Subspace-Map: Interactive Visual Analysis for Subspace Data with a Map Metaphor

Category: Research

Abstract— Subspace analysis for high-dimensional data is extremely challenging due to the huge exploration space. We propose Subspace-Map, a novel approach with a map metaphor to support the interactive exploration of various subspaces. Specifically, we use a subspace search algorithm to find a moderate number of potentially valuable subspaces. Then we visualize each subspace as a city on the map. Similar cities are clustered into provinces and countries, whose features reveal the common data and dimensional patterns that can guide users in constructing desired subspaces. With the map, users can grasp an overview of the exploration space. They can also explore different subspaces via recommended tour routes. We provide various kinds of evaluation to demonstrate the effectiveness of Subspace-Map, including cases with real-world data, experiments with user feedback, and a comparison with state-of-the-art subspace data visualizations.

Index Terms—High-Dimensional Data, Subspace Analysis, Map Metaphor

1 Introduction

In a high-dimensional dataset, each data item is characterized by multiple attributes. However, some of the attributes may not be as informative as the others. Worse still, these redundant attributes may bury deep some important patterns such as clusters and correlations. For example, the physical attributes of a person (e.g. weight and height) are not very helpful for the relational analysis between education level and income. The more time we spend studying redundant attributes, the less likely it is to reveal the information of interest. Therefore, experienced analysts usually pick a small dimension subset that is most relevant to the task at hand before going deep into details. The data space formed by partial dimensions is usually called a **subspace**.

Subspaces can be classified into two types: the axis-aligned and the non-axis-aligned. Axes of the former are parallel to the original dimensions, while axes of the latter are weighted combinations of the original dimensions. Subspaces generated by linear dimensionality reduction [51] are usually non-axis-aligned. They excel at preserving certain data features, but seldom preserve semantics of the original dimensions. In this paper, we only focus on the axis-aligned subspaces, since they are easier to use and interpret for users. We refer to them as "subspaces" for short.

Subspace analysis could be extremely complicated given the challenges it poses to analysts:

- **The exploration space is too large due to the curse of dimensionality.** The curse of dimensionality is a notion referring to various problems that occur when analyzing data in high-dimensional spaces. In combinatorics, it appears as the combinatorial explosion problem. For a d-dimensional dataset, there are altogether $2^d - d - 1$ subspaces with no less than 2 dimensions. Each additional dimension doubles the efforts needed to explore all possible combinations. Without any mental map, users are easily overwhelmed faced with such a huge exploration space.

- **The interplay is too complicated between dimension and data pattern.** Including or excluding a single dimension may seem like a subtle change, but it can lead to dramatic data changes. Unable to foresee the data changes, analysts may choose dimensions merely based on semantics, which could lead to undesired results.

- **There lacks a direction for subspace exploration.** After exploring one subspace, analysts often face the same difficulty: which to explore next? They may search for a candidate with the least dimension and data change. However, such a trial-and-error process could be extremely tedious.

For many high-complexity problems, heuristic algorithms are often adopted to provide feasible solutions. Subspace search is no exception. Various algorithms [26,30] have been proposed to find out subspaces with valuable data clusters. However, such algorithms easily produce redundant results that need to be further organized with the help of visualization [23,44]. Besides, they provide no guidance for dimension selection, making it hard for users to manually adjust the subspaces. There are also methods aiming to guide users in subspace exploration. However, they either rely on inefficient manual planning [14,55] or only apply to 2D subspaces [33].

To address the above challenges, we aim to achieve three goals:

- **G1** Provide an overview of the exploration space to help users understand relationships between subspaces and build up their mental maps.

- **G2** Reveal the interplay between dimension and data pattern, thus provide guidance for dimensional decisions.

- **G3** Guide users through the exploration space, so that they can learn in a short time about the representative subspaces and their differences.

We propose Subspace-Map, a visualization approach that utilizes map metaphors to present an overview of subspaces and guide users in the exploration. In Subspace-Map, the exploration space is visualized as a geographic map with each subspace conceptualized as a city (**G1**). The landscape of each city is the data patterns, which are shaped by the natural factors (i.e. dimension combination). By comparing the landscapes, we are able to cluster similar cities into larger regions like provinces and countries. To better describe each cluster, we extract the featured dimensions and data patterns that are shared by most cluster members. Such a high-level summarization is able to reveal the complex dimension-data interplay (**G2**). Routes are built to go through various cities, allowing users to explore different types of subspaces in a gradual way (**G3**).

The remainder of this paper is structured as follows. In Section 2, we briefly review the literature regarding high-dimensional data visualization and subspace analysis. We derive design considerations and introduce the conceptual design of Subspace-Map in Section 3. Details for map construction are elaborated in Section 4. Section 5 introduces the user interface and interactions of our prototype system. In Section 6, we evaluate Subspace-Map with real-world cases and user studies. Section 7 discusses current deficiencies and future work. The final section concludes the whole paper.

2 Related Work

In this section, we first take a look at the general high-dimensional data visualizations to see how people extract the important dimensions. Then we introduce the subspace mining techniques along with the related visualization methods. We also review the visualizations using the map metaphors.

2.1 Dimension Selection in High-Dimensional Data

Dimension selection is an important task that seeks to reduce dimensionality while maintaining useful characteristics. It requires the analyst to have rich experience with the data, which is seldom the case in data analysis. A lot of research works have been carried out to address this challenge. They can be divided into three categories.

The first type of method is based on the similarity between dimensions. In the early research of parallel coordinates [22], Yang et al. [54] proposed to cluster the similar dimensions and extract a representative dimension for each cluster, which is the centroid or average of the cluster members. Turkay et al. [46] applied dimensionality reduction and statistical modeling to generate the representative factors. Zhang et al. [56, 57] utilized correlation strengths and carried out more delicate rules for dimension clustering.

The second type aims at selecting important dimensions suggested by quality metrics. The Rank-by-Feature Framework [37] allows to rank dimensions based on user-specified statistical criteria. It is also applied to select dimension pairs in a scatterplot matrix (SPLOM) [6] and order axes in parallel coordinates. Some measures of pattern salience in scatterplots [24, 38, 43] have been used to rank the plots in a SPLOM [19, 33] to improve the exploration efficiency, including the well-known Scagnostics [49, 50]. For parallel coordinates, metrics are also proposed to detect the inter-axis patterns [12, 24] and rank different ordering schemes [25, 34]. We refer to [1] for a comprehensive review. Our method falls into this type, where we extract important dimensions by examining whether they dominate the pattern.

As opposed to the automatic methods, the third type interactively selects dimensions without making any assumptions about what is interesting. Voyager [52] and its advanced version [53] allow free selection from a complete list and recommend dimensions that may be overlooked. Sarvghad et al. [36] achieved the same goal by simply showing the coverage of the dimensions explored. Turkay et al. [45] proposed to brush dimensions in the projection to support dual space analysis. It was later extended into a more extensible framework by Yuan et al. [55], in which the exploration process is organized in a hierarchical way.

2.2 Subspace Mining and Visualization

Dimension selection methods work well in identifying a small number of subspaces or low-dimensional subspaces. Subspace mining techniques become indispensable when it comes to high-dimensional cases. They are designed to find subspaces with interesting patterns, which in most cases are hidden clusters. Hence, they are also called subspace clustering techniques [26, 30].

Subspace clustering often produces a large number of results and easily causes redundancy. Tatu et al. [44] developed a visual analytics system to help users organize the redundant subspace candidates. It shows the relationship of all candidates by comparing their dimension similarity. TripAdvisorND [31] also shows a projection that measures dimension differences. Pattern Trails [23] adopts a 1D layout to help users trace data changes across different subspaces. We provide a complete comparison with these methods in Sect. 6. Watanabe et al. [47] did not depend on subspace clustering, but proposed a bi-clustering approach to divide the data into multiple complementary subparts.

2.3 Map Metaphor for Non-Spatial Data Visualization

Maps are a very popular and important technique to depict the spatial relationships between elements. Many different types of maps, such as choropleth map and road map, have been created to convey a variety of information. As a familiar and easy-to-understand way, it can also be used as a carrier for non-spatial information. In the field of visualization, how to create non-geographic visualizations with the help of geographic metaphors has also been studied [10] and several attempts have been made [39, 40].

GMap [16] is a seminal work to accelerates research in this field. It was designed to highlight communities in a network, and was later applied to the analysis of dynamic graphs [29], video content [28], etc. Cao et al. [5] proposed to visualize multi-label data with ternary plots in uniform triangular grids. Map metaphors are also widely used in the social media domain. Chen et al. [7] introduced D-Map for analyzing user-centric information diffusion patterns. Chen et al. [8] proposed R-Map for analyzing the information reposting process. Recently, a survey has been conducted [20] to give an overview of the literature on maplike visualizations. Targeting at subspace visual analysis, Subspace-Map introduces abundant map metaphors and supports multi-level exploration.

3 The Design of Subspace-Map

In this section, we first discuss our design considerations and why we choose to adopt a map-like representation. We then introduce the visual encodings and explain how the design considerations are met.

3.1 Design Considerations

We formulate three goals in Section 1. As we shall see, the design must meet several requirements in order to achieve the aforementioned goals. Many of these requirements are also met by previous works [23, 44] for subspace visual analysis.

DC1: The overview should present a limited number of subspaces and indicate their relationships. Due to the combinatorial explosion, it's impossible to display all subspaces in the same view. To make an effective mental map (G1), the overview should be able to present relationships between different subspaces.

DC2: Strategies are required for choosing a moderate number of potentially valuable subspaces. To meet DC1, we demand algorithms that can reduce the number of visualized subspaces. The chosen subspaces should not only be informative for data analysis but representative enough to reflect important trends in the exploration space.

DC3: Summarization is needed to extract data and dimension features shared by the displayed subspaces. On the one hand, providing summarization can mitigate the data scale problem by promoting higher-level analysis in the overview. On the other hand, summarization of subspaces can reveal the data-dimension interplay inherent to the dataset (G2).

DC4: Recommendations should be made to guide users in subspace exploration. A small number of representatives can be recommended to help users quickly grasp the information of different types of subspaces. Based on the representatives, a tour path [21] with gradual variations can also be suggested to avoid cognitive confusion caused by abrupt data changes. Combined together, they can guide users in deciding the target subspaces as well as their visiting orders (G3).

3.2 Design Choices and Decisions

To start with, we use a toy example (8D Car dataset, 247 subspaces) to refine our design of the overview. We tried various visualization schemes during the design process. The most straightforward way is to present a 2D projection with each point being a subspace [44]. However, when communicating our designs with domain experts, we soon discovered two problems.

First, users expressed their desires to see both trends and details of subspaces. It requires the overview to be space-efficient so that more subspaces can be displayed for revealing trends (**DC2**). There is no limit to how much data a projection can show, but projections could be easily cluttered, making it hard to assign enough space to each subspace for showing its own details. Both 2D [44] and 1D [23] projections suffer from this problem. That is because projections use much space to authentically reflect data similarity. We found that compared to the exact values, relative relationships are more concerned by the users. It allows us to consider designs that are more space-efficient but less accurate in conveying subspace similarities.

Second, users generally found it difficult to use the subspace concepts for data analysis. Subspace concepts are often hard to understand and use. A subspace is anonymous and can only be referred to by its entire dimension combination. Things become more complicated when users try to make sense of different combinations or refer to higher-level notions (e.g. a set of subspaces). It inspired us to adopt metaphors in later designs to ease data analysis.

Through the above analysis, we can derive two requirements for the overview's style. First, it should be space-efficient and occlusion-free. It should be able to show hundreds to thousands of subspaces, with each one possessing an individual display space. Second, it should be familiar to users in other contexts so that metaphors can be used to facilitate comprehension and communication.

With these requirements, it's not hard to see that maps are perfect candidates for the overview. A geographic map can show thousands of regions with details (e.g. terrains, names) displayed inside them. Besides, maps are frequently used in all sorts of situations to help people communicate geographical concepts. They have also been adopted in previous works [7, 16, 20] to visualize non-spatial data information.

As the next step, we decide which type of map to use. Hogräfer et al. [20] classified four types of map-like visualizations for abstract data based on their core primitives. Point-based maps (e.g. projection maps [44]), as stated before, suffer from occlusions. Line-based maps (e.g. metro maps [32]) emphasize connections, while field-based maps (e.g. contour maps [35]) demand data continuity, neither of which are inherent features of subspace data. Area maps (e.g. geographic maps) fit our requirements well. The enclosing nature of areas allows us to display individual concepts like subspaces or their higher-level summarization (**DC3**). The layout of areas can reflect relationships between concepts (**DC1**).

According to Hogräfer et al., there are three ways to generate an area map: i.e. geometric hulls, irregular tessellation and regular grids. Geometric hulls [42] are areas containing groups of points. They visualize high-level summarization but don't guarantee the readability of low-level data. We do not consider irregular tessellation (e.g. Voronois [16]) because their cells in various shapes and sizes could imply inequality among subspaces. Without assuming users' analysis interests, we cannot impose any weighting on the data, not to mention misleading users about it. Besides, irregular cells make it hard to unify the display style of each subspace. Given the above considerations, regular grids, i.e. tilings with regular polygons, seem like the best choice to construct our overview. There do exist other forms of area maps, such as a force-directed layout of circles [2], but they are inherently less space-efficient than regular grids.

Grünbaum and Shephard have done in-depth research [17, 18] to systematically classify various types of tilings. They point out that amoung all possibilities, regular tilings (i.e. edge-to-edge tilings by congruent regular polygons) are the most popular ones that have been widely used since antiquity. We stick to regular tilings to make the visualization more approachable. Only three kinds of regular tilings exist: the triangular, the square, and the hexagonal tiling. It has been proved by previous works [4, 20] that hexagonal grids can express more adjacency relationships (6 neighbors) than the other two. Since adjacency is the major visual cue in tilings for indicating closeness (**DC1**), we decide to adopt hexagonal grids.

3.3 The Map Metaphors

As discussed before, one of the major benefits of maps and hexagonal grids is their wide usage in the general public. It allows users to perform subspace analysis using geographical concepts. Before diving into algorithms, we would like to introduce the map metaphors by explaining correspondence between geographical and subspace concepts.

In Subspace-Map, we treat the exploration space as an unknown **world** that is made up of **lands** (subspaces) and **oceans** (distance between subspaces). We visualization designers are like **cartographers**, **exploiters** and **tour guides**.

3.3.1 Geography

As cartographers, we measure the topography and draw a map to depict it. Since it is a fictional world, there is no ground truth about how the subspaces are arranged. Making use of Gestalt Principles [41], we assume that similar subspaces should be closely placed to imply similarity (**DC1**, Fig. 1).

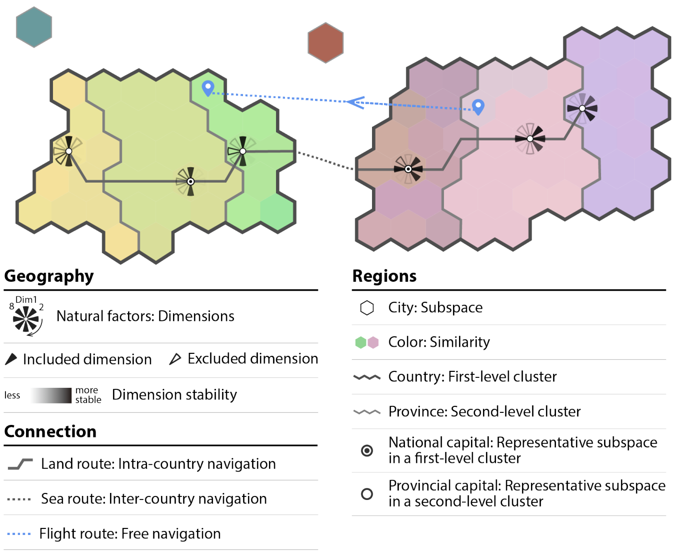
Geography
Dim1
8
2
Natural factors: Dimensions
Included dimension
Excluded dimension
less
more stable
Dimension stability
Connection
Land route: Intra-country navigation
Sea route: Inter-country navigation
Flight route: Free navigation
Regions
City: Subspace
Color: Similarity
Country: First-level cluster
Province: Second-level cluster
National capital: Representative subspace in a first-level cluster
Provincial capital: Representative subspace in a second-level cluster

Fig. 1. The visual encoding of Subspace-Map. Each hexagon represents a subspace. Each region represents a cluster of subspaces. Various map metaphors are utilized to make subspace analysis more approachable.

With each subspace conceptualized as a piece of land, we define the **landscape** as its data distribution, which is usually the primary interest to **tourists** (users). Dimension settings are analogous to **natural factors** that play a decisive role in shaping landscapes. Apart from sightseeing, tourists often compare different landscapes to understand what kind of natural factors are responsible for the local features (**DC3**). Lands with similar landscapes will be joined together to form a **continent** (a cluster of subspaces), like the dry and freezing Greenland, or the humid and hot Amazon Rainforest. **Islands** (outliers) are lands with highly distinctive landscapes. They are separated from other continents by **oceans**, which are void spaces with no specific meaning.

3.3.2 Regions

As exploiters, we establish cities and set up administrative divisions. Lands with beautiful scenery are perfect candidates to build up a tourist **city**, which refers to a subspace with valuable data patterns. We choose them via a certain aesthetic standard (subspace interestingness metric) that most likely meets the public taste (**DC2**). Only the cities will be displayed on the tourist map.

It's impossible and unnecessary for a tourist to visit all cities. Therefore, we set up administrative divisions like **provinces** and **countries** to showcase regional characteristics in a higher-level image (**DC3**). They are clusters in different granularities. For each region, a **capital city** (representative subspace) is chosen to represent its peers. Cities that don't resemble their peers are called **municipalities**. They are outliers within a larger cluster.

With these concepts, we create a hierarchy of subspace summarization that has three levels (i.e. national, provincial, and urban level). High-level summarization indicates subspace relationships, while low-level details present data and dimension patterns.

3.3.3 Transportation

As tour guides, we need to schedule sightseeing paths for tourists (**DC4**). We set up **flight routes** (the blue path in Fig. 1) to facilitate long-distance trips between different countries. Tourists on a flight may experience jet lag that is caused by swift data changes between distinct subspaces. We also set up **land routes** and **sea routes** (the black path in Fig. 1), which connect neighboring cities on the same continent and across the sea, respectively. Travels over land and sea are much slower but provide a more stable and comfortable tour experience.

4 Subspace-Map Construction

In this section, we describe the processing of the data and the steps for map construction.

4.1 Data Preprocessing

The first step in data preprocessing is normalization, which is an essential step to make the dimensions comparable. We then sample the subspaces, measure the similarity between subspace samples, cluster similar subspaces together, and extract features such as data stability, dimension stability, etc.

Subspace sampling. To provide a good overview of the entire exploration space, we need to ensure the diversity and representativeness of the subspace samples. For this purpose, we keep the sampling frequency to remain the same for each dimension combination and each dimension.

More specifically, for sampling m subspaces from a d-dimensional dataset, we firstly add the d-dimensional subspace to our sampling results as it characterizes the full-dimensional space. We do not consider 1D subspaces due to their simplicity. For k-dimensional subspaces where k ranges from 2 to $d-1$, we use inverse transform sampling to obtain the number of samples m_k in k-dimensional subspaces ($\sum_{i=2}^{d-1} m_i = m-1$). Then we randomly sample m_k subspaces from $\binom{d}{k}$ k-dimensional subspaces and add them to the results.

Similarity measure. Even with normalization, traditional similarity measures such as Euclidean distance are still not suitable for subspace comparison, because variations in subspace dimensionality affect the distance calculation. Jäckle et al. [23] proposed to compare the projected distance, which inevitably suffers from information loss.

Similar to the method adopted by Tatu et al. [44], we use the k-NN (k-Nearest Neighbors) topology to measure similarity. Specifically, we compute the k-NN list for each data item. The similarity between the same data item in two subspaces is evaluated by the percentage of agreement between their k-NN lists. By averaging the similarity of all data items, we obtain the similarity between the two subspaces, which can be further converted to distance by subtracting it from 1. In this way, we get the distance matrix of all sampled subspaces. We can also compute the k-NN list for each subspace by sorting the corresponding row or column of the distance matrix in ascending order and selecting the first k subspaces. Note that the parameter k for data items and subspaces is not the same, and does not need to be equal. To distinguish, we denote k for data items as k_d and k for subspaces as k_s. They are set based on the number of data items and the number of subspaces, respectively.

Clustering. To obtain information at different granularities, we hierarchically cluster subspaces by DBSCAN [15], which clusters points in density regions together and marks points in low-density regions as outliers. DBSCAN has two parameters and we currently determine them manually based on the number of input subspaces. The basic criterion is to select the parameters when the output is relatively stable, i.e. perturbations in the parameters do not affect the results. In the future, we may apply heuristic methods [15, 27] to help specify parameters quickly.

The clustering is up to 2 levels. To avoid visual intersection of clusters and to improve readability and layout interpretation [48], we compute the projection by applying MDS [11] to the distance matrix and perform DBSCAN on the projection. We use MDS because it globally preserves the pairwise distance between subspaces. Other dimensionality reduction techniques are also optional. This operation produces first-level clusters and first-level outliers that correspond to countries and islands in the map. We then proceed to execute DBSCAN on the first-level clusters to obtain second-level clusters and second-level outliers, which correspond to provinces and municipalities.

Data stability. For the data item in a subspace, the stability is obtained by averaging the percentage of agreement on the k_d-NN list for the corresponding data item in the k_s-NN subspaces for that subspace. For the data item in a subspace cluster, its stability is calculated by averaging the stability in each subspace. With the stability, we can find out which data items change slightly and which change drastically.

Dimension stability. The dimensions contained in a subspace determine its pattern. We want to figure out why a subspace is similar to others and why a subspace cluster is generated. For a subspace, we count the number of co-occurrences and co-non-occurrences of each dimension between that subspace and its k_s-NN subspaces, and divide by k_s to obtain the dimension stability. For a subspace cluster, we count the number of occurrences of each dimension and divide by the number of subspaces it contains.

Featured dimensions. Based on dimension stability, we set high and low thresholds to find commonly included dimensions and commonly excluded dimensions for subspace clusters. They are defined as featured dimensions.

Representative subspaces. In order to show the main features of a cluster, we select the subspace with the highest average similarity with other subspaces as its representative subspace.

4.2 Map Construction

Subspace-Map is a hexagonal map. The construction of Subspace-Map first places the anchor points on hexagonal grids, and then places other subspaces according to the traversal order as well as the available grids (Fig. 2).

4.2.1 Anchor Point Location and Traversal Order

We start by placing anchor points on the map. The anchor point of a country is the national capital, and the anchor point of an island is itself. Their location is determined from the projection, which has been computed in the clustering.

Next we determine the order in which countries and islands are built, i.e., the traversal order of anchor points. A traversal list is created. We first add to the list the point that has the smallest average distance from the other points. We continue by adding the point with the smallest average distance from all the points in the list. Determining the traversal order of cities in a country is the same as determining the traversal order of anchor points.

4.2.2 Grid Tiling

After placing the anchor points, we place the cities in each country according to their traversal order. We use a location queue to maintain the currently available grids (Fig. 2a). For a used grid, we add its adjacent grids that meet the following two conditions to the queue in clockwise order: (1) the grid is not occupied by other subspaces, and (2) the grid is not a disabled grid. Disabled grids are set up to avoid adjacency between different countries and islands (Algorithm 1-Step 1).

The location queue will be continuously updated according to the clustering relationship between two adjacent subspaces in the traversal order (Algorithm 1-Step 2). If they belong to the same second-level cluster, the queue is updated as usual (Fig. 2b). If not, the current queue is cleared and a new one is rebuilt based on the subspace to be placed (Fig. 2c). We do not rebuild the queue when a second-level outlier is encountered. Second-level outliers are usually clustered at the end of the traversal order, and rebuilding the queue will result in the country having a long tail.

After all the countries and islands are built (Fig. 2d), we make the layout compact (Fig. 2e). We remove empty rows and columns while ensuring that countries and islands do not border. Then we center and enlarge the remaining grids to fill the entire view space (Algorithm 1-Step 3). In the compact layout, the relative position between countries and islands have changed, resulting in the similarity not being well conveyed by position. To cope with this problem, we additionally color-code the subspaces based on similarity. The idea is to project the distance matrix into three-dimensional space, with each dimension representing a variable in the RGB color space.

4.2.3 Additional Map Metaphors

Once the map layout is generated, additional map metaphors, including capital cities, routes, and natural factors, are created to make the map more informative (Fig. 2f).

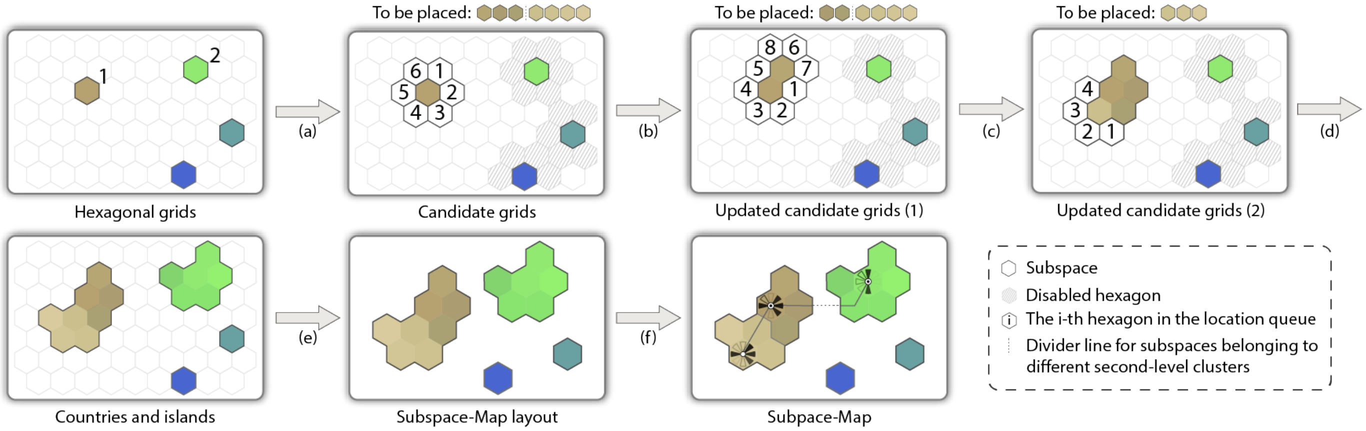
To be placed:
1
2
6
1
5
2
4
3
8
6
5
7
4
1
3
2
4
3
2
1
(a)
(b)
(c)
(d)
(e)
(f)
Hexagonal grids
Candidate grids
Updated candidate grids (1)
Updated candidate grids (2)
Countries and islands
Subspace-Map layout
Subpace-Map
Subspace
Disabled hexagon
The i-th hexagon in the location queue
Divider line for subspaces belonging to different second-level clusters

Fig. 2. The construction process of Subspace-Map: (a) initialize the location queue; (b) update the queue after each placement; (c) reset the queue for a new cluster; (d) continuously tiling; (e) reduce redundant map space; and (f) render map metaphors.

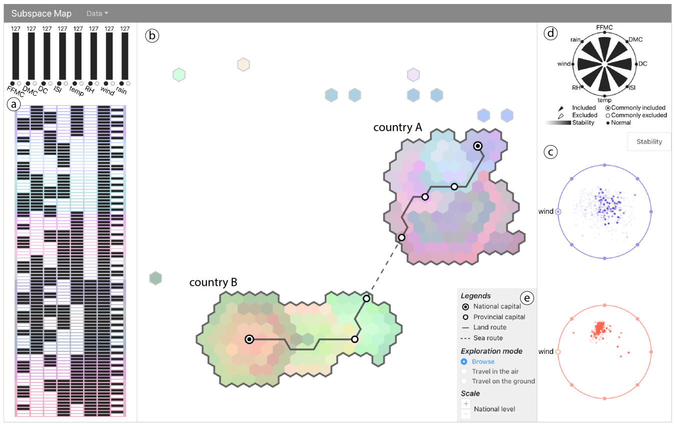

Fig. 3. The user interface of Subspace-Map. (a) Subspace List View shows the dimensions of each subspace sample. Black/White indicates the presence/absence of a certain dimension. (b) Map View shows the distribution of all subspace samples based on their similarities. (c) Map Detail View presents the dimension and data patterns of a selected map region. (d) and (e) explain the visual encoding and map metaphors, and allow users to switch the exploration mode.

The representative subspace of a country or a province is selected as the national capital or provincial capital, respectively. The national capital can also be chosen as the provincial capital. There are three kinds of routes. The flight route supports users to travel between any two cities, the land route connects capital cities of the same country, and the sea routes connects port cities of different countries. The latter two routes are derived by calculating the path that minimizes the distance between the start and end points. To avoid clutter, we further apply minimum spanning tree to reduce the number of land and sea routes. The contained dimensions and dimension stability constitute the natural factors of a city.

Because of the importance of the city landscape (i.e. the data pattern), we do not embed it in the map but rather display it in another view.

Algorithm 1 Map Layout Algorithm

```
Require:
    A hexagonal map $Map$;
    A list of anchor points $V[i]$ with initialized locations $V[i].loc, i = 1, 2 ... N_a$, $N_a$ is the number of anchor points;
    A list of traversal order lists for countries and islands $T[i], i = 1, 2 ... N_a$;
    A list of city-province objects for countries and islands $O[i], i = 1, 2 ... N_a$;
Ensure:
    A list of hexagonal grid lists $V[i]'$ for countries and islands, with assigned locations $V[i]'.loc, i = 1, 2 ... N_a$;
for $i = 0; i < N_a; i++$ do
    $V'[i]$.push($V[i]$)
end for
for $i = 0; i < N_a; i++$ do
    if $T[i].length == 1$ then                                ▷ This is an island
        continue
    end if
    // Step 1: Calculate the disabled grid list
    $G_{disabled}$ = []
    for $j = 0; j < N_a \&\& j != i; j++$ do
        Calculate the adjacent grids $G_{adjacent}$ of $V'[j]$
        $G_{disabled}$.concat($G_{adjacent}$)
    end for
    // Step 2: Maintain the location queue $Q$
    for $j = 0; j < T[i].length - 1; j++$ do
        for each adjacent grid $g$ of $T[i][j]$ do
            if $g$ is not in $G_{disabled}$ && $g$ is empty then
                $Q$.enqueue($g$)
            end if
            $V'[i]$.push($Q$.dequeue())
            if $j > 0 \&\& O[i][j] != O[i][j-1]$ then     ▷ Two cities belong to different provinces
                $Q$.clear()
            end if
        end for
    end for
end for
// Step 3: Make the map compact
Remove empty rows/columns not causing bordering
Center and enlarge $Map$
```

5 Subspace-Map System

The prototype system (Fig. 3) of Subspace-Map mainly consists of three views: the Map View, the Subspace List View, and the Map Detail View.

5.1 Map View

The Map View (Fig. 3b) presents the hierarchical map as introduced in Sect. 3.3. All the cities are shown as hexagons, whose colors and distances indicate subspace similarity. Clusters and sub-clusters are conceptualized as countries and provinces respectively, with the representatives being their capitals. The outliers become islands and municipalities. Users can navigate through different cluster levels by zooming and panning. At the municipal level, the natural factors of each city are shown in a fan-shape glyph (Fig. 3d and Fig. 7a). Each filled/unfilled fan indicates the presence/absence of a dimension. We use transparency to encode the stability of each dimension (see Sect. 4.1), which helps to reveal the shared dimension patterns between each subspace and its neighbors (Fig. 7b). Users can switch to travel mode to trace pattern transitions between subspaces. Two travel modes are available: air-travel and ground-travel. The former supports travel between any city via flight routes, while the latter allows travel between capital cities via land and sea routes (Fig. 6).

5.2 Subspace List View

The Subspace List View (Fig. 3a) provides a list of sampled subspaces. A dimension histogram shows the dimension distribution. The black and white buttons below each dimension bar are used to filter the subspaces that contain or do not contain the corresponding dimension. Each row in the list represents a subspace whose color matches that of the corresponding subspace in the Map View. A black/white rectangle at the corresponding dimension position indicates whether the subspace contains the dimension or not. The islands are placed at the bottom of the list. Users can select subspaces by clicking. When they switch to ground-travel mode, the list shows all capital cities and cities passed through during the trip.

5.3 Map Detail View

The Map Detail View (Fig. 3c) displays the dimension and data patterns of subspaces. MDS projections show the data patterns of representative subspaces, with point opacity encoding data stability. Procrustes transformation [3] is used to avoid abrupt changes between projections, in order to preserve users' mental maps. A stability matrix is provided (Fig. 5) to help compare different data patterns. It shows all data items in a matrix in their original order. Users can always switch between the matrix and the projection. Icons on the boundary circle indicate featured dimensions, whose names are specifically displayed. At the national/provincial level, this view shows the data stability and featured dimensions of the cluster/sub-cluster members. At the municipal level, it only shows information about the chosen subspace. In ground-travel mode, it shows the origin, the current location, as well as the destination.

5.4 Exploration Workflow

All three views are highly coupled to support the subspace exploration workflow (Fig. 4). As a start, an overview is given showing multiple clusters of subspaces. Then users can drill down to different cluster levels and conduct their exploration. They can analyze the dimension and data patterns of each cluster/sub-cluster in the Subspace List View and the Map Detail View. It helps to reveal data patterns shared by most cluster members, as well as dominant dimension patterns that may account for it. Following the suggested routes, users can quickly browse the transition of patterns across different cluster regions. At the municipal level, they can further understand local subspace similarities by analyzing the glyph patterns.

6 Evaluation

In this section, we demonstrate the effectiveness of Subspace-Map with two real-world datasets. A comparison with the other four state-of-the-art approaches is also conducted.

6.1 Case 1: Forest Fires Data

The Forest Fires dataset [9] contains 517 instances collected from Montesinho Natural Park in Portugal. It has been used by Jäckle et al. [23] for case study. The park is divided into 72 regions. Each instance records the date and the region of a specific forest fire, along with 8 environmental monitoring factors: *Temperature (temp)*, *Relative Humidity (RH)*, *Wind speed (Wind)*, *Rainfall (Rain)*, *Fine Fuel Moisture Code (FFMC)*, *Duff Moisture Code (DMC)*, *Drought Code (DC)*, and *Initial Spread Index (ISI)*. These factors come from the Forest fire Weather Index (FWI), an index system widely used by domain scientists to estimate the risk of wildfire. By excluding all 1D subspaces, we have obtained altogether 247 subspaces.

Fig. 3 shows an overview. Two countries can be observed from the map, meaning that the subspaces form two major clusters. From Map Detail View (Fig. 3c), we can see that *Wind* is the dominating dimension at the national level. Most subspaces in country A include *Wind* while things are the opposite for country B. Comparing the two projections, we know that data points are more likely to form a high-density cluster in the subspaces of country B. Moreover, the projection of A has less high-opacity points, suggesting that data structures are less stable in country A.

Curious about the inner division of cluster A, we then move on to the provincial level to continue the analysis. The map shows that A can be divided into four provinces, i.e. 4 sub-clusters (Fig. 5). Among them, sub-cluster 4 is significantly larger than the other three. Seen from Map Detail View, dimensions *RH* and *temp* play a decisive role in the division of provinces. Specifically, *RH* contributes to most subspaces in sub-cluster 4 while it is commonly excluded by sub-cluster 1 to 3. *temp* along with *ISI* and *DMC* helps distinguish the 3 small sub-clusters. The projected data structures appear more clustered in sub-cluster 2 and 4 due to the existence of a few extreme instances. From the list view, we can see that *RH* and *Wind* are dominating sub-cluster 4 with the same amount of occurrences. The other dimensions are equally included with less contributions. It validates the insight previously provided by Map Detail View.

Before proceeding to the next step, we would like to probe into the semantics behind the above findings. In the FWI system, *temp*, *RH*, *Wind*, and *Rain* are four recorded natural factors used to generate the other four indicators. *FFMC*, *DMC*, and *DC* reflect water contents in different levels of surface coverings. *ISI* is derived from *FFMC* to indicate how fast a wildfire may spread. *RH* and *temp* directly impact the three water content indicators. It explains why they dominate the provincial-level clustering. As for *Wind*, its decisive role is simply due to its high independence to other dimensions. The climate of Portugal is characterized by hot and dry summers, cold and wet winters. *RH* and *temp* are strongly correlated. They vary with seasons but are similar across different regions. *Wind*, on the other hand, is more affected by topography but less related to seasons. It is simply the most informative dimension since it cannot be derived or approximated by other indicators.

Knowing the features of each sub-cluster, we can further explore how patterns change across these sub-clusters by traveling along the routes. We switch to the ground-travel mode and set the transition path to go over all provincial capitals (Fig. 6). Altogether 14 subspaces have been visited along the route. Subspace 1 and 2 are highly similar with scattered projections. Subspace 3 is an outlier with abrupt pattern changes. Subspace 4 to 8 belong to sub-cluster 2 and well present the gradual changes in data patterns. Subspace 9 to 11 of sub-cluster 3 do not look very consistent. Subspace 12 to 14 show how data pattern varies from scattered to clustered.

In order to find out why subspace 9 to 11 have different data patterns, we navigate to the municipal level of province 3 (Fig. 7). Fig. 7a shows the dimensions of all subspaces in a glyph style. However, it is difficult to visually find the trends. For each subspace, we highlight the common dimension patterns in its neighborhood, resulting in a better visual effect in Fig. 7b. Note that it does not change the design in Fig. 7a, but simply displays different dimensions with different opacity. Now we see that there exist several dimension patterns inside sub-cluster 3. Subspace 9, 10, and 11 exhibit 3 local patterns with different states of *ISI*, which explains the diversity in their data patterns. Highlighted in the figure are two most popular patterns. They come from split regions, suggesting that some neighborhoods are separated in the map. It is a flaw in the layout algorithm, which will be discussed in Sect. 7.

6.2 Case 2: Glass Identification Data

The Glass Identification dataset [13] contains 9 kinds of measurements for 214 glass samples. All glass samples have been divided into 6 classes: float-processed building windows, non-float-processed building windows, float-processed vehicle windows, containers, tableware, and headlamps. The measurements include *refractive index (RI)* and 8 types of oxide contents: *sodium (Na)*, *magnesium (Mg)*, *aluminum (Al)*, *silicon (Si)*, *potassium (K)*, *calcium (Ca)*, *barium (Ba)*, and *iron (Fe)*. The 9 factors generate altogether 502 subspaces with no less than

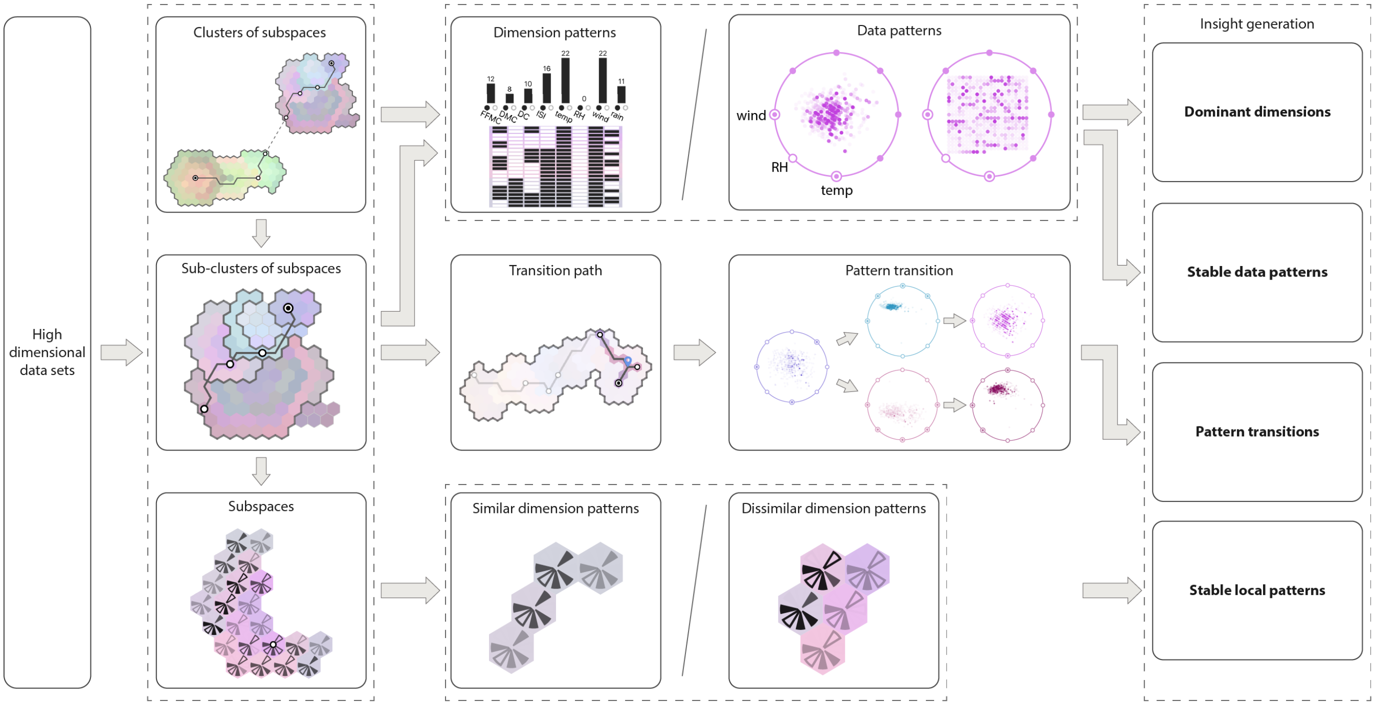

Fig. 4. Subspace-Map workflow. After constructing the map based on the input data, an overview showing the clusters of subspaces is provided. Users can hierarchically conduct exploration at cluster level, sub-cluster level, and subspace level. By analyzing various kinds of patterns and pattern transitions, they can gain insight into the dominant dimensions, stable data patterns, etc.

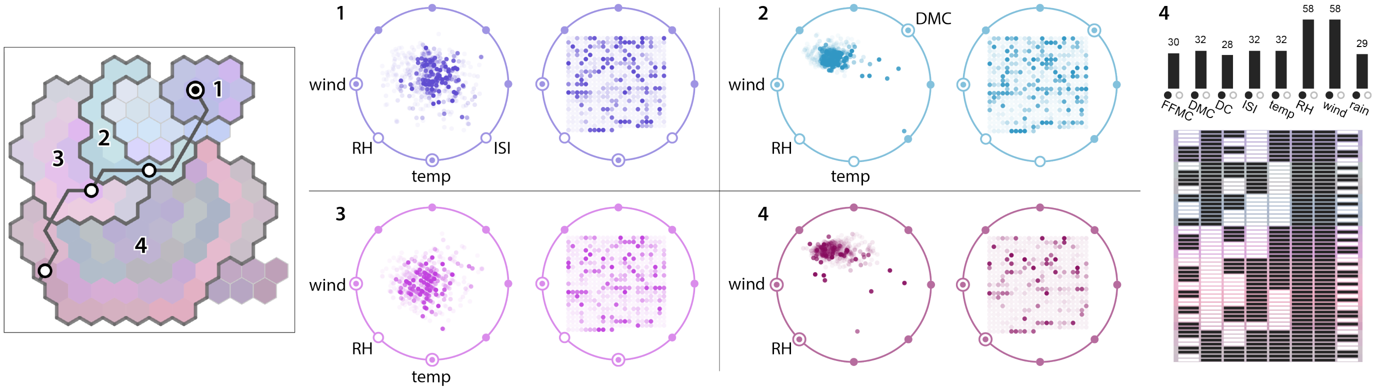

Fig. 5. Forest Fire Data: the analysis of cluster A. A can be divided into four sub-clusters. *RH* and *temp* are dominating the clustering at this level. *DMC* and *ISI* also plays a part in some sub-clusters. Judging from Subspace List View, *temp* does not stand out in sub-cluster 4.

2 dimensions. The map is formed by a randomly selected subset of 277 subspaces.

In the Map View, we can see that the subspaces are divided into two clusters. *Fe* is the unique featured dimension for both clusters. Most subspaces in cluster A involve iron while the opposite happens in cluster B. By comparing the stability matrices, we find that the featured stable data is similar between these two clusters. Judging from the projections, the featured data is more clustered in A.

At the provincial level, A is divided into 4 sub-clusters. All sub-clusters share one featured dimension: *Na*. Specifically, sub-cluster 1 to 3 exclude *Na* while sub-cluster 4 is characterized by involving it. The stability patterns are similar for sub-cluster 2 to 4, which focus on the lower half of the stability matrix. For sub-cluster 1, more stable data are involved in the upper part of the matrix. Judging from the projections, sub-cluster 1 to 3 share a highly similar data pattern featured by the linear alignment of data points. Sub-cluster 4 loses the linear feature and becomes more scattered.

Without considering *Fe*, the linear structure simply disappears in cluster B. In its 7 sub-clusters, 3 dimensions stand out to be the common featured dimensions: *Na*, *Al*, and *Si*. They are also featured dimensions in the sub-clusters of A. *RI* is also featured, but only appears in sub-cluster B-6 and A-1. Data patterns in B show a gradual change along the travel route passing through all provincial capitals. Specifically, more of the upper part data is involved as the stable structure, judging from the stability matrices. B-1 highlights only a few items, while B-7 highlights most of the data.

To comprehend the above observations, we seek help from the domain of glass manufacturing. It turns out that iron has a critical impact on the property of glass. The clarity of glass improves along with the decrease of iron oxide. Low-iron glass, also known as ultra-clear glass, is a specific type of glass very suitable for making optical and lighting equipment. Apparently, some glass samples in this dataset are low-iron glass while the others are standard glass. The two types are so close in the other measurements such that they can hardly be separated without considering iron. The discrimination effect of iron makes it highly informative. It explains why *Fe* dominates the national level clustering.

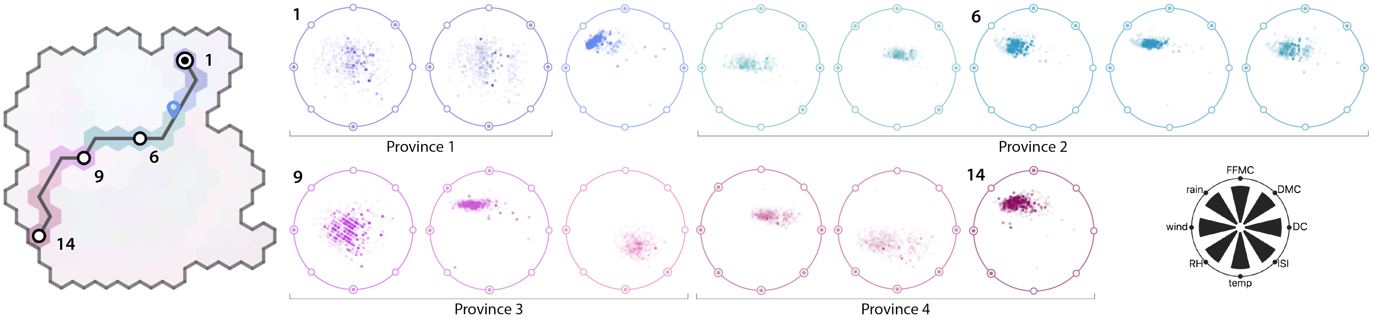

Fig. 6. Forest Fire Data: pattern transitions in cluster A. The travel route goes through 14 cities across 4 provinces, allowing users to browse the gradual change of patterns. It is worth noticing that subspace 9 to 11 present quite different data patterns.

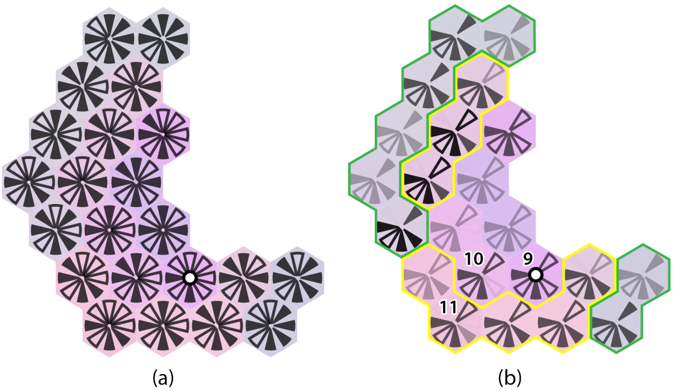

Fig. 7. Forest Fire Data: local patterns within sub-cluster A-3. (a) For each subspace, we show its dimensions in a glyph style. However, it is hard to find trends visually. (b) Therefore, we highlight shared dimension patterns in different neighborhoods for better perception. Subspace 9 to 11 shows different patterns, which explains their data diversity.

Then we focus on the featured data. It turns out the lower part of the stability matrix corresponds to three classes of glass: containers, tableware, and headlamps. Their samples exhibit a remarkably lower level of iron contents (with an average of 0.023) than the other classes (with an average of 0.068). An intuitive explanation is that glass for buildings and vehicle windows has lower clarity requirements than lamps and glassware. In fact, most samples of the 3 classes of low-iron glass contain 0 iron oxide. It explains the linear data structure highlighted in the sub-clusters of A.

As for *Na*, *Al*, and *Si*, we find that these factors help to distinguish between the three classes of low-iron glass. Specifically, Container glass is featured by low *Na*, high *Al*, and low *Si*. Tableware glass is featured by high *Na*, low *Al*, and low *Si*. All three elements are relatively high for headlamps. It explains why these attributes stand out at the provincial level, with the low-iron glass being the featured data.

6.3 Comparison with State-of-the-Art Methods

We provide a comparison between our proposed method and four relevant approaches: Dimension Projection Matrix/Tree [55], Subspace Search and Visualization [44], TripAdvisorND [31], and Pattern Trails [23]. The comparison is made from five perspectives, namely subspace search strategy, similarity measure, subspace grouping strategy, subspace layout, and pattern detection strategy (Table 1).

Subspace search strategy. Three approaches use or partially use subspace clustering algorithms to generate subspaces. To ensure the diversity and representativeness, we generate subspaces by controlling the sampling frequency of dimension combinations and dimensions. The sampling strategy is further discussed in Sect. 7.

Similarity measure. To make subspaces with different dimensionalities comparable, TripAdvisorND computes the Euclidean distance between the dimension vectors. Such dimension-based approach does not characterize the similarity well, since small changes in dimensions can cause huge differences in data patterns. On the other hand, it suffers from information loss due to projection. Pattern Trails is based on the projected distance of data, which is also affected by information loss. Subspace Search and Visualization computes resemblance in the data topology, which is not affected by dimensionality and does not cause information loss. We use the same method as it.

Subspace grouping strategy. Pattern Trails takes the union of dimensions within a cluster as the dimension of the cluster and compute the projection accordingly to represent the cluster. It is not reasonable because dimension distribution is not considered. Subspace Search and Visualization selects the representative subspace and computes the dimension distribution of the cluster. We provide richer information, including data stability, dimension stability, and featured dimensions of the cluster.

Subspace layout. Dimension Projection Matrix/Tree hierarchically organizes views in a tree layout, where nodes can be matrices representing both dimension and data aspects. It supports exploration in a dual and recursive manner. However, it hides the relationship between subspaces and is not very visually scalable. The other three methods use a projection layout. Although the relationship between subspaces can be conveyed, there may be severe visual occlusion. We use a easy-to-understand map layout. It displays the relationship relatively accurately and allocates the same screen space to each subspace, thus allowing embedding more additional information.

Pattern detection strategy. The first three methods support manual exploration only. Pattern Trails provides automatic pattern detection. It computes clusters for each subspace projection and identifies transition patterns based on cluster changes between two adjacent subspaces. This strategy is not scalable and, as seen in the example, can only identify a small number of existing patterns. We provide featured dimensions and data stability of clusters to help users quickly identify the impact of dimensions on data patterns. On the other hand, we present the dimension stability of individual subspaces to help users identify local patterns within clusters. Routes are also provided to trace pattern transitions in a gradual style.

7 Discussion

In this section, we discuss the current limitations and the possible extensions of Subspace Map.

Sampling strategy. We ensure the diversity and representativeness of the sampling results by controlling the sampling frequency of dimension combinations and dimensions. We do not use subspace clustering algorithms directly because they can only find subspaces with clusters. However, we do not consider data patterns, which risks missing important patterns in the data. We argue that important patterns possessed by a number of subspaces can be preserved to some extent due to the uniformity of our sampling guarantees. Later we can better solve this issue by introducing techniques for adaptively sampling subspaces.

Map layout. As shown in Fig. 7(b), the same local pattern appears in split regions. It suggests that subspaces with similar patterns in

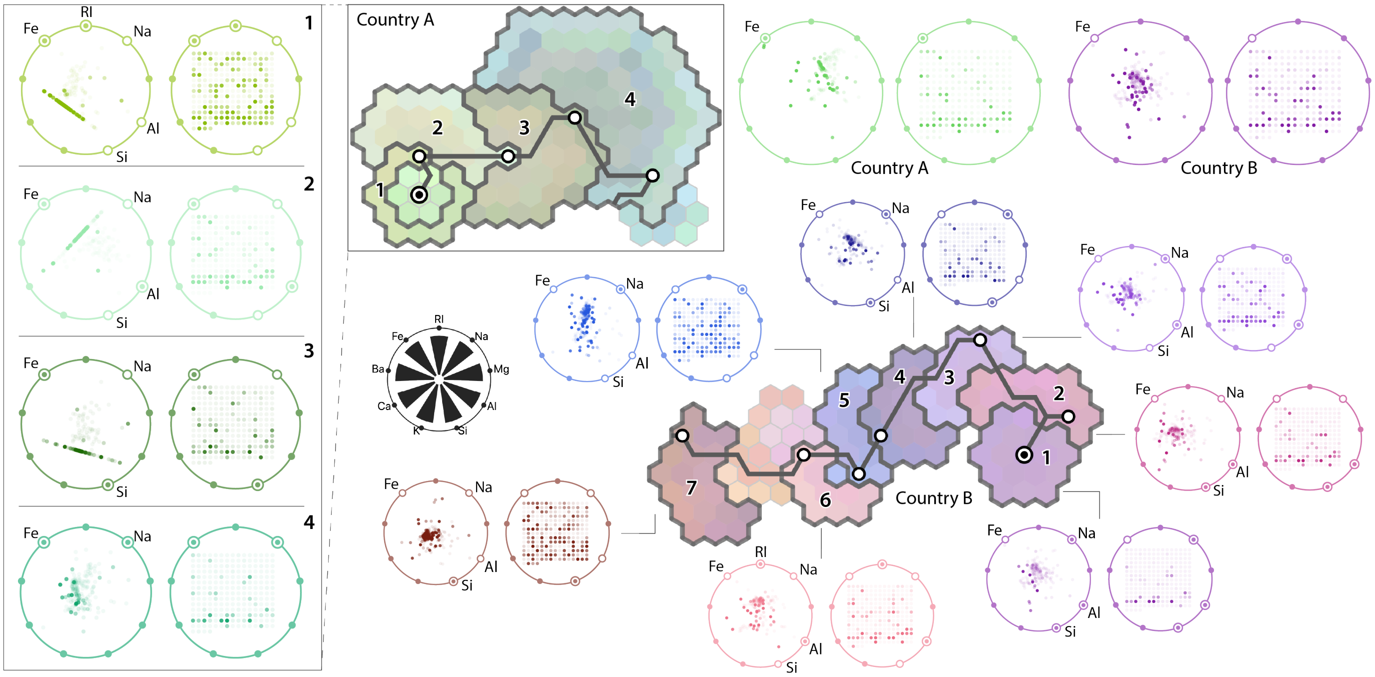

Fig. 8. Glass Identification Data: a comprehensive analysis. The subspaces form two clusters featuring *Fe*. Cluster A has 4 sub-clusters while B has 7 sub-clusters. *Na*, *Si*, and *Al* are the most featured dimensions at the provincial level. Most sub-clusters in A featured a linear data structure.

Table 1. Comparison of our work with four state-of-the-art approaches.

	Subspace search strategy	Similarity measure	Subspace grouping strategy	Subspace layout	Pattern detection strategy
Dimension Projection Matrix/Tree [55]	Manual selection	None	None	Tree and matrix	Observation and manual brushing
Subspace Search and Visualization [44]	Subspace clustering algorithm	Dimension overlap and data topology	Dimension distribution and representative subspace	2D projection	Observation and manual brushing
TripAdvisorND [31]	Subspace clustering algorithm and manual selection	Dimension vectors	None	2D projection	Observation and manual adjustment
Pattern Trails [23]	Subspace clustering algorithm	Projected distance of data	Dimension union of the cluster and subspace formed by the union	1D projection	Automatic detection based on clusters between adjacent subspaces
Ours	Self-design sampling method	Data topology	Representative subspace, data stability, and featured dimensions	Map layout	Automatic detection by 1) dimension and data features of subspaces and clusters and 2) pattern transition routes

the same sub-cluster may be separated due to flaws in the current layout algorithm. At the national and provincial level, this problem is avoided by separating the placement of different clusters and sub-clusters. Therefore, one possible solution is to construct lower level clusters before placing individual subspaces.

Color encoding. To show the similarity between subspaces through color, we project the subspace distance matrix to 3D so that each dimension represents a variable in the RGB color space. This is not an optimal solution because when the parameters in the RGB change, the color change is not so in line the human color perception. And without further constraints, some of the colors assigned to representative subspaces may be too light. We could project the distance matrix to 2D and pick colors from a designed 2D colormap based on the projected position of the subspace.

Scalability. The computational cost mainly lies in the dimensionality reduction algorithm and k-NN algorithm. Consider a dataset with n data items and d dimensions. For each subspace, the time complexity of the projection computation via MDS is $\mathscr{O}\left(n^3\right)$, and the k-NN list computation requires an additional $\mathscr{O}\left(d \cdot n\right)$ runtime. As dimensionality increases, the computational cost will soon become prohibitive. We can alleviate this problem in the following ways. First, we can reduce the number of subspaces to be computed by subspace sampling. Second, the computation for each subspace is mutually independent. A parallel approach can be used to speed up this process. k-NN computation can be further accelerated by approximation. Moreover, after one calculation, the results are saved to avoid repeated calculation. From a visual perspective, the scalability is influenced by the number of displayed subspaces. For optimization, we hierarchically organize the subspaces in the map and adaptively change the size of the hexagon according to their number. For the Map Detail View, the overlap in the scatterplot becomes severe when the number of data items is large. It can be solved by replacing the current representation with aggregation techniques such as the heatmap.

In our future work, we would like to integrate more analytical techniques. For example, we may provide multiple dimension reduction techniques. Users can choose the most suitable one based on their needs. We also plan to augment the guidance mechanism for automatically providing informative exploration directions or even insights to users.

8 Conclusion

We propose a novel approach called Subspace-Map to help users visualize and explore subspaces. It presents subspaces with various map metaphors, such as representing clusters as regions, showing representatives as capital cities, etc. Routes are also built, through which users are able to trace the transition of patterns between subspaces. We develop a prototype system and demonstrate its effectiveness through two case studies with real-world datasets. A comparison with state-of-the-art methods shows its advantages in different aspects. In the future, we will enhance the analytical functions of Subspace-Map and provide more guidance for subspace exploration.

References

[1] E. Bertini. Quality metrics in high-dimensional data visualization: An overview and systematization. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2203–2212, 2011.

[2] R. P. Biuk-Aghai, M. Yang, P. C. Pang, W. H. Ao, S. Fong, and Y. Si. A map-like visualisation method based on liquid modelling. *Journal of Visual Languages and Computing*, 31:87–103, 2015.

[3] I. Borg and P. Groenen. *Modern Multidimensional Scaling: Theory and Applications*. Springer, 2005.

[4] R. G. Cano, K. Buchin, T. Castermans, A. Pieterse, W. Sonke, and B. Speckmann. Mosaic drawings and cartograms. *Computer Graphics Forum*, 34(3):361–370, 2015.

[5] N. Cao, Y. Lin, and D. Gotz. Untangle map: Visual analysis of probabilistic multi-label data. *IEEE Transactions on Visualization and Computer Graphics*, 22(2):1149–1163, 2016.

[6] D. B. Carr, R. J. Littlefield, W. Nicholson, and J. Littlefield. Scatterplot matrix techniques for large n. *Journal of the American Statistical Association*, 82(398):424–436, 1987.

[7] S. Chen, S. Chen, Z. Wang, J. Liang, X. Yuan, N. Cao, and Y. Wu. D-map: Visual analysis of ego-centric information diffusion patterns in social media. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 41–50, 2016.

[8] S. Chen, S. Li, S. Chen, and X. Yuan. R-map: A map metaphor for visualizing information reposting process in social media. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1204–1214, 2020.

[9] P. Cortez and A. Morais. A data mining approach to predict forest fires using meteorological data. In *Proceedings of the 13th Portuguese Conference on Artificial Intelligence (EPIA 2007)*, pages 512–523, 2007.

[10] H. Couclelis. Worlds of information: The geographic metaphor in the visualization of complex information. *Cartography and Geographic Information Systems*, 25(4):209–220, 1998.

[11] T. F. Cox and M. A. Cox. *Multidimensional Scaling*. Chapman and Hall/CRC, 2000.

[12] A. Dasgupta and R. Kosara. Pargnostics: Screen-space metrics for parallel coordinates. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1017–1026, 2010.

[13] D. Dua and C. Graff. UCI machine learning repository, 2017.

[14] N. Elmqvist, P. Dragicevic, and J. Fekete. Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1539–1148, 2008.

[15] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[16] E. R. Gansner, Y. Hu, and S. G. Kobourov. Gmap: Visualizing graphs and clusters as maps. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 201–208, 2010.

[17] B. Grünbaum and G. C. Shephard. Tilings by regular polygons. *Mathematics Magazine*, 50(5):227–247, 1977.

[18] B. Grünbaum and G. C. Shephard. *Tilings and patterns*. Courier Dover Publications, 1987.

[19] D. Guo. Coordinating computational and visual approaches for interactive feature selection and multivariate clustering. *Information Visualization*, 2(4):232–246, 2003.

[20] M. Hogräfer, M. Heitzler, and H. Schulz. The state of the art in map-like visualization. *Computer Graphics Forum*, 39(3):647–674, 2020.

[21] P. J. Huber. Projection pursuit. *The Annals of Statistics*, pages 435–475, 1985.

[22] A. Inselberg. The plane with parallel coordinates. *The Visual Computer*, 1(2):69–91, 1985.

[23] D. Jäckle, M. Hund, M. Behrisch, D. A. Keim, and T. Schreck. Pattern trails: Visual analysis of pattern transitions in subspaces. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 1–12, 2017.

[24] J. Johansson and M. D. Cooper. A screen space quality method for data abstraction. *Computer Graphics Forum*, 27(3):1039–1046, 2008.

[25] S. Johansson and J. Johansson. Interactive dimensionality reduction through user-defined combinations of quality metrics. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):993–1000, 2009.

[26] H. Kriegel, P. Kröger, and A. Zimek. Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(1):1:1–1:58, 2009.

[27] P. Liu, D. Zhou, and N. Wu. Vdbscan: Varied density based spatial clustering of applications with noise. In *Proceedings of 2007 International Conference on Service Systems and Service Management*, pages 1–4, 2007.

[28] C. Ma, Y. Liu, G. Zhao, and H. Wang. Visualizing and analyzing video content with interactive scalable maps. *IEEE Transactions on Multimedia*, 18(11):2171–2183, 2016.

[29] D. Mashima, S. G. Kobourov, and Y. Hu. Visualizing dynamic data with maps. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1424–1437, 2012.

[30] E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment*, 2(1):1270–1281, 2009.

[31] J. E. Nam and K. Mueller. Tripadvisor$^{\mathrm{n\text{-}d}}$: A tourism-inspired high-dimensional space exploration framework with overview and detail. *IEEE Transactions on Visualization and Computer Graphics*, 19(2):291–305, 2013.

[32] K. V. Nesbitt. Getting to more abstract places using the metro map metaphor. In *Proceedings of the 8th International Conference on Information Visualization*, pages 488–493. IEEE Computer Society, 2004.

[33] D. T. Nhon and L. Wilkinson. Scagexplorer: Exploring scatterplots by their scagnostics. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 73–80, 2014.

[34] W. Peng, M. O. Ward, and E. A. Rundensteiner. Clutter reduction in multi-dimensional data visualization using dimension reordering. In *Proceedings of IEEE Symposium on Information Visualization*, pages 89–96, 2004.

[35] R. Preiner, J. Schmidt, K. Krösl, T. Schreck, and G. Mistelbauer. Augmenting node-link diagrams with topographic attribute maps. *Computer Graphics Forum*, 39(3):369–381, 2020.

[36] A. Sarvghad, M. Tory, and N. Mahyar. Visualizing dimension coverage to support exploratory analysis. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):21–30, 2017.

[37] J. Seo and B. Shneiderman. A rank-by-feature framework for interactive exploration of multidimensional data. *Proceedings of IEEE Symposium on Information Visualization*, 4(2):96–113, 2005.

[38] M. Sips, B. Neubert, J. P. Lewis, and P. Hanrahan. Selecting good views of high-dimensional data using class consistency. *Computer Graphics Forum*, 28(3):831–838, 2009.

[39] A. Skupin. From metaphor to method: Cartographic perspectives on information visualization. In *Proceedings of IEEE Symposium on Information Visualization*, pages 91–97, 2000.

[40] A. Skupin and S. I. Fabrikant. Spatialization methods: A cartographic research agenda for non-geographic information visualization. *Cartography and Geographic Information Science*, 30(2):99–119, 2003.

[41] E. S. Spelke. Principles of object perception. *Cognitive science*, 14(1):29–56, 1990.

[42] J. Stahnke, M. Dörk, B. Müller, and A. Thom. Probing projections: Interaction techniques for interpreting arrangements and errors of dimensionality reductions. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):629–638, 2016.

[43] A. Tatu, G. Albuquerque, M. Eisemann, J. Schneidewind, H. Theisel, M. A. Magnor, and D. A. Keim. Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 59–66, 2009.

[44] A. Tatu, F. Maass, I. Färber, E. Bertini, T. Schreck, T. Seidl, and D. A. Keim. Subspace search and visualization to make sense of alternative clusterings in high-dimensional data. In *Proceedings of IEEE Conference on Visual Analytics Science and Technology*, pages 63–72, 2012.

[45] C. Turkay, P. Filzmoser, and H. Hauser. Brushing dimensions - A dual visual analysis model for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2591–2599, 2011.

[46] C. Turkay, A. Lundervold, A. J. Lundervold, and H. Hauser. Representative factor generation for the interactive visual analysis of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2621–2630, 2012.

[47] K. Watanabe, H. Wu, Y. Niibe, S. Takahashi, and I. Fujishiro. Biclustering

multivariate data for correlated subspace mining. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 287–294, 2015.

[48] J. E. Wenskovitch, I. Crandell, N. Ramakrishnan, L. House, S. Leman, and C. North. Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 24(1):131–141, 2018.

[49] L. Wilkinson, A. Anand, and R. L. Grossman. Graph-theoretic scagnostics. In *Proceedings of IEEE Symposium on Information Visualization*, page 21, 2005.

[50] L. Wilkinson, A. Anand, and R. L. Grossman. High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics*, 12(6):1363–1372, 2006.

[51] S. Wold, K. Esbensen, and P. Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.

[52] K. Wongsuphasawat, D. Moritz, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2016.

[53] K. Wongsuphasawat, Z. Qu, D. Moritz, R. Chang, F. Ouk, A. Anand, J. D. Mackinlay, B. Howe, and J. Heer. Voyager 2: Augmenting visual analysis with partial view specifications. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 2648–2659, 2017.

[54] J. Yang, W. Peng, M. O. Ward, and E. A. Rundensteiner. Interactive hierarchical dimension ordering, spacing and filtering for exploration of high dimensional datasets. In *Proceedings of IEEE Symposium on Information Visualization*, pages 105–112, 2003.

[55] X. Yuan, D. Ren, Z. Wang, and C. Guo. Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2625–2633, 2013.

[56] Z. Zhang, K. T. McDonnell, and K. Mueller. A network-based interface for the exploration of high-dimensional data spaces. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 17–24, 2012.

[57] Z. Zhang, K. T. McDonnell, E. Zadok, and K. Mueller. Visual correlation analysis of numerical and categorical data on the correlation map. *IEEE Transactions on Visualization and Computer Graphics*, 21(2):289–303, 2015.