rightarrow m \in H_S^{2w}(m_{H_S})$
        $C^* \leftarrow ClusterAndMappingDetection(M, t)$
        if $C^* \neq \perp$ then                          // let $C^* = \big(O^*, S^*\big)$
            $Clus_t = Clus_t \cup \{C^*\}$
            $Remain = Remain \backslash O^*$
    while $Remain \neq \emptyset$ do                     // still unclustered objects
        randomly draw initiators $M \subseteq Remain$
        $C^* \leftarrow ClusterAndMappingDetection(M, t)$
        if $C^* \neq \perp$ then                      // let $C^* = \big(O^*, S^*\big)$
            $Clus_t = Clus_t \cup \{C^*\}$
            $Remain = Remain \backslash O^*$
        else                                   // all clusters detected, next time step
            break;
return $G$;

method: $ClusterAndMappingDetection$(initiators $M$, time step $t$)
// —— calculate best cluster ——
$(m^*, S^*) = \arg \max_{\substack{(m,S) \in \\ M \times \mathcal{P}(\{1,\dots,D\})}} \begin{cases} q(H_S^w(m)) & \text{if } |Obj(H_S^w(m))| \geq minSup \\ -1 & \text{else} \end{cases}$
if $|Obj(H_{S^*}^w(m^*))| < minSup$ then return $\perp$;        // only invalid clusters
$C^* = (Obj(H_{S^*}^w(m^*)), S^*)$                           // the novel cluster
// —— update mapping graph ——
$V = V \cup \{C^*\}$                                 // new node in mapping graph
for $C_{t-1} \in Clus_{t-1}$ do
    if $dist(C_{t-1}, C^*) < \tau$ then
        $E = E \cup \{(C_{t-1}, C^*)\}$                 // new edge in mapping graph
return $C^*$;
```

Algorithm 1: Processing scheme of the subspace cluster tracing method

Uncovered objects and the initial clustering. If behavior disappears or emerges, there will be some objects of the current time step that are not part of any identified cluster. In other words: if we denote the set of clusters generated so far by $Clus_{t+1}$, the set $Remain_{t+1} := DB_{t+1} \setminus \bigcup_{\substack{C_i=(O_i,S_i) \\ C_i \in Clus_{t+1}}} O_i$ can still contain objects and therefore clusters. Especially for the initial clustering at time step $t = 1$ we have no predecessor information and hence $Clus_1 = \emptyset$ at the start. To discover as many patterns as possible, we have to check if the objects within $Remain_{t+1}$ induce novel clusters. We cannot infer the initiators of possible hypercubes based on previous clusters; instead, we use the remaining objects itself as initiators for the hypercubes. We draw a set of initiators $M \subseteq Remain_{t+1}$, where each $m \in M$ induces a set of hypercubes $H_S^w(m)$ in different subspaces. Finally, we choose the hypercube that maximizes our quality function. If this hypercube corresponds to a valid cluster, we add it to $Clus_{t+1}$, and thus the set $Remain_{t+1}$ is reduced. This procedure is repeated until no valid cluster is identified or the set $Remain_{t+1}$ is empty. Note that our method has the advantage of generating overlapping clusters. We select the initiators from the set $Remain_{t+1}$; the objects covered by the hypercubes, however, are a subset of the whole database. Thereby we realize a meaningful non-partitioning clustering.

Summary. The overall processing scheme of our algorithm is illustrated in Algorithm 1. For each point in time (line 3) we first perform clustering based on the given predecessor information (lines 5–11), followed by our method to detect emerging clusters (lines 12–19). The actual clusters and corresponding mappings between clusters are detected in lines 21–30, using our distance based quality function. Overall, our clustering method specifically utilizes the advantages of temporal data to obtain high quality temporal continuations by nesting mapping and clustering: We steer the cluster identification to regions in the data space where good clusters are expected. Thus, cluster tracing is no longer independent of the clustering but we integrate model specific properties in this step.

3.4 Computational aspects

In the following we briefly analyze the computational complexity of our model. Essentially, we can distinguish two phases within the method: the actual clustering of each time step and the calculation of mapping distances. Since in general our model is independent of the underlying clustering algorithm and thus any choice would be possible, we focus on the second aspect. Anyhow we want to mention the exponential complexity of many subspace clustering algorithms w.r.t. the dimensionality of clusters hidden in the data. Thus, this clustering step is usually the determining factor for the run times in practical applications of the method.

For determining a mapping graph consisting of T time steps, we have to calculate the mapping distances between $T - 1$ many pairs of clusterings. For two successive clusterings $Clus_t$ and $Clus_{t+1}$ we have to determine $|Clus_t| \cdot |Clus_{t+1}|$ distance values according to Def. 6 in which the value based distance function is the more complex summand, because the optimal core is identified by a minimization procedure (cf. Def. 5). If the intersection between the relevant dimensions of two clusters has a cardinality of i, $2^i - 1$ potential cores have to be checked. Let k and l be the average dimensionality for the clusters at time step t and $t+1$ respectively. By assuming that the relevant dimensions are randomly drawn from all dimensions $\{1, \ldots, D\}$, the number of cluster pairs whose intersection has a cardinality of i can be determined by binomial coefficients. By averaging over all possible pairs, we get the expected number of cores to be tested given two clusters with dimensionality k and l:

$$\begin{aligned}
&\frac{\sum_{i=0}^{l}(2^i - 1) \cdot \binom{k}{i} \cdot \binom{D-k}{l-i}}{\sum_{i=0}^{l} \binom{k}{i} \cdot \binom{D-k}{l-i}} \\
&\leq \frac{\sum_{i=0}^{\min\{k,l\}} 2^i \cdot \binom{k}{i} \cdot \binom{D-k}{l-i}}{\binom{D}{l}} \\
&\leq \sum_{i=0}^{\min\{k,l\}} 2^i \cdot \frac{k^i \cdot (D-k)^{l-i} \cdot l^l}{i^i \cdot (l-i)^{l-i} \cdot D^l} \in O(2^{\min\{k,l\}})
\end{aligned}$$

Summarized, the overall number of cores to be tested and thus the overall number of value based similarity calculations for all time steps is given by

$$\begin{aligned}
&\sum_{t=1}^{T-1} |Clus_t| \cdot |Clus_{t+1}| \cdot O(2^{\min\{a_t, a_{t+1}\}}) \\
&\leq \left[(T-1) \cdot \max_{t \in \{1,\ldots,T\}} \{|Clus_t|^2\} \right] \cdot O(2^{a_{max}})
\end{aligned}$$

with average dimensionality $a_t := \sum_{C_{t,i} \in Clus_t} \frac{|S_{t,i}|}{|Clus_t|}$ and maximal average dimensionality $a_{max} := \max_{t=1,\ldots,T}\{a_t\}$. Thus, our method scales linear with the number of points in time but exponential with the cluster dimensionality.

As a proof of concept, we generate synthetic data comprising two time steps each with 1,000 objects and 10 hidden clusters. In Fig. 5 we depict the runtime of our method for an increasing number of relevant dimensions per cluster. While the runtime of our approach increases exponentially w.r.t. the number of relevant dimensions, the absolute runtime is still acceptable. Moreover, since in real world data the cluster dimensionality is often much smaller than the data dimensionality, we believe that our method is applicable to a broad range of data sets.

4 Experimental evaluation

4.1 Setup

To evaluate the tracing quality of our approach we use real world and synthetic data. For real world data we use scientific grid data reflecting oceanographic characteristics as temperature and salinity of the oceans[1]. It contains 20 time steps, 8 dimensions,

[1] Provided by the Alfred Wegener Institute for Polar and Marine Research, Germany.

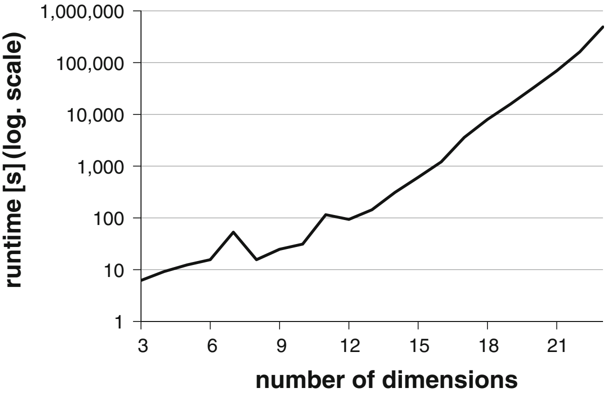


Fig. 5 Scalability with respect to the cluster dimensionality

and 71,430 objects. The synthetic data covers 24 time steps and 20 dimensions. In average, each time step contains 10 clusters with 5–15 relevant dimensions. Since we hide all kinds of developments (emerge, converge, diverge, or disappear) and evolution (subspace and value changes) within this data, these values are slightly changed. In our experiments we focus on the quality of our approach.In synthetic data sets the correct mappings between the clusters are given. Based on the detected mappings of our approach we calculate the precision and recall values: we check whether our approach detects all but only the true mappings between clusters. For tracing quality we use the F1 value corresponding to the harmonic mean of recall and precision. Our approach tackles the problem of tracing clusters with varying subspaces and is based on object-value-similarity. Even if we would constrain our approach to handle only full-space clusters as existing solutions, such a comparison is only possible when we artificially add object ids to the data (to be used by these solutions). Tracing clusters based on these artificial object ids, however, cannot reflect the ground truth in the data. Summarized, comparisons to other approaches are not performed since it would be unfair. We use Opteron 2.3 GHz CPUs and Java6 64bit.

4.2 Tracing quality

In this section we analyze how the different parameters of our algorithm affect the cluster tracing effectiveness. The influence of γ is evaluated in Fig. 6 for three different values of τ using synthetic data. By γ we determine the trade-off between subspace similarity and value similarity in our overall distance function. The objective of this function is to allow that clusters can gain or lose dimensions, and also to allow that cluster object values can slightly shift. Obviously we want to prevent extreme cases for a meaningful tracing, i.e., subspace similarity with no attribute similarity at all ($\gamma \rightarrow 0$), or the other way round. This is confirmed by the figure, as the tracing quality highly degrades, if γ reaches 0 or 1 for all τ values. As $\gamma = 0.3$ enables a good tracing quality for all three τ, we use this as default. Note that with the threshold τ we can directly influence how many cluster mappings are created. The figure shows

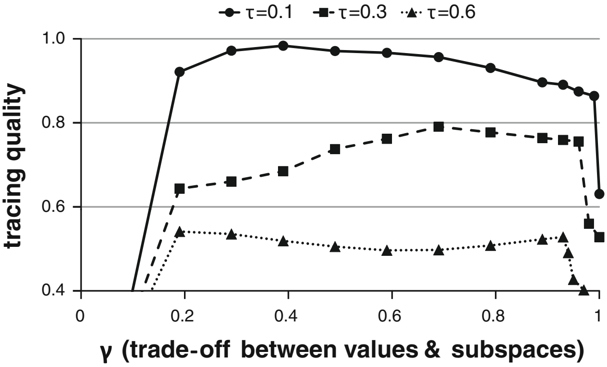


Fig. 6 Tracing quality for different γ & τ

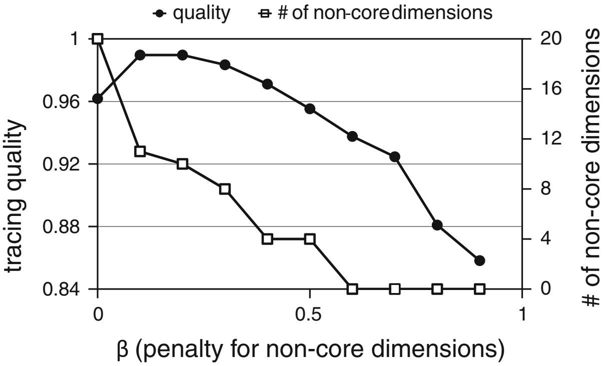


Fig. 7 Evolutions of core dimension concept

that $\tau = 0.1$ is a good trade-off and is thus used as default. With a bigger τ the tracing quality worsens: too many mappings are created and we cannot distinguish between meaningful or meaningless mappings. The same is true for $\tau \rightarrow 0$: no clusters are mapped and therefore the clustering quality reaches zero; thus we excluded plots for $\tau \rightarrow 0$.

The core dimension concept is evaluated in Fig. 7. We analyze the influence on the tracing quality (left axis) with a varying β on the x-axis; i.e., we change the penalty for non-core dimensions. The non-core dimensions are those shared dimensions of two compared clusters in which large changes are allowed. Remember, non-core dimensions are a different concept than non-relevant ones; non-core dimensions are shared relevant dimensions with differing values. The higher the penalty, the more dimensions are included in the dimension core; i.e., more shared dimensions are used for the value-based similarity. In a second curve, we show the absolute number of non-core dimensions (right axis) for the different penalties: the number decreases with higher penalties. Note that in this experiment the exact number of non-core dimensions in the synthetic data is 10. We can draw the following conclusions regarding tracing quality: A forced usage of a full core (all shared dimensions, $\beta \rightarrow 1$) is a bad choice, as there

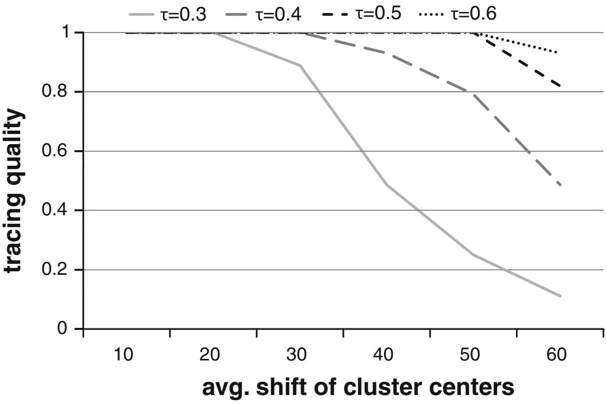


Fig. 8 Varying shift of cluster centers

can be some shared dimensions with different values. By lowering the penalty we allow some dimensions to be excluded from the core and thus we can increase the tracing quality. With $\beta = 0.1$ the highest tracing quality is obtained; this is plausible as the number of non-core dimensions corresponds to the number that is existent in the data. A too low penalty, however, results in excluding nearly all dimensions from the core (many non-core dimensions, $\beta \rightarrow 0$) and hence the quality drops for this case. In the experiments, we use $\beta = 0.1$ as default.

The objective of our tracing approach is to map clusters of similar behavior, i.e., two clusters are mapped if the corresponding object values are similar; thus, strong value differences in core dimensions of two compared clusters should prevent a mapping. This is evaluated in Fig. 8 with synthetic data. On the x-axis, the average cluster center shift between consecutive time steps is plotted. The figure shows that with greater shifts less clusters are mapped and thus the tracing quality degrades. It can also be seen that this effect can be counterbalanced with a higher value of τ; by this, greater cluster center shifts are allowed.

The effect of input clustering quality on cluster tracing quality is evaluated in Fig. 9. We analyze how mappings between clusters are affected in the case that hidden clusters are incorrectly identified by the clustering algorithm. This is achieved by a varying hypercube width in our clustering model. The figure shows that clustering quality (measured via RNIA (Patrikainen and Meila 2006)) and tracing quality are highly correlated; a decreasing clustering error results in an increasing tracing quality. Thus, for meaningful cluster tracing a clustering algorithm providing high-quality clusters is essential.

4.3 Detection of behavior developments

In the following, we analyze whether our model is able to detect the different behavior developments. Up to now, we used our enhanced clustering method that utilizes the predecessor information and the distance based quality function. Now, we additionally

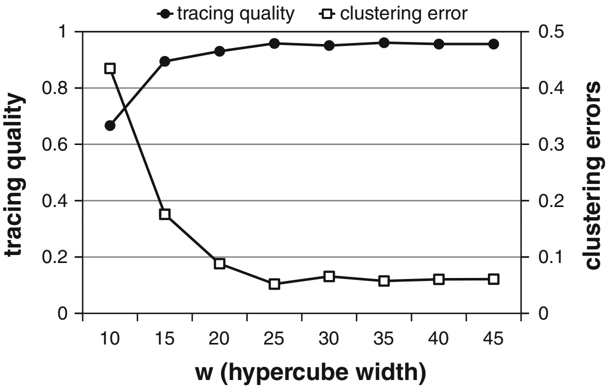


Fig. 9 Influence of clustering quality

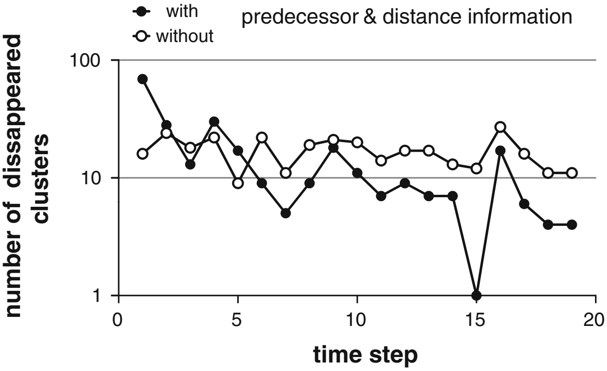


Fig. 10 Comparison of our approach with and w/o predecessor information and distance quality function

compare this method with a variant that performs clustering of each step independently. Intuitively, this corresponds to the idea that each time step is the first one, i.e. ,without preceding information. In Fig. 10 we use the oceanographic data set and we determine for each time step the number of disappeared behaviors for each clustering method. The experiment indicates, that the number of unmapped clusters for the approach without any predecessor or distance information is larger than for our enhanced approach. By transferring the clustering information between the time steps, can map a larger amount of clusters from one time step to the next. We map clusters over a longer time period; thus, yielding a more meaningful tracing of evolving clusters.

The aim of tracing is not just to map similar clusters but also to identify different kinds of evolution and development. In Fig. 11 we plot the number of clusters that gain or lose dimensions and the four kinds of development cumulated over all time steps. Beside the numbers our approach detects, we show the intended number based on this synthetic data. The first four bars indicate that our approach is able to handle dimension gains or losses; i.e., we enable subspace cluster tracing, which is not considered by other models. The remaining bars show that also the developments

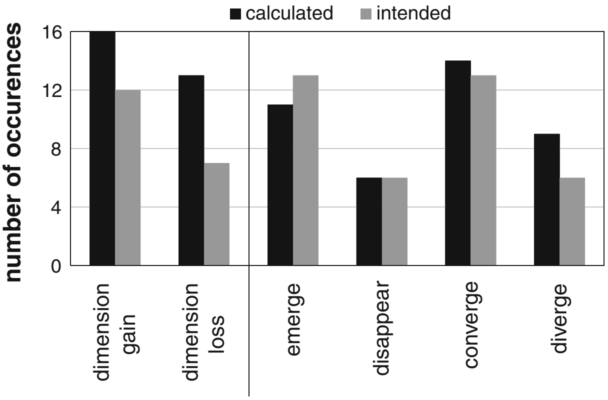


Fig. 11 Cumulated number of evolutions and developments over 24 time steps on synthetic data

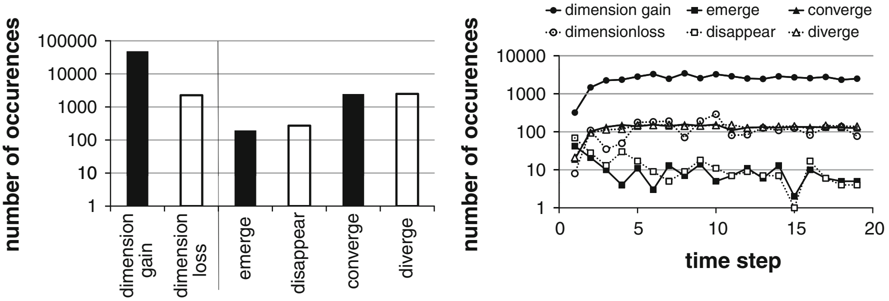


Fig. 12 Number of evolutions and developments on real world data; *left* cumulated over 20 time steps, *right* for each time step

can be accurately detected by our model. Overall, the intended transitions are found by our tracing. In Fig. 12 we perform a similar experiment on real world data. We report only the detected number of patterns because exact values are not given. On the left we cumulate over all time steps. Again, our approach traces clusters with varying dimensions. Accordingly, on real world data it is a relevant scenario that subspace clusters lose some of their characteristics. Thus, it is mandatory to use a tracing model that handle these cases, as our model does. The developments are also identified in this real world data. To show that the effectiveness is not restricted to single time steps, we analyze the detected patterns for each time step individually on the right. Based on the almost constant slopes of all curves, we see that our approach performs effectively.

4.4 Application scenario

To demonstrate that our tracing approach detects reasonable mappings on real world data, a tracing result for the oceanographic grid data is shown in Fig. 13. In the figure, different colors correspond to different clusters. Our method detects and traces sev-

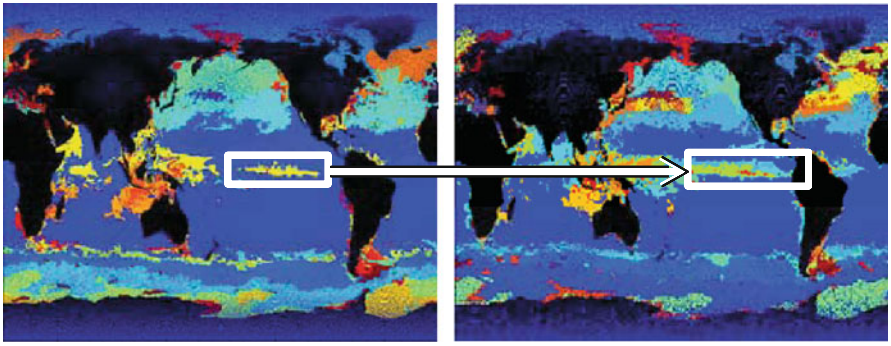


Fig. 13 Clusterings of oceanographic grid data for Jan and Feb and one mapping

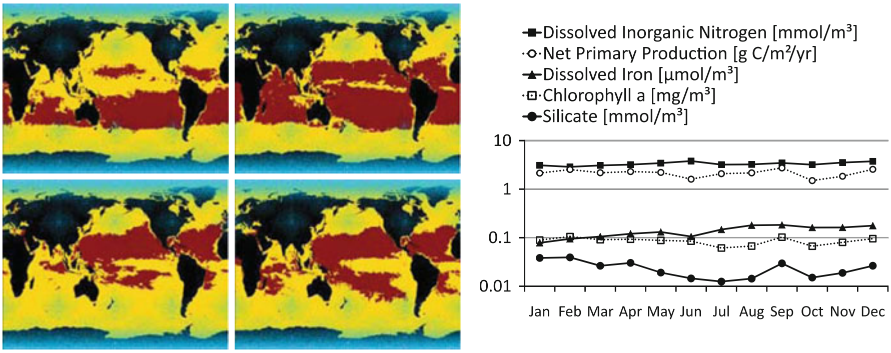


Fig. 14 Tracing of a single cluster over a 1 year period; *left* illustrations of Jan., Apr., Jul., and Oct.; *right* selection of the corresponding values per time step

eral of the oceanic provinces (Longhurst 1998). For example, a cluster similar to the Pacific Equatorial Divergence Province is found in both time steps and our method accomplishes a mapping between these clusters; we illustrate this mapping in the figure.

Another traceable cluster is the cluster representing the low productivity regions of the oceans located at the position of the subtropical gyres. We analyzed the temporal behavior of this cluster over a time period of 1 year, and 4 months are illustrated in Fig. 14 (left). A selection of the corresponding values per time step is given on the right. Noticeable in this figure is the increase of chlorophyll, net primary production, and silicate in September. This could indicate a connection to phytoplankton blooms (Siegel 2002) occurring regularly in temperate and sub-polar water areas. In winter, waters are well mixed and nutrients circulate up from bottom waters. As soon as the ocean warms in late spring, the warm water will stay at the top of the water column as it is less dense. At the same time light level increases and phytoplankton population grows exponentially. In most cases the available nutrients are used up within weeks or months. Sometimes a second bloom occurs at autumn.

5 Conclusion

In this article, we proposed a model for tracing evolving subspace clusters in high dimensional temporal data. In contrast to existing methods, we trace clusters based on their behavior; that is, clusters are not mapped based on the fraction of objects they have in common, but on the similarity of their corresponding object values. Therefore, our approach is especially suited for climate data. An example are oceanographic data, where values are recorded by sensors, which are positioned at fixed grid cells. For maximal flexibility, our tracing approach distinguishes several developments of behavior. We enable effective tracing by introducing a novel distance measure that determines the similarity between clusters; this measure comprises subspace and value similarity, reflecting how much a cluster has evolved. In the experimental evaluation we showed that high quality tracings are generated.

As future work we plan to generate the mapping graph based on global optimization considering all possible mappings and all possible points in time simultaneously. Until now, the mappings between two successive points in time are independent of the other time steps. Besides efficiency issues, global optimization has a second challenge: predecessor information to increase tracing quality cannot be easily utilized since each clusterings has to be given a-priori, before such an optimization. We also want to investigate other statistical representations of clusters that inherently cope with the temporal characteristics of data, for example based on dynamic Gaussian processes. Possible challenges adapting such approaches include the handling of complex clusters evolutions as emerging or disappearing clusters.

Acknowledgements We thank the Alfred Wegener Institute for Polar and Marine Research for providing the Oceanographic Grid Data. This article has been supported by the UMIC Research Centre, RWTH Aachen University.

References

Aggarwal CC (2005) On change diagnosis in evolving data streams. IEEE TKDE 17(5):587–600

Aggarwal CC, Han J, Wang J, Yu PS (2003) A framework for clustering evolving data streams. In: VLDB, pp 81–92

Aggarwal CC, Han J, Wang J, Yu PS (2004) A framework for projected clustering of high dimensional data streams. In: VLDB, pp 852–863

Agrawal R, Gehrke J, Gunopulos D, Raghavan P (1998) Automatic subspace clustering of high dimensional data for data mining applications. In: ACM SIGMOD, pp 94–105

Barnett T, Pierce D, Schnur R (2001) Detection of anthropogenic climate change in the world's oceans. Science 292(5515):270

Boriah S, Kumar V, Steinbach M, Potter C, Klooster SA (2008) Land cover change detection: a case study. In: ACM SIGKDD, pp 857–865

Böttcher M, Höppner F, Spiliopoulou M (2008) On exploiting the power of time in data mining. ACM SIGKDD Explorations 10(2):3–11

Brodeur R, Mills C, Overland J, Walters G, Schumacher J (1999) Evidence for a substantial increase in gelatinous zooplankton in the bering sea, with possible links to climate change. Fisheries Oceanograp 8(4):296–306

Cao F, Ester M, Qian W, Zhou A (2006) Density-based clustering over an evolving data stream with noise. In: SIAM SDM, pp 328–339, 2006

Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the EM algorithm. J Royal Stat Soc. Series B, pp 1–38

Ester M, Kriegel H-P, JS, Xu X (1996) A density-based algorithm for discovering clusters in large spatial databases with noise. In: ACM SIGKDD, pp 226–231

Fu T (2011) A review on time series data mining. Eng Appl Artif Intel 24(1):164–181

Gaffney S, Smyth P (1999) Trajectory clustering with mixtures of regression models. In: ACM SIGKDD, pp 63–72

Günnemann S, Kremer H, Seidl T (2010) Subspace clustering for uncertain data. In: SIAM SDM, pp 385–396

Hinneburg A, Aggarwal CC, Keim DA (2000) What is the nearest neighbor in high dimensional spaces? In: VLDB, pp 506–515

Hoegh-Guldberg O (1999) Climate change, coral bleaching and the future of the world's coral reefs. Marine Freshw Res 50(8):839–866

Hoffman F, Hargrove WJr, Erickson DIII, Oglesby R (2005) Using clustered climate regimes to analyze and compare predictions from fully coupled general circulation models. Earth Interact 9(10):1–27

Huntington T (2006) Evidence for intensification of the global water cycle: Review and synthesis. J Hydrol 319(1-4):83–95

Jensen CS, Lin D, Ooi BC (2007) Continuous clustering of moving objects. IEEE TKDE 19(9):1161–1174

Kalnis P, Mamoulis N, Bakiras S (2005) On discovering moving clusters in spatio-temporal data. In: SSTD, Springer, pp 364–381

Kremer H, Günnemann S, Seidl T (2010) Detecting climate change in multivariate time series data by novel clustering and cluster tracing techniques. In: IEEE ICDM Workshops, pp 96–97

Kremer H, Kranen P, Jansen T, Seidl T, Bifet A, Holmes G, Pfahringer B (2011) An effective evaluation measure for clustering on evolving data streams. In: ACM SIGKDD, pp 868–876

Kriegel H-P, Kröger P, Zimek A (2009) Clustering high-dimensional data: a survey on subspace clustering, pattern-based clustering, and correlation clustering. ACM TKDD 3(1):1–58

Li Y, Han J, Yang J (2004) Clustering moving objects. In: ACM SIGKDD, pp 617–622

Liao TW (2005) Clustering of time series data: a survey. Patt Recogn 38(11):1857–1874

Longhurst A (1998) Ecological geography of the sea. Academic Press, London

Müller E, Günnemann S, Assent I, Seidl T (2009) Evaluating clustering in subspace projections of high dimensional data. In: VLDB, pp 1270–1281

Parsons L, Haque E, Liu H (2004) Subspace clustering for high dimensional data: a review. ACM SIGKDD Explorations 6(1):90–105

Patrikainen A, Meila M (2006) Comparing subspace clusterings. IEEE TKDE 18(7):902–916

Procopiuc CM, Jones M, Agarwal PK, Murali TM (2002) A monte carlo algorithm for fast projective clustering. In ACM SIGMOD, pp 418–427

Rosswog J, Ghose K (2008) Detecting and tracking spatio-temporal clusters with adaptive history filtering. In: IEEE ICDM Workshops, pp 448–457

Siegel D, Doney S, Yoder J (2002) The North Atlantic spring phytoplankton bloom and Sverdrup's critical depth hypothesis. Science 296(5568):730

Spiliopoulou M, Ntoutsi I, Theodoridis Y, Schult R (2006) MONIC - modeling and monitoring cluster transitions. In: ACM SIGKDD, pp 706–711

Steinbach M, Tan P-N, Kumar V, Klooster SA, Potter C (2003) Discovery of climate indices using clustering. In: ACM SIGKDD, pp 446–455

Vlachos M, Gunopulos D, Kollios G (2002) Discovering similar multidimensional trajectories. In: IEEE ICDE, pp 673–684

Yiu ML, Mamoulis N (2003) Frequent-pattern based iterative projected clustering. In: IEEE ICDM, pp 689–692

Zhou D, Li J, Zha H (2005) A new mallows distance based metric for comparing clusterings. In: ICML, pp 1028–1035