J Vis (2019) 22:1225–1240
https://doi.org/10.1007/s12650-019-00599-w

REGULAR PAPER

Guodao Sun · Zhixiu Zhou · Baofeng Chang · Jingwei Tang · Ronghua Liang

PermVizor: visual analysis of multivariate permutations

Received: 7 July 2019 / Accepted: 5 August 2019 / Published online: 17 September 2019

Abstract Permutation exists in various domains such as mathematics, combinatorics, and computer science. Enumerating each permutation, as well as the multivariate information among different items, allows us, for example, to observe distribution, similarity, and dissimilarity of all possible permutations and select a satisfactory permutation or solution. However, the number of permutations increases dramatically along with the number of items in the permutation, which makes it challenging for users to evaluate potential solutions and identify interesting insights. In this paper, we propose PermVizor, a novel and scalable visualization system that aims assisting users exploring the arrangement, distribution, and comparison of permutations. Necessary and comprehensive analysis of requirements is presented for visualization of permutations. PermVizor enables users to explore overall distribution of each permutation with a glyph-based MDS view, investigate statistical information of selected permutations with a parallel coordinates view, and examine detailed arrangement of the items as well the multivariate information among them for each permutation with pixel-based and block-based PermView. Case studies are conducted on classical datasets such as the axis reordering issue in parallel coordinate data and permutation of traveling salesman problem, which shows that PermVizor could facilitate users in exploring unexpected and desired permutations and confirm their finding and decisions in expected permutations.

Keywords Permutation · Multi-dimensional data · Parallel coordinate

1 Introduction

Permutation is not only a basic topic in mathematical combinatorics, but also appears in many other fundamental and applied disciplines such as mathematics, combinatorics, computer science, operations

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s12650-019-00599-w) contains supplementary material, which is available to authorized users.

G. Sun · Z. Zhou · B. Chang · J. Tang · R. Liang (✉)
Zhejiang University of Technology, Hangzhou, China
E-mail: rhliang@zjut.edu.cn

G. Sun
E-mail: godoor.sun@gmail.com

Z. Zhou
E-mail: 2568516551@qq.com

B. Chang
E-mail: baofeng.chang@foxmail.com

J. Tang
E-mail: eejwtang@foxmail.com

research, computational molecular biology (Paterson et al. 2011), natural languages processing (Munakata and Barták 2006), or even color schemes generation (Tennekes and De Jonge 2014). Permutation is defined as the task of arranging all the items of a set into non-repeating sequences. Here, a set is defined as a collection of distinct items. Different sequences may present different implications for the sequential order of the items they present.

Understanding the meanings and obtaining the insights behind each permutation presents many useful applications and leads solution to many scenarios. For instance, the parallel coordinates in visualization area always suffer from the pain of reordering axes for pattern optimization, which appears to be a NP-complete problem. The same issue also existed in classical traveling salesman problem or Sugiyama-style graph drawing (Sugiyama 2002). Many researchers have investigated sophisticated methods, a variety of automatic axes reordering methods to discover better visual structures and patterns (Blaas et al. 2008; Lu et al. 2012; Sun et al. 2013). However, these works either focus on optimizing specific mathematics indicators during pattern recognition with parallel coordinates (Lu et al. 2012) (i.e., mean crossing angles) or try to reduce visual clutter for better visualization (i.e., reduce the amount of overlapped polylines between the neighboring axes). The reasoning and explaining process of such above algorithms is not investigated.

However, unfolding such permutation process is hindered by the following major challenges. Firstly, the number of permutations increases dramatically along with the increment of the number of the items in a permutation. Directly rendering all the permutations would instantly consume the display space. How to solve the problem of displaying the information of all sequences in the system interface is an important challenge for this task

Secondly, the connection between items in a permutation may contain multivariate information such as the correlation coefficient of two axes. For example, the traveling salesman problem is a typical permutation problem of examining all possible cities; however, multimodal transportation may exist for each permutation. Thus, the permutation visualization should not only focus on the sequence or order of each items, but should also present detailed multivariate connection information between each item. Thus, an assistant tool for both displaying order and connection information of permutations is necessary to help users in analyzing multivariate permutations. How to effectively encode and display the multiple attributes of the sequence and facilitate the user to understand the sequence from the visual perspective remains challenging. Finally, how to measure the similarity between multivariate sequences and distribute them effectively to facilitate finding, comparing and analyzing patterns also faces challenges.

We propose a new visual analysis system that can assist analysts in exploring and analyzing the arrangement, distribution, and comparison of permutations. Eight core and comprehensive analysis of requirements are first identified for analysis and visualization of permutations. We then propose a new and scalable method to measure the similarity between multivariate sequences. In view of the characteristics of sequence data, Kendall distance and Euclidean distance are combined to measure the similarity among sequences. It takes into account the order information and connection properties of multivariate sequences. On this basis, we propose a visualization with multiple coordinate views that include glyph-based MDS view, parallel coordinates view, pixel-based PermView, and block-based PermView. The glyph-based MDS view enables users to explore overall distribution of permutations. The nodes on MDS view is clustered to form cluster glyph to better present the statistical distribution of each sequence's connection attributes. PCP view is used to present statistical information of selected permutation, and that can help users narrow down the search space for optimal permutations. Detailed arrangement of the items as well the multivariate information among them for each permutation is finally presented with pixel-based and block-based PermView. To solve the scalability problem and express more information in limited display space of the screen, the idea of focus + context technique is employed. The block-based visualization is to showcase the finely grained permutations while the pixel-based visualization is to present the aligned or reordered background permutations. Case studies are conducted on classical datasets such as the axis reordering issue in parallel coordinate data and permutation of traveling salesman problem to evaluate the usefulness and effectiveness of the proposed system.

The key contributions of this work are as follows:

- We identify core analysis requirements for understanding permutations and present a new model to measure similarity between multivariate sequences.
- We propose PermVizor, a novel and scalable visualization technique aiming at exploring the arrangement, distribution, and comparison of permutations.
- We provide empirical findings based on two real all permutation datasets to test the effectiveness of our similarity model and the visual analysis system.

2 Related works

2.1 Permutation and combination

In almost every area of mathematics, permutations occur in more or less prominent ways. The concept of permutation involves the act of arranging all the members of a set into a certain sequence or order, or if the set has been sorted, rearranging (reordering) its elements. This process is called permutation. Permutation is different from the combination, and the combination is a subset of the full arrangement, ignoring certain sequences in the full arrangement as required. For example, the set 1,2,3 has six permutations, namely (1,2,3), (1,3,2), (2,1,3), (2,3,1), (3,1,2), and (3,2,1). These are all the possible sorts of these three element sets. When considering different orderings of certain finite sets, permutations usually arise probably because people want to know how many sorting schemes are created. The number of *n* different objects arranged is the factorial of *n*, usually written as *n!* Similarly, the ranking algorithms often appear in the study of sorting algorithms in computer science. In algebra field, especially graph theory, a permutation of a set *S* is defined as a bidirectional map from *S* to itself (McCoy 1968; Stoll 2013; Mycielski et al. 1997). In other words, it is a function from *S* to *S*, and each element appears only once as a value. The process is related to the rearrangement of elements in *S*, where each element is replaced by the corresponding *f*(*S*).

Permutation and combination are important branches of many fields such as engineering (e.g.,image analysis and communication networks), computer science (e.g., natural languages process and intelligent computing), natural and social sciences, bio-medicine [e.g., molecular biology (Liu 1968; Roberts and Tesman 2009; Apostolico and Hein 2005]. Operational research (e.g., transportation and scheduling) and commercial services also have a wide range of applications. The most common situation is that many real-world problems are mathematically difficult to solve. In these situations, it is necessary to combine mathematical techniques to calculate, enumerate, or even present possible solutions to solve application problems. The generation of permutations such as the process of arrangement and combination has been extensively studied due to their basic nature and importance in practical applications. It is commonly seen in communication networks, parallel and distributed systems, for example ranking different routes for performance evaluation is a common problem in these areas (Massini 2003; Yang and Wang 2004). Moreover, many communication networks require the secure transmission of information, which promotes the development of cryptography and network security. One core part of above encryption process involves the operation of code sequences such as numbers, characters, and words (Stallings 2003).

2.2 Permutation ordering, selection, and visualization

Ordering and selecting certain permutation exist in many areas, and scientists try to solve them with various optimization algorithms. For example, the traveling salesman problem is a typical all permutation problem. As the scale of the problem increases, scientists employ different algorithms such as genetic algorithms, simulated annealing, ant colony algorithms, tabu search algorithms, greedy algorithms, and neural networks to solve the problems. The traveling salesman problem is a large-scale NP problem. In the field of visualization, there also exist permutation problems and visualization layout which often require ordering of their components (Hurley and Oldford 2010), such as parallel coordinate rearrangement and matrix reordering.

Automatic matrix reordering methods have been extensively studied in three domains, namely statistics, linear algebra, and graph theory. Liiv (2010) provides an extensive review of seriation (another term for matrix reordering). According to this review, “seriation is an exploratory combinatorial data analysis technique to reorder objects into a sequence along a one-dimensional continuum so that it can reveal regularity and patterning among the whole series.” However, seriation only focuses on finding one order and leaves operations like sequence grouping to the users to interpret.

According to the above theory, Kim et al. (2007) propose four reordering methods for sets, namely ordering based on HAC (hierarchical agglomerative clustering) (Eisen et al. 1998), moving rows to the top, ordering by name, and ordering by cardinality. These reordering methods would carry out a permutation matrix and provide users with a succinct and clear overview of sets and elements. Perin et al. (2014) present Bertifier, a web app for rapidly creating tabular visualizations from spreadsheets. It introduces visual reordering, a semi-interactive reordering approach that makes it possible for users to apply and tune automatic reordering algorithms in a WYSIWYG manner. Pilhofer et al. (2012) present an efficient algorithm based on Bertin’s idea (Bertin 1981) and Kendall’s tau’s concepts (Kendall 1938) to optimize the ordering work. It demonstrates the idea that data should be organized to reflect the effect and result you want to observe. To be more concrete, they consider an ordering to be good one if it makes the identification of structural information as well as separable groups (or clusters) easier and brings similar categories together (Friendly and Kwan 2003).

Parallel coordinates plot projects high-dimensional datasets onto a series of parallel axes on the plane. In the process of observing the parallel coordinates, researchers find that the patterns represented by two adjacent axes of parallel coordinates of the dataset can be observed easily. However, the visibility and pattern of information expressed by non-adjacent axes is greatly reduced. Therefore, how to determine the mapping relationship between axes and dimensions, that is, the order of dimensions, has an important influence on the visualization and insight obtaining of parallel coordinates (Johansson and Forsell 2016). Blaas et al. (2008) discussed that how to handle reordering is to pre-compute the bin maps for all possible axis permutations. Heinrich et al. (2012) use a graph-theoretic approach to determine a list of axis orderings such that all pairwise relations can be displayed without redundancy, while each parallel coordinates plot can be used independently to visualize all variables of the dataset. Lu et al. (2016) employ a similarity-based reordering method based on the combination of nonlinear correlation coefficient (NCC) and SVD algorithms, to obtain meaningful distribution.

In most cases, the goal of permutation problems is to find the best result through mathematical model optimization algorithm. However, in different areas, different people have different requirements for different problems, which makes the process more complicated. In the meanwhile, we also need to understand the process of producing the best solution and find similar solutions to the best solutions. In this paper, We try to propose PermVizor, a visual analysis system aiming at exploring the processing of searching the best sequence from permutations, comparing differences between sequences and finding meaningful patterns from permutations to help users in their decision making.

3 Analytical tasks

We identified the following requirements based on research from permutation and combinations literature above, as well as typical permutation problems. We also conduct interview with experts from computer science/engineering background that are not co-authors of this paper to collect their feedback to refine the requirements and analysis tasks.

R1 *Encode permutations.* The visualization should help users recognize the arrangement of the items of each permutation, as well as the overall distribution of all permutations.

R2 *Support multivariate permutations.* The connection between items in a permutation contains multivariate information. For example, the traveling salesman problem is a typical permutation problem of examining all possible cities; however, multimodal transportation may exist for each permutation. Thus, the permutation visualization should not only focus on the sequence or order of each items, but should also present detailed multivariate connection information between each item.

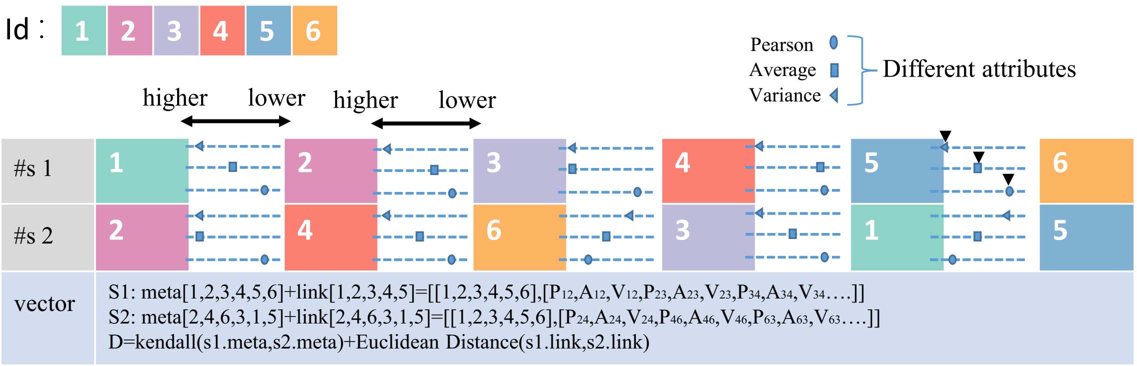
R3 *Support permutations aggregation.* The number of permutations of n items is the factorial of the integer n, which increases much faster than polynomials and exponential functions. Aggregation of all permutations helps users quickly grasp the similarity and dissimilarity among the permutations and provides a hint for subsequent exploration.

R4 *Scalability support.* Since the number of permutations increases dramatically along with the increment of the value n, there should be a trade-off between delivering the overview distribution of permutations and detailed information (e.g., the connection between items) of partial permutations. Different permutation data may lead to different analysis tasks, and the value n may range from single digits to dozens. Thus, the permutation visualization technique should be designed to optimize the trade-off and satisfy different specific tasks.

R5 *Statistical information support* Different permutations may lead to totally different orders as well as the consequent statistical information. Providing an intuitive visualization of the statistical information is necessary for users to understand the characteristic of each sequence.

R6 *Support permutation comparison* Different permutations have different orders and different values of attributes linking each item. Presenting the similarity and dissimilarity among permutations is crucial for users to distinguish and discover the desired sequence or patterns.

R7 *Support alignment* The sequence should be aligned at a specific item to better illustrate the medial patterns among each permutation.



S1: meta[1,2,3,4,5,6]+link[1,2,3,4,5]=[[1,2,3,4,5,6],[P_{12},A_{12},V_{12},P_{23},A_{23},V_{23},P_{34},A_{34},V_{34}….]]
S2: meta[2,4,6,3,1,5]+link[2,4,6,3,1,5]=[[1,2,3,4,5,6],[P_{24},A_{24},V_{24},P_{46},A_{46},V_{46},P_{63},A_{63},V_{63}….]]
D=kendall(s1.meta,s2.meta)+Euclidean Distance(s1.link,s2.link)

Fig. 1 Process of computing the distance between two permutations. Each permutation is denoted as a feature vector, and the feature vector is constructed with two parts, i.e., the meta-attributes and link attributes

R8 *Support filter and zoom* The system should provide functions of filtering and avoid specific patterns.

These requirements helped us derive appropriate design principles and make judicious decisions on our visual design. The following contents of this paper will introduce the design of each component, visual mapping, and its interaction in detail.

4 Visualization techniques

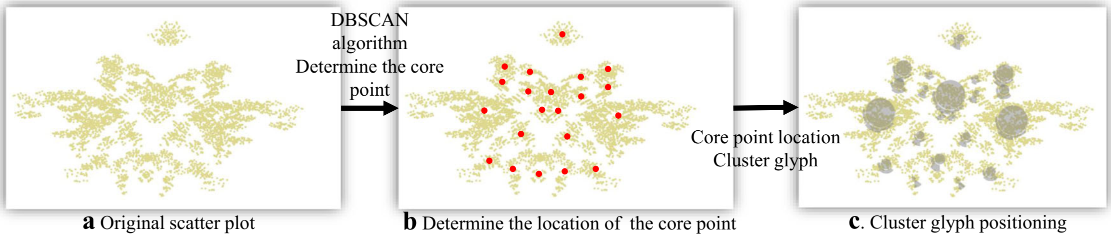
4.1 MDS view

**We proposed a glyph-based MDS view to demonstrate the overall distribution of all permutations. In the MDS view, each node represents a permutation. The closer the nodes on MDS view are, the more similar the two permutations represented by nodes are. MDS view adopts the commonly used MDS (multi-dimensional scaling) to evaluate the difference between multi-dimensional vectors. The MDS first computes the relative distance between two feature vectors and projects the nodes representing the feature vectors onto two-dimensional space. In our scenario, each permutation is denoted as a feature vector. The feature vector is constructed with two parts, i.e., the meta-attributes and link attributes. The meta-attributes indicate the own attributes of each item in each permutation, for example the order of an item in a permutation. The link attributes mean the connection attributes of neighboring items, for example the traveling distance or cost between two cities in traveling salesman problem and the Pearson correlation coefficient between two vectors in parallel coordinates. The above two parts are integrated together to create a complete feature vector that can reflect characteristic behind a permutation.

The distance between two feature vectors is calculated as follows. We use Kendall tau distance to measure the order disparity in the meta-attributes between two permutations. The Kendall's tau distance is frequently used to calculate the similarity distance of permutation sequences. As shown in Fig. 1: All the binary constraint sets of sequence S_1 are as follows:

$$
\begin{aligned}
&(1,2),(1,3),(1,4),(\mathbf{1},\mathbf{5}),(1,6),\\
&(\mathbf{2},\mathbf{3}),(\mathbf{2},\mathbf{4}),(\mathbf{2},\mathbf{5}),(\mathbf{2},\mathbf{6}),\\
&(3,4),(\mathbf{3},\mathbf{5}),(3,6),\\
&(\mathbf{4},\mathbf{5}),(\mathbf{4},\mathbf{6}),\\
&(5,6)
\end{aligned}
$$

Regarding sequence S_2, all the binary constraint sets are as follows:



a Original scatter plot **b** Determine the location of the core point **c**. Cluster glyph positioning

Fig. 2 Demonstration of the process of cluster glyph position determination. DBSCAN is first executed to determine the number of clusters and the core points of the clusters, and cluster glyph is further added to the MDS view to map the statistical properties of the cluster at the core point location

$$\begin{array}{l}
(\mathbf{2},\mathbf{4}),(\mathbf{2},\mathbf{6}),(\mathbf{2},\mathbf{3}),(2,1),(\mathbf{2},\mathbf{5}),\\
(\mathbf{4},\mathbf{6}),(4,3),(4,1),(\mathbf{4},\mathbf{5}),\\
(6,3),(6,1),(6,5),\\
(3,1),(\mathbf{3},\mathbf{5}),\\
(\mathbf{1},\mathbf{5})
\end{array}$$

The number of the paired divergences of the above two ordered pair sequences S_1 and S_2 is counted as $d_\Delta = (S_1, S_2)$. (The same ordered pairs are in bold in above examples, which indicates the number of the different ordered pairs are 7.)

Thus, we can obtain the Kendall's tau coefficient as follows:

$$\tau = 1 - \frac{2 \times [d_\Delta(S_1, S_2)]}{N(N-1)} = 1 - \frac{2 \times 7}{6(6-1)} = 0.53 \tag{1}$$

We use Formula (1) to calculate the similarity of two permutations. The closer the τ to zero, the less similar the permutations are. We further adopt classical Euclidean distance to calculate the distance of the connection attributes of two permutations. All the attribute sets are normalized. Here, we take the time attributes normalization as an example. The time set $T = \left\{T_{12}, T_{13}, \ldots T_{ij} \ldots T_{N(N-1)}\right\}$ is mapped to the interval of normal distribution $N(0,1)$ as:

$$T_{ij}^{*} = \frac{T_{ij} - T_{\min}}{T_{\max} - T_{\min}} \tag{2}$$

To standardize all the data attribute, we use Euclidean distance to calculate the connection attributes of sequence, as shown in Fig. 1. Finally, the Kendall's tau coefficient and the normalized Euclidean distance will be added as the quantitative similarity of the two sequences schemes. The similarity distance algorithm between the two schemes is determined, and the formula of the dissimilarity matrix is constructed as follows:

$$D = \begin{pmatrix}
0 & & & & & & \\
D_{2,1} & 0 & & & & & \\
D_{3,1} & D_{3,2} & 0 & & & & \\
D_{4,1} & D_{4,2} & D_{4,3} & 0 & & & \\
. & & & & & & \\
. & & & & & & \\
\ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots
\end{pmatrix} \tag{3}$$

The pairwise distance measurement among each permutation is fed into MDS algorithm to generate the MDS distribution. Users are also allowed to adjust the weight for meta-attributes and link attributes to obtain different MDS distribution.

In the MDS view, we map all the results of the permutations to the two-dimensional space. As the number of items increases, the number of permutations sets increases exponentially. Displaying them all in browser page will cause the browser to crash easily. Therefore, when the number of items exceeds eight, that is, the number of permutation sets exceeds 40,000, we selectively show the points in the two-dimensional space. We first use DBSCAN algorithm to determine the number of clusters and core points based on density. Regarding the points in each cluster, we integrate the overlapped points and the ones whose distance from the core point of the cluster is less than a certain threshold value together to reduce the number of rendered points. Although the number of points in the MDS view has decreased, the information about the reduced points remains in the MDS space. In this way, our MDS view is capable of showing more than 40,000 permutation sets (R4) and help relax the scalability problem of full permutation.

Through the above MDS algorithm projection and DBSCAN clustering, we can visually and clearly discover the clustering pattern of the sequence through our MDS view (R3). Although we can find clusters and outliers in the MDS view, due to the reality of large data scatter plots and limited display space , users can hardly get more information at the beginning to help them understand clusters and find interesting ones. Therefore, as shown in Fig. 2c, we further design and propose a projected view with glyph added to help users identify the differences in cluster objects and help users understand the clusters. Our cluster glyph design is based on the DBSCAN clustering algorithm, which maps the statistical properties of the cluster at the core point location. Our core point is to determine the central location of each cluster through the DBSCAN algorithm. The cluster glyph design at the core point consists of two parts: the inner and outer rings (see Fig. 3).

As shown in Fig. 3, the size of the inner circle depicts the number of points in a cluster. The larger the inner circle, the greater the number of points in the cluster. The outer sector represents the statistical distribution of each sequence's connection attribute in the cluster (R5). Thus, users can quickly identify all sequence cluster patterns based on their meta-attributes and link attributes through the MDS view in a two-dimensional plane. The glyph shape can help discover special clusters or clusters of interest. For example, in the parallel coordinate dataset, as shown in Fig. 3, the fan shape in the system is distributed clockwise from the 12 o'clock position of the outer circle, and the fan shape in the clockwise direction indicates the correlation level of the sequence. As a result, the items are connected in pairs and contain three connection attribute values. In comparison with the slope of the connection line and the variance of the slope, in the parallel coordinate rearrangement, the user pays more attention to the Pearson correlation coefficient between the item connections. We set the threshold and set the rank of each cluster in accordance with the Pearson correlation coefficient of its connection. For example, in the sequence consisting of seven items, if the Pearson correlation coefficient in the four connection attributes is greater than the threshold, then the correlation level is four (see Fig. 3). If only two connections have a Pearson coefficient greater than the threshold, then the correlation level is two.

We map the correlation level of each sequence in the cluster clockwise. In the clockwise direction, the correlation level of the fan is increased. The length of the fan indicates the number of sequences of the corresponding level. In addition, user can customize the cluster glyph, select the attribute value that the user is interested in, and set the threshold according to his/her prior knowledge. The default threshold value of the system is the median value of the corresponding attribute value in the dataset. We design the cluster glyph to help users quickly understand the characteristics of each cluster in the MDS view according to their needs and guide users to further explore the selection of interest areas.

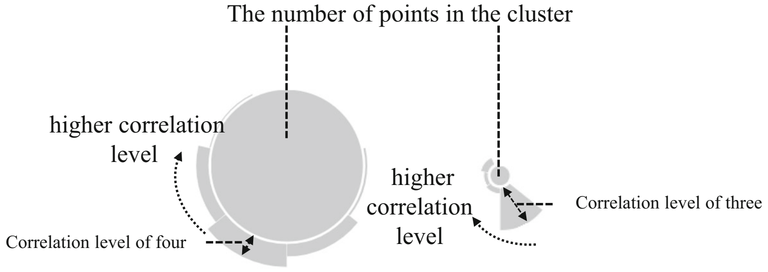

The number of points in the cluster
higher correlation level
Correlation level of four
higher correlation level
Correlation level of three

Fig. 3 Our proposed glyph in MDS view: The size of inner circle encodes the number of sequences in a cluster, and the outer sector radii encode the correlation level, namely the number of correlation between two dimensions that are larger than a predefined threshold

4.2 Parallel coordinates visualization

MDS view is useful to deliver the overall distribution of similarity and dissimilarity of all permutations. However, we still need to explore the detailed information of selected clusters or abnormal points in MDS view. Therefore, we provide a parallel coordinates visualization to present the statistical information of selected permutations (R5). In this view, each polyline corresponds to a permutation in MDS view, as shown in Fig. 6b.

In order to deliver more detailed information, in the PCP view, each polyline corresponds to the average of the connection properties of each sequence, and the connection properties between the item in the sequence vary with the sequence. For example, regarding traditional multivariate data that can be represented by parallel coordinates, PCP view shows the average Pearson correlation coefficient, mean variance, and average slope for each sequence after data statistics. In the traveling data, on account of the connection attribute of the item in the sequence, the transportation mode can be arranged and combined, and the system measures the statistical attribute of the mapping sequence by calculating the unified evaluation standard time, price, and distance of different combinations.

In the PCP view, the user can select a specific sequence whose attribute value is within a certain range by brush on each coordinate. The polyline selected by the brush is highlighted, and the unselected polyline is grayed out and remains as background. At the same time, the corresponding point of the brush selection in the PCP view will change to the color corresponding to the highlighted line in the MDS view. In our prototype system, the PCP view and the MDS view are mapped to and complement each other. Users can select the sequence of different attribute values through the brush on the coordinate axis to check the position change of the corresponding sequence in the MDS view. At the same time, users can also find special point cluster that meets user's needs through the PCP view (R3), examine detailed information through PermView, or expand the comparison in PermView through the box selection set of the MDS view.

4.3 PermView

During the design process of PermView, our initial design is a circular petal design; however, the circular space has low utilization of screen space, and it is more likely to cause visual clutter. Therefore, we use the matrix view, which makes full use of screen space and is able to present all sequences..

In PermView, we show all the permutations set in the form of $n \times m$ matrices, where n is equal to the number of items and m is the number of fully arranged sets (m=n!). In the matrix, each column represents a sequence, and the sequence consists of n items from top to bottom. We encode small rectangles with different colors to map different items and represent the difference in sequence order by different distribution of rectangular blocks of different colors in the vertical axis direction. The PermView we designed is a combination of these different color block sequences into a huge matrix (R1), showing all the permutation schemes, providing the user with a sort map of all the sequences.

Of course, as we mentioned in the previous section, the permutation itself is an NP problem. As the number of items increases, the number of permutations increases exponentially. In the process of visual display, the number of m in the $n \times m$ matrix will become very large.

In this regard, our prototype system optimizes the drawing of the matrix view in PermView when the number of items in the dataset is greater than six. The process of sorting and combining items is similar to the expansion of hierarchical data. In the first layer, there are n items to choose from, and the second layer has n-1 possibilities from the respective first layers, and so on. So in the set of all the full permutation sequences schemes, a large number of sequences are arranged in the same position in front of the items. We use the characteristics of the permutation expansion method to redraw the matrix in proportion using the idea of TreeMap. Starting from the first line, n rectangles represent n items, and the tree diagram expands downward. Of course, the matrix thus expanded cannot avoid pixel overlap in the following rows, but compared to the case where each sequence overlaps or the matrix exceeds the screen interface, our optimization method provides a good overview of all the sequences from a visual and user-friendly perspective.

4.3.1 Pixel- and block-based visualization

In order to solve the scalability problem and express more information in the limited display space of the screen, we employ focus+context, a universal visual design principle for exploring big data, to combine two ways, namely pixel-based and block-based visualization, to provide users with an interactive view. We propose a pixel-based visualization to showcase the background (see Fig. 6c1) and a block-based visualization to showcase the details (see Fig. 6c2). Therefore, users could zoom in to see the portion of interest (full detail) and its position in the matrix through our MDS view(R8), while the context structure is expanded in an overview (e.g., a user may want to know that the connection properties between individual items in a permutation (R2)).

In our block-based visualization, we design two sequence rearrangement modes to help users explore the characteristics of the sequence clustering pattern, understand the impact of sorting changes on the sequence and the similarity between the sequences. Sexuality and diversity to address the mission requirements that can be addressed above. Two modes are provided as follows.

4.3.2 *Sequence statistical mode*

According to the proportion of each item in each line, the sequence statistical mode rearranges the order of sequence and displays the aggregation pattern according to the proportion. As shown in Fig. 4, firstly, the number of different items in each row is counted, according to the first item packet rearrangement that appears at the most number of times in one row, so that the sequence of the same item is near in the same position. Therefore, the sequence after grouping continues to rearrange according to the different items of the maximum number of times in the next row, until the whole positions are traversed. The statistical mode of sequence enables users to find out the location distribution of item in the sequence collection of interest very intuitively and find some strong items. The location of these items has a great influence on the comprehensive properties of sequence.

4.3.3 *Alignment mode*

The sequence set in the block-based visualization is rearranged according to the order in which an item appears in the sequence. As shown in Fig. 5, the alignment mode is mainly aimed at some strong item found in the statistical mode. By rearranging and aligning each sequence in the set according to a strong item, it can intuitively discover the influence of the different positions of the strong item on the change of each connection attribute of the sequence in the focus set.

In conclusion, the proposed three main visualizations help us address above eight analysis tasks. The summary is shown in Table 1, which introduces the functions of each visualization in detail.

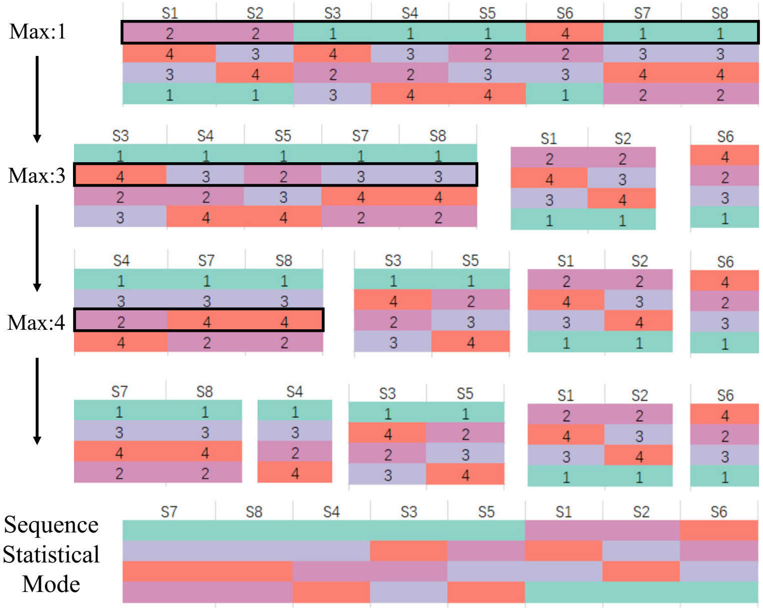


Fig. 4 Demonstration of the generation process of sequence statistical mode. By default, the maximum number of the first item for all permutation is 1, and thus, all the permutations that starting from 1 are aligned to the left. The second and subsequent elements are process in a similar way

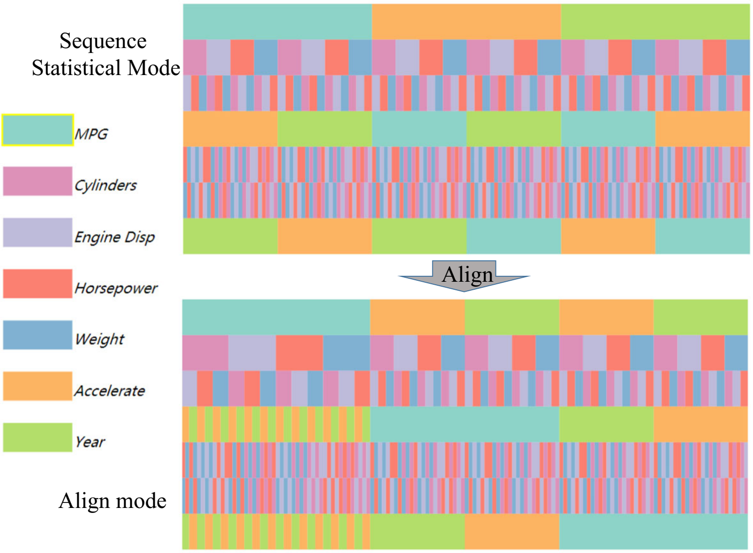


Fig. 5 Statistical mode converts to alignment mode

Table 1 Tasks implementation and mappings with different system visual components

System module	Visual components		Task number
MDS view	MDS-based scatter plot		R3; R4
	DBSCAN-based cluster glyph design		R3; R4
PCP view	Parallel coordinates of the statistical information		R5
PermView	Matrix color block	Pixel- and matrix-based focus + context visualization	R1; R2
		Sequence statistical mode	R7
		Alignment mode	R2; R6; R8

4.4 Interaction

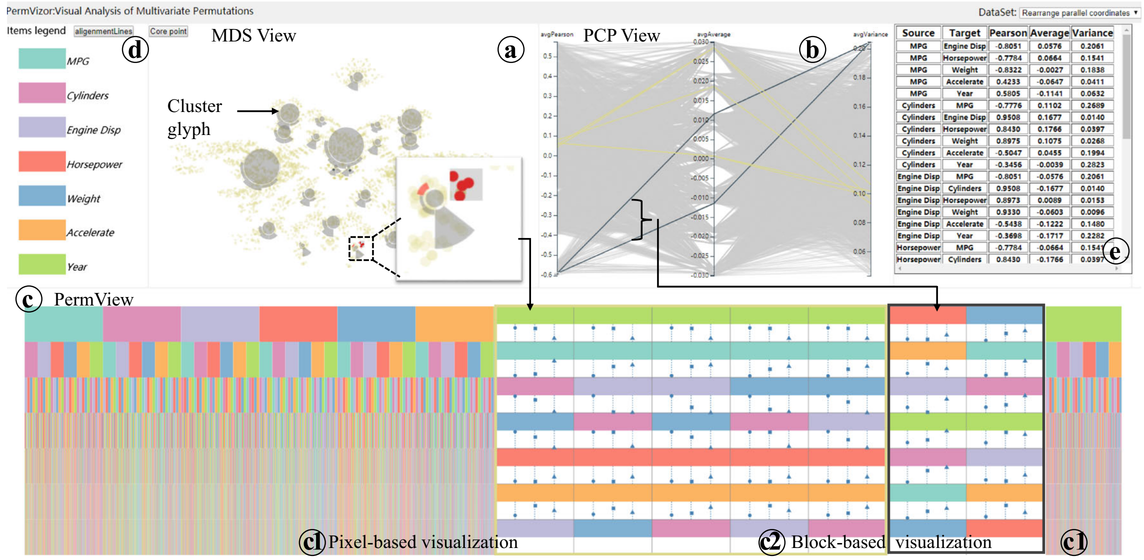
We have developed a set of interactions to help users coordinate the switch between views. First of all, the user can freely select the sequences of the region of interest in the MDS view as the content in the focus view in PermView. Correspondingly, the PCP view shows the average attribute information of all the selected sequences.At the same time, in the MDS view, the user can click on the sector area of the cluster glyph to highlight the corresponding sequence of points to help the user to select the area of interest. In addition, the coordinate selection in the PCP view also has the same response function. The PCP view highlights the selected sequences, and the focus view shows the details of the corresponding sequences. Clicking on the legend rectangles of different colors in the items legend, the focus view will rearrange the sequences according to the corresponding item and display them in alignment mode.

5 Case study

5.1 Data preparation

We collected two types of datasets. The first type of data is vehicle data, which is often used to show data visualization of parallel coordinates. The dataset contains 392 pieces of data, each of which contains one vehicle type attribute and seven vehicle performance attributes. Usually, we use parallel coordinates to visualize the seven performance data attributes. We map seven attributes of this set of data to seven parallel coordinates (items), each of which contains 392 corresponding values.

Another type of dataset is the travel dataset. The travel data contain six popular cities (items) in Scandinavia. Each travel data contains a city information and different connections with other five cities



Source	Target	Pearson	Average	Variance
MPG	Engine Disp	-0.8051	0.0576	0.2061
MPG	Horsepower	-0.7784	0.0664	0.1541
MPG	Weight	-0.8322	-0.0027	0.1838
MPG	Accelerate	0.4233	-0.0647	0.0411
MPG	Year	0.5805	-0.1141	0.0632
Cylinders	MPG	-0.7776	0.1102	0.2689
Cylinders	Engine Disp	0.9508	0.1677	0.0140
Cylinders	Horsepower	0.8430	0.1766	0.0397
Cylinders	Weight	0.8975	0.1075	0.0268
Cylinders	Accelerate	-0.5047	0.0455	0.1994
Cylinders	Year	-0.3456	-0.0039	0.2823
Engine Disp	MPG	-0.8051	-0.0576	0.2061
Engine Disp	Cylinders	0.9508	-0.1677	0.0140
Engine Disp	Horsepower	0.8973	0.0089	0.0153
Engine Disp	Weight	0.9330	-0.0603	0.0096
Engine Disp	Accelerate	-0.5438	-0.1222	0.1480
Engine Disp	Year	-0.3698	-0.1717	0.2282
Horsepower	MPG	-0.7784	-0.0664	0.1541
Horsepower	Cylinders	0.8430	-0.1766	0.0397

Fig. 6 Analyzing the parallel coordinates reordering using PermVizor: **a** MDS view to demonstrate overall distribution of all permutations; **b** PCP view to present statistical information of selected permutations; **c** PermView to display detailed ordering information of each permutation as well as attributes connecting different; **d** legend box to provide color mapping for each item; and **e** table view to present raw data that generates the permutation

(transport modes).The connection attributes are different because of different connection methods (different traffic modes lead to different time, distance, and price).

Finally, we will carry out a full arrangement of two datasets for case analysis. The first set of data is a set of coordinate axes obtained from vehicle data, and the data of the seven axes will get 5040 full permutations. Second datasets are tourist data, and 6-city data will get 720 full permutation results.

For the above two datasets, we extract the connection attributes for their data characteristics. In the parallel coordinate data, hundreds of connecting lines between coordinate axes, we use the Pearson correlation coefficient, the average angle of the connecting line, and the value of the diagonal line variance as the connection attributes between the coordinate axes. We average these three attributes as the connection attributes between cities and cities.

5.2 Reordering in parallel coordinate

For the first use scenario, we use our prototype system to solve the parallel coordinate rearrangement problem. As mentioned in the above data processing, each attribute in the parallel coordinate is treated as an item, and we use the association between attribute values as the connection attribute.

Pearson correlation coefficient, the slope of the connecting line, and the variance of slope change are extracted as connection attributes. These three connection attributes are universal and are applicable to all multi-dimensional attributes displayed by parallel axes. Of course, for different data, users can customize the connection properties between items, for example in the travel data, users can define different modes of traffic as connection properties. Our system attempts to explore arrangement and candidate solutions that conform to user requirements by focusing on the connection properties between items and the sequence arrangement after permutation.

The user first saw the rectangular blocks of the seven items color-mapped in the items legend and had a preliminary understanding of the colors corresponding to the respective items; then the user viewed the MDS view. In the MDS view, the user found that the scatters appear to be symmetrically distributed along a diagonal line in the middle(see Fig. 6a). Obviously, this is because of the inverse numbers between two permutations in a sort. At the same time, the user can clearly find that some scatter points are concentrated in the middle of the MDS view (R3) and are obviously separated from other scatter points. Apparently, this is a special cluster, in which the user was very interested. As shown in Fig. 7, with using the brush provided by the MDS view, the user selected the region of this special cluster to view the specific information of each sequence in the cluster. After the selection, the user found that the corresponding polyline was highlighted in

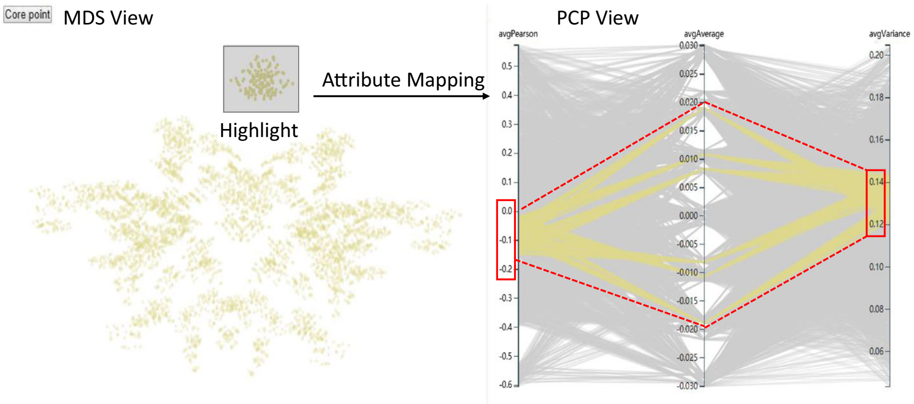


Fig. 7 Select a special cluster from MDS view and map the permutations in the PCP view

the PCP view, showing the statistical average attribute value (R5).The sequence average Pearson correlation coefficient in this cluster was concentrated between 0 and − 0.2, which was in the middle of the corresponding coordinates, and the average variance was concentrated between 0.12 and 0.14, which was also in the middle of the corresponding coordinates. From which, the user learnt that the performance of each aspect in the cluster was at an intermediate level.

The user then turned the attention to the PermView that displays the details(see Fig. 6c). The selected parameters in the focus were presented in a data scale mode. Through this mode, the order of item sorting in each sequence in the cluster can be discovered extremely intuitively. In the focus view, we found some information very interesting, and the three strong items (axis MPG, Accelerate, and Year) occupy the beginning, middle, and last three positions of each sequence. Click on the focus view to further view its detailed connection properties (R2). The user saw through the detail view that the position of the circles on each dotted line bounces very much between the lines and was close to the position above or below the dotted line, i.e., the absolute value of the Pearson correlation coefficient of each item was relatively large, as well as the correlation; the shape positions of the other two attributes changed a little and were all in the middle position. At the same time, the user found that the attribute shapes corresponding to these sequences are very similar in position on the dotted line, although the items in the two positions were different. In other words, when the above three strong items are randomly placed in these three positions, regardless of how the items of other positions are sorted, the various sequences presented will be similar in all aspects. Through this discovery, the user rearranged the parallel coordinates according to the rule of the above exploration and put the coordinate axes MPG, Accelerate, and Year at the beginning, the middle, and the last three positions, with the other coordinates randomly sorted, finally found that the obtained parallel coordinate maps were very similar; the parallel graphs obtained from these sequences also performed very well in terms of the intuitiveness of the information expressed. In the meanwhile, the user also found that the larger the variance of the two items, the lower the visual clutter after the parallel axis was sorted.

Through the above findings, the user decided the sorting scheme: In the three strong items, the user paid attention to the Accelerate information of the car and decided to put the coordinate axis of the attribute Accelerate at the beginning position. In this regard, the user used the alignment mode to rearrange the sequence in the focus view and viewed the detail view in the alignment mode (R7). As shown in Fig. 8, the user found that when the Accelerate was placed at the beginning, there was a higher Pearson correlation coefficient between the two sequences. The average variance was large, so the order of the sequence was selected as the final sorting scheme of the parallel coordinates of the last car data (Accelerate–Horsepower–Cylinders–MPG–Engine–Disp–weight–year).

5.3 Finding specific patterns in parallel coordinates

In the first case, the user has no special requirements for parallel axis rearrangement. The special clusters are directly found through MDS view, and a series of interactive explorations provided by PermVizor are used to determine the better parallel coordinate ordering program. In the second usage scenario, the user wants to find a sorting scheme with high correlation, good data rules, and low visual clutter with using our PermVizor. Therefore, the user customized the cluster glyph according to the Pearson correlation coefficient

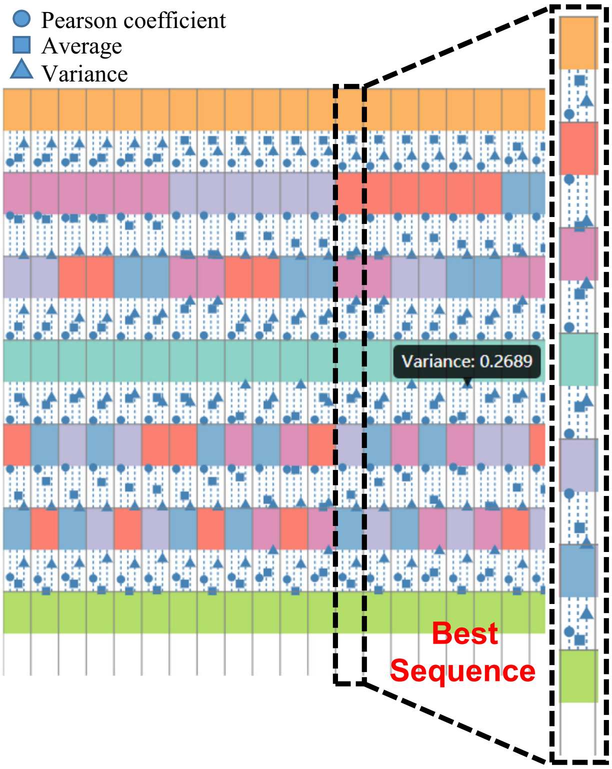


Fig. 8 Block-based view in the PermView helps identify the best and salient candidate sequence, where the attribute of *Accelerate* is the first element

through PermVizor. He clicked the core point button and then loaded the cluster glyph in the MDS view. The user can view the cluster distribution of all the sequences(R3) .

With cluster glyph, as shown in Fig. 9, the user found that only a few clusters have correlation level of six. The user clicked a fan of a small cluster with correlation level of six in the lower right corner, and then the corresponding point appears red in the MDS view (R8). The user viewed the details of the corresponding sequences in the focus view in PermView. From the focus view, the user found that the sequences with correlation level of six arranged very regularly. The strong items (Year, MPG, Horsepower, Accelerate) occupied the first, second, fifth, and sixth positions, respectively, and could be seen in the detail view that the circles between these sequences are located at either ends of the dotted line. The Pearson correlation coefficient was very high, but the triangle shape was at a lower position with a smaller variance, and then these sequences were weaker in visual clutter.

In order to find the best permutation scheme, the user selected a batch of permutation schemes with the largest average variance and the largest average Pearson negative correlation by brushing in parallel coordinates(see the black box selection in Fig. 6c2). The comparison of the selected two batches of sequences is shown in the focus view (R6) . The two sequences selected by the parallel coordinate brush

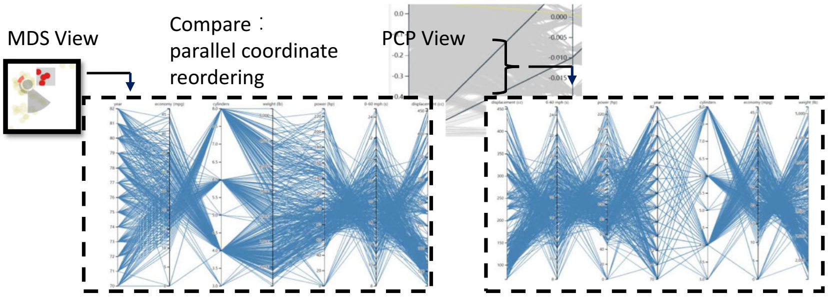


Fig. 9 Best and salient candidate sequence is obtained in the comparison mode

were arranged in reverse order: Year was in the middle position, the circles between the sequences were below the dotted line, and the triangle was above the dotted line. The user randomly selected two sequence permutation schemes from the two clusters, as shown in Fig. 9; with comparison, it was found that the sequences with correlation level of six had high correlation of the overall data in the parallel coordinate view, but the visual performance was messy; regarding the sequence with large variance, we found that the data between the parallel coordinate axes were extremely strong, the angle of the connecting line changed greatly, the overall visual effect was good, the sequence scheme put the Year in the middle, and the other six attributes were on both sides of the Year attribute. The user found it clear and intuitive to find the trend of each property changing with time from the parallel coordinates. After comparison, the user selected a sequence ordering scheme with a large variance as the final scheme.

To evaluate the effectiveness of our approach, we use http://www.bertifier.com/ (Perin et al. 2014), a website with excellent online matrix rearrangement, to sort the seven attributes of the original car data to obtain a new permutation scheme. As shown in Fig. 10, Year is also placed in the middle, and the order of the axes on both sides is slightly different. This permutation scheme is obtained by an automatic permutation algorithm based on the similarity between column data. With the comparison of the permutation scheme derived by the automatic permutation algorithm between the axis alignment scheme obtained by the user through PermVizor, we find that these two are similar in the expression of visual information, and the permutation scheme obtained by the user through PermVizor is slightly superior in visual clutter. This proves that our PermVizor has an advantage in reordering.

5.4 Multi-destination travel planning

In the third case, user Laura wants to find the best city travel arrangements through PermVizor. Laura hopes to visit six popular tourist cities (Paris, Arles, Rome, Venice, Berlin, and Munich) in the Nordic countries. Since European countries are free to enter and exit, although these six cities belong to different countries, they can be treated together as cities within the European Union. Therefore, PermVizor can be used to fully permutate the six countries. Loading the data, Laura first focused on the MDS view, and unfortunately, in the MDS view, Laura did not find the special clustering. It is easy to understand that the transportation between cities cannot form a very strong relationship. Laura then shifted the focus to the PCP view. By brushing, Laura found that the three connection properties (time, price, and distance) of the sequence were positively correlated. Usually, for the transportation between two cities, the time spent and price should be inversely proportional. However, in multi-purpose travel, due to the spatial location between cities, there was a considerable difference in the distance between different arrangements, and the distance determined the time and price. In this regard, Laura selected a permutation scheme with the lowest time, lowest price,

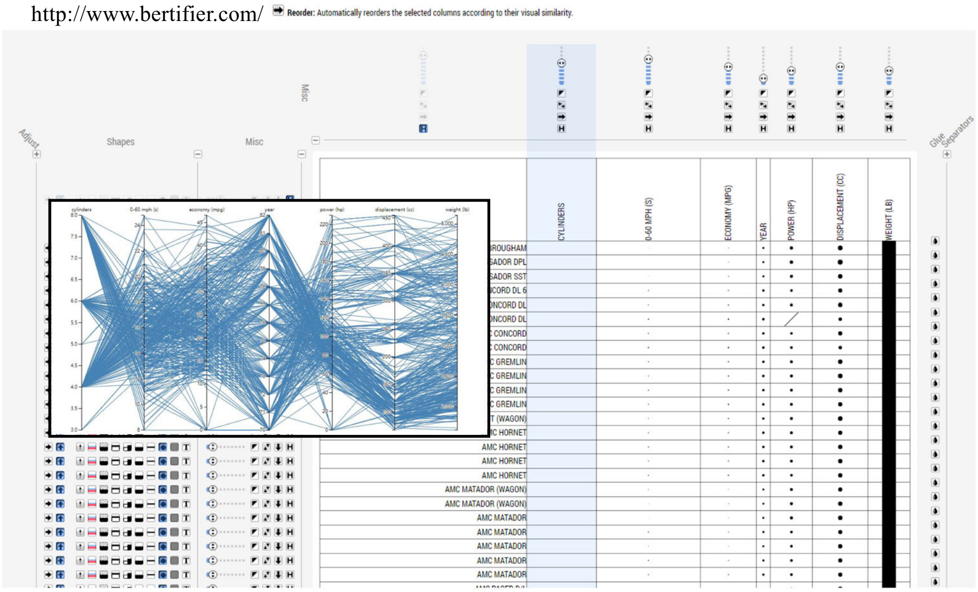


Fig. 10 Automatic sorting of car data obtained from the online tool Bertifier (http://www.bertifier.com/)

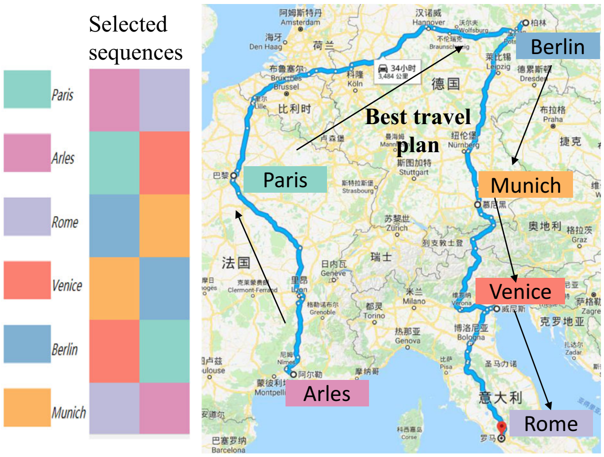


Fig. 11 Best travel solution is obtained with our PermVizor. The best routes selected by Laura start from Arles, pass by Paris, Berlin, Munich, Venice, and end at Rome

and lowest distance through PCP view and checked their specific permutation through PermView. Laura labeled this permutation scheme in Google Maps and found that its trajectory formed a fluctuating ring on the map with no intersections (see Fig. 11). This travel scheme made it easy to travel through six cities in Northern Europe, and then Laura planed her travel arrangements based on PermVizor's permutation scheme.

In this case, the user's results obtained reflect the spatial relationship between the real cities through the traffic connection between the cities. Fully permutation visual analytics can mine the other hidden information content with easy-to-understand connection information for user-defined data.

6 Conclusion

Permutation is one of the common topics in combinatorics which is widely used in various domains. Understanding the meanings and obtaining the insights behind each permutation presents many useful applications and leads solution to many scenarios. We propose PermVizor, a new visual analytic system that aims assisting users exploring the arrangement, distribution, and comparison of permutations from a user-friendly perspective. Our model is constructed based on MDS algorithm, to extract the order and connection properties from multivariate sequences. Kendall distance and Euclidean distance are combined to measure the vector similarity and to create the dissimilarity matrix. This model overall lays a foundation for visual components and analysis tasks.

Our work is a new attempt to use visualization technology on the research of multivariate permutation data. Due to the diversity of visualization technology, there are more other solutions, so the visual analysis method we designed is not the most perfect. We try to use the pixel-based graph visualization technology to solve issue of displaying all permutations, but the scalability of permutations itself is difficult to solve. Once the number of the all permutation projects exceeds 10, it will be difficult for our visualization scheme to solve. For the follow-up work, we will challenge a larger number of all permutation projects problems. We will solve the problem of scalability by optimizing the generation algorithm and clustering method of the all permutations, combined with new visualization design. Hope that our work will inspire other researchers and enable more people to participate in the research of using visualization technology to explore the full permutation data.

Acknowledgements This work is partly supported by National Natural Science Foundation of China (No. 61972356), Zhejiang Provincial Natural Science Foundation of China (No. LY19F020026), National Natural Science Foundation of China (No. 61602409), Zhejiang Povincial Key Research and Development Program of China No. 2019C01009), and Fundamental Research Funds for the Provincial Universities of Zhejiang (No. RF-C2019001).

References

Apostolico A, Hein J (2005) Combinatorial pattern matching, vol 213, 1st edn. Springer, Berlin, pp 8–289

Bertin J (1981) Graphics and graphic information processing. In: Graphics and graphic information processing, pp 24–31. https://doi.org/10.1515/9783110854688

Blaas J, Botha CP, Post FH (2008) Exetensions of parallel coordinates for interactive exploration of large multi-timepoint data sets. IEEE Trans Vis Comput Gr 14(6):1436–1443

Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci USA 95(25):14863–14868

Friendly M, Kwan E (2003) Effect ordering for data displays. Comput Stat Data Anal 43(4):509–539

Heinrich J, Stasko J, Weiskopf D (2012) The parallel coordinates matrix. In: Proceeding of eurographics conference on visualization—short papers, pp 37–41. https://doi.org/10.2312/PE/EuroVisShort/EuroVisShort2012/037-041

Hurley CB, Oldford RW (2010) Pairwise display of high-dimensional information via eulerian tours and hamiltonian decompositions. J Comput Gr Stat 19(4):861–886. https://doi.org/10.1198/jcgs.2010.09136

Johansson J, Forsell C (2016) Evaluation of parallel coordinates : overview. Categ Guidel Future Res 22(1):579–588. https://doi.org/10.1109/TVCG.2015.2466992

Kendall MG (1938) A new measure of rank correlation. Biometrika 30(1/2):81–93

Kim B, Lee B, Seo J (2007) Visualizing set concordance with permutation matrices and fan diagrams. Interact Comput 19(5–6):630–643. https://doi.org/10.1016/j.intcom.2007.05.004

Liiv I (2010) Seriation and matrix reordering methods: an historical overview. Stat Anal Data Min ASA Data Sci J 3(2):70–91

Liu CL (1968) Introduction to combinatorial mathematics. McGraw-Hill College, New York, pp 1–18

Lu LF, Huang ML, Huang TH (2012) A new axes re-ordering method in parallel coordinates visualization. In: Proceedings of 11th international conference on machine learning and applications, vol 2. ICMLA, pp 252–257. https://doi.org/10.1109/ICMLA.2012.148

Lu LF, Huang ML, Zhang J (2016) Two axes re-ordering methods in parallel coordinates plots. J Vis Lang Comput 33(1):3–12

Massini A (2003) All-to-all personalized communication on multistage interconnection networks. Elsevier, Amsterdam

McCoy NH (1968) Introduction to modern algebra. Allyn and Bacon, Boston

Munakata T, Barták R (2006) Combinatorics in logic programming: implementations and applications. Int J Inf Technol Intell Comput 1(2):419–428

Mycielski J, Rozenberg G, Salomaa A (1997) Structures in logic and computer science. Springer, Berlin

Paterson T, Graham M, Kennedy J, Law A (2011) Evaluating the VIPER pedigree visualisation: detecting inheritance inconsistencies in genotyped pedigrees. In: IEEE symposium on biological data visualization, BioVis 2011-proceedings, pp 119–126. https://doi.org/10.1109/BioVis.2011.6094056

Perin C, Dragicevic P, Fekete JD (2014) Revisiting Bertin matrices: new interactions for crafting tabular visualizations. IEEE Trans Vis Comput Gr 20(12):2082–2091. https://doi.org/10.1109/TVCG.2014.2346279

Pilhofer A, Gribov A, Unwin A (2012) Comparing clusterings using Bertin's idea. IEEE Trans Vis Comput Gr 18(12):2506–2515. https://doi.org/10.1109/TVCG.2012.207

Roberts F, Tesman B (2009) Applied combinatorics. CRC Press, Boca Raton

Stallings W (2003) Cryptography and network security: principles and practice, vol 46, 3rd edn. International Annals of Criminology, Rome, pp 121–136

Stoll RR (2013) Linear algebra and matrix theory. Courier Corporation, Mineola

Sugiyama K (2002) Graph drawing and applications for software and knowledge engineers, vol 11. World Scientific, Singapore

Sun GD, Wu YC, Liang RH, Liu SX (2013) A survey of visual analytics techniques and applications: state-of-the-art research and future challenges. J Comput Sci Technol 28(5):852–867. https://doi.org/10.1007/s11390-013-1383-8

Tennekes M, De Jonge E (2014) Tree colors: color schemes for tree-structured data. IEEE Trans Vis Comput Gr 20(12):2072–2081. https://doi.org/10.1109/TVCG.2014.2346277

Yang Y, Wang J (2004) Routing permutations on optical baseline networks with node-disjoint paths. In: Proceedings of tenth international conference on parallel and distributed systems. ICPADS 2004. IEEE, pp 65–72