DOI: 10.1111/cgf.14290
Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

Exploring Multi-dimensional Data via Subset Embedding

Peng Xie[1], Wenyuan Tao[1], Jie Li[1†], Wentao Huang[1] and Siming Chen[2]

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]School of Data Science, Fudan University, Shanghai, China

Abstract

Multi-dimensional data exploration is a classic research topic in visualization. Most existing approaches are designed for identifying record patterns in dimensional space or subspace. In this paper, we propose a visual analytics approach to exploring subset patterns. The core of the approach is a subset embedding network (SEN) that represents a group of subsets as uniformly-formatted embeddings. We implement the SEN as multiple subnets with separate loss functions. The design enables to handle arbitrary subsets and capture the similarity of subsets on single features, thus achieving accurate pattern exploration, which in most cases is searching for subsets having similar values on few features. Moreover, each subnet is a fully-connected neural network with one hidden layer. The simple structure brings high training efficiency. We integrate the SEN into a visualization system that achieves a 3-step workflow. Specifically, analysts (1) partition the given dataset into subsets, (2) select portions in a projected latent space created using the SEN, and (3) determine the existence of patterns within selected subsets. Generally, the system combines visualizations, interactions, automatic methods, and quantitative measures to balance the exploration flexibility and operation efficiency, and improve the interpretability and faithfulness of the identified patterns. Case studies and quantitative experiments on multiple open datasets demonstrate the general applicability and effectiveness of our approach.

CCS Concepts

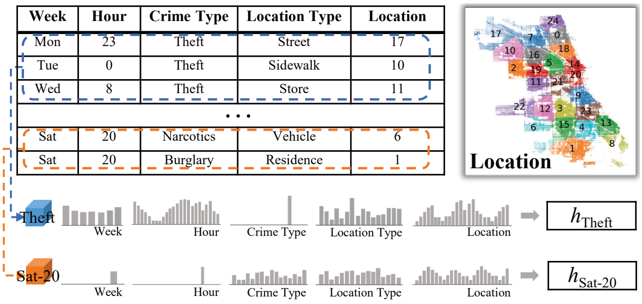
• ***Human-centered computing*** → ***Visual analytics; Visualization systems and tools;***

1. Introduction

Multi-dimensional data exploration is a classic research topic in visualization. Most existing approaches work at the record level, using various machine learning algorithms to find distinctive distributions (e.g. clusters or outliers) in dimensional space or subspace as data patterns [YRWG13, PPM*15, XYC*17].

Patterns of multi-dimensional data, however, are often related to subsets rather than records. A ***subset*** consists of data records and has multiple features, as in Figure 1. Each ***feature*** reflects an aspect of statistical information of all the included records. Typically, a ***subset pattern*** is a group of subsets having similar values on specific features. We consider exploring subset patterns a more general task for multi-dimensional data, since a subset can only include one data record in an extreme setting. In that case, the subset pattern exploration degenerates to the record pattern exploration.

There are many ways to partition a given multi-dimensional dataset into subsets and a subset can have a large number of features. Subset patterns, however, may associate with a few subsets and features. Without prior knowledge, analysts have to attempt different partition methods, repeatedly select a portion of generated subsets, and correlate them on different combinations of features to identify patterns. The huge search space makes the discovery of subset patterns a challenging and time-consuming process.



Week	Hour	Crime Type	Location Type	Location
Mon	23	Theft	Street	17
Tue	0	Theft	Sidewalk	10
Wed	8	Theft	Store	11
. . .				
Sat	20	Narcotics	Vehicle	6
Sat	20	Burglary	Residence	1

Figure 1: *Two subsets of the Chicago crime dataset [chi]. Each consists of records (crimes) with specified attribute values, and takes the distribution of the number of records on an attribute as a feature. The purpose of SEN is representing a large number of subset as uniformly-formatted vectors.*

The great success of representation learning [BCV13] inspires us to apply a subset embedding network (SEN) in multi-dimensional data exploration. The SEN can generate uniformly-formatted embeddings for a group of given subsets, as in Figure 1. Moreover, the embedding similarity reflects that of subsets in terms of their

[†] Jie Li is the corresponding author. Email: jie.li@tju.edu.cn

feature values. We thus can perform automatic algorithms, such as clustering and dimensionality reduction (DR), according to their embeddings to identify patterns in an efficient manner.

In this paper, we propose the SEN. The design challenges include 1) adapting the SEN to arbitrary subsets to achieve better applicability (R1), 2) capturing the similarity on single features to accurately encode patterns (R2), and 3) obtaining a high training efficiency to enable incorporation into visualization systems (R3). The three aspects prevent the application of existing techniques in subset embedding (Section 2.2). Inspired by multi-view learning [LYZ18], we consider a subset a multi-view object and each subset feature a snapshot taken by a virtual camera around the subset (Section 2.3), and propose SEN of multi-subnet structure to satisfy the three requirements.

We integrate the SEN into a visualization system to explore subset patterns in multi-dimensional data. The system follows an “overview->details->patterns” explorative workflow [Shn96]: analysts slice the data into subsets, project subsets according to their embeddings obtained from a SEN trained on-the-fly, and select portions of subsets to determine the existence of patterns. The system contains three components. Specifically, one utilizes the tree metaphor to achieve progressive data partition, which enables analysts to generate a variety of subsets through few operations. Another incorporates interactions and automatic methods to assist in exploring the projection. The third one provides a group of views implemented as classic visualization techniques for visualizing features of selected subsets to identify patterns. Generally, the three components balance the exploration flexibility and operation efficiency, and improve the interpretability and faithfulness of the identified patterns.

We conduct case studies and quantitative experiments to evaluate the approach on six open datasets. Experiment results illustrate its general applicability and effectiveness. Specifically, analysts can identify rich patterns using our approach by conducting tasks on drastically different subsets. Meanwhile, quantitative experiments demonstrate the high training efficiency and the effectiveness in capturing patterns of the SEN.

The main contribution of our work is a visualization approach to exploring subset patterns in multi-dimensional data, which integrates 1) ***a subset embedding network*** that can accurately and quickly represent a large number of given subsets as uniformly-formatted embeddings , and 2) ***a visualization system*** following a classic workflow, which combines 1) with three visual components to implement flexible and efficient subset pattern exploration.

The rest of the paper is organized as follows. Section 2 gives design requirements. Sections 3 and 4 introduce the embedding network and the visualization system. Sections 5 demonstrates the usability of our approach through case studies and quantitative experiments. Section 6 discusses the limitations. We reviews the related work in Section 7 and conclude the paper in Section 8.

2. Problem Statement

We give the subset definition, identify three requirements, and outline the approach.

2.1. Subset Definition

Let $D(d_1,...,d_n)$ be a multi-dimensional dataset, where d_i is an attribute with the domain $dom(d_i)$. A ***subset*** consists of records selected by a group of filters, i.e. $r(d_1),...,r(d_n)$, where $r(d_i)$ is a filter that specifies a value range on $dom(d_i)$, i.e. $r(d_i) \in dom(d_i)$.

Filters whose value ranges cover the whole domain, i.e. $r(d_i) = dom(d_i)$, can be hidden for brevity. We call the attribute of an unhidden filter a ***slicing attribute***. For example, in Figure 1, the subset (Crime-type: theft) has a single slicing attribute, i.e. Crime-type, while the subset (Week: Saturday, Hour: 20) takes Week and Hour as two slicing attributes.

The number of unhidden filters describes the ***dimensionality*** of the subset, denoted as $l = |\{r(d_i)|r(d_i) \neq dom(d_i)\}|$. For example, (Crime-type: theft) is a 1-dimensional subset, and (Week: Saturday, Hour: 20) is a 2-dimensional subset.

Each subset can have multiple ***features***, as in Figure 1. We write the vth feature of subset S_i as $X_i^{(v)}$. A feature, describing an aspect of statistical information of all the included records, can be in any form, such as a number (e.g. count of all included records) or a vector (e.g. distribution of aggregate values on an attribute).

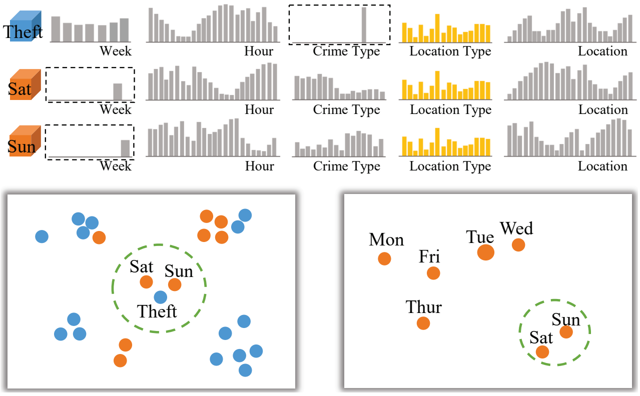
2.2. Design Requirements

According to the above definition, we can represent a subset as a group of feature vectors. The function of the SEN thus is to take feature vectors of a subset as the input and output its embedding. The nature of subset, however, makes existing embedding techniques [MH08, XMTH11] inapplicable, as follows:

First, existing techniques target on objects with features of the same shape (e.g. number and size). Patterns, however, may exist within arbitrary subsets that have different slicing attributes and features. Figure 2a show such a case, in which we project subsets sliced on Crime-type and Week separately together and exclude features of slicing attributes (marked with dashed borders). Taking them as features will incorrectly increase the similarity of subsets, since their values are zero at most positions. Projections containing diverse subsets involve more interesting patterns. In Figure 2a, the theft subset is adjacent to Saturday and Sunday subsets. We thus speculate theft crimes mainly occur at that time. For applicability, the SEN should be able to deal with subsets with arbitrary slicing attributes (R1).

Second, existing techniques are designed to capture the overall similarity of target objects across all features. In contrast, subset patterns are often related to few features. For example, in Figure 2b, two week subsets (Saturday and Sunday) have similar values on a feature (see the feature marked in yellow). Existing techniques cannot output similar embeddings for them due to their significant differences on the other features, resulting in missing patterns or finding incorrect patterns when exploring the latent space. To capture the similarity of subsets on single features is necessary for the SEN (R2).

Finally, existing techniques are always restricted by the slow training speed. Moreover, as a self-supervised technique, a well-trained embedding network can only output embeddings for the ob-



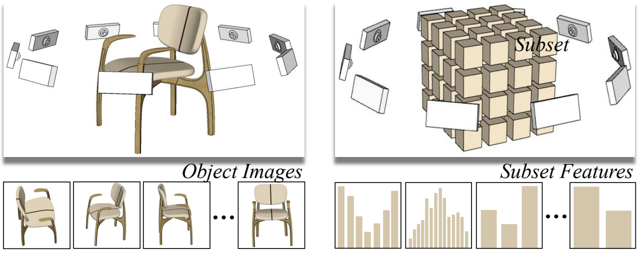
(a) (b)

Figure 2: *Two important design requirements of the SEN that should (a) accommodate arbitrary subsets with different slicing attributes and features and (b) capture the similarity of subsets on single features.*

jects in the training set, different from the offline training of classification models. Therefore, we have to train a SEN on-the-fly after slicing a multi-dimensional data into subsets. A high training efficiency is desired for the network to facilitate the application in a visual analytics environment (R3).

2.3. Mutli-view Learning-inspired Subset Embedding

We use the idea of multi-view learning to design the SEN. Multi-view learning is a common kind of machine learning techniques for multi-view objects [LYZ18, ZXXS17, JY17]. A typical multi-view learning model takes multiple snapshots (views) of an object as the input, as in Figure 3a. Its purpose is taking advantage of the complementary information between views to generate embeddings, thus improving the accuracy of the following object identification or classification tasks. Inspired by multi-view learning, we consider each subset a multi-view object. Each feature can be viewed as a snapshot taken by a virtual camera from different perspectives around the subset, as in Figure 3b.



(a) (b)

Figure 3: *Multi-view learning-inspired subset embedding. (a) A multi-view learning model takes multiple views (snapshot) of a target object as the input. (b) We consider a subset as a multi-view object with each feature being a snapshot taken by a virtual camera around the subset.*

A common characteristic of multi-view learning techniques is they separately treat individual views. Along this line, we propose a SEN of multi-subnet structure, as in Figure 4b. The core idea of the structure is using independent subnets to handle different features and fuse information of different features into final embeddings. In the next section, we will show how the multi-subnet structure satisfies the above three requirements.

3. Subset Embedding Network

We propose a SEN of multi-subnet structure. Without loss of generality, we allow target subsets to have different numbers of features, as in Figure 4a. The network structure is shown in Figure 4b. The SEN consists of multiple subnets ($f^{(1)}$,$f^{(2)}$,$\dots$,$f^{(n)}$), each corresponding to a feature. Let h_i be the embedding of subset S_i. A subnet $f^{(v)}$ takes h_i as the input and predicts the vth feature vector of S_i. All the subnets share embeddings of subsets as inputs. The embeddings are randomly initialized and iteratively updated during the training of the subnets. By using the multi-subnet structure, the network size is linearly proportional to the number of features (the number of subnets). Below we define the losses for updating embeddings and parameters of each subnet.

Let $Loss^{(v)}$ be the loss of the vth subnet $f^{(v)}$, we use the sum of losses of all the subnets to update the embeddings, i.e.:

$$Loss_{emb} = \sum_{v=1}^{n} Loss^{(v)} \tag{1}$$

We use the difference between the predicted feature vector $f^{(v)}(h_i)$ and the original feature vector $X_i^{(v)}$ as the loss of subnet $f^{(v)}$, i.e.:

$$Loss^{(v)} = \sum_{i \in I^{(v)}} \left\| f^{(v)}(h_i) - X_i^{(v)} \right\| \tag{2}$$

$I^{(v)}$ represents the set of subsets containing the vth feature. Using Figure 4a as an example, S_1 belongs to $I^{(1)}$, $I^{(3)}$ and $I^{(n)}$, while S_i belongs to $I^{(2)}$ and $I^{(n)}$. By introducing $I^{(v)}$, each subset will activate different subnets. That is we only use the losses of subnets corresponding to features owned by a subset to update its embedding. The structure thus is applicable for arbitrary subsets with different numbers of features (R1).

The multi-subnet structure enables to capture the similarity of subsets on single features. First, embeddings of subsets with few similar features can be similar. Let (h_i, h_j) and $(X_i^{(v)}, X_j^{(v)})$ be the embeddings and the vth feature vectors of two subsets S_i and S_j. When $X_i^{(v)}$ and $X_j^{(v)}$ are similar, h_i and h_j should be similar to some extent, as they will go through the same subnet $f^{(v)}$ to obtain similar outputs, i.e. h_i->$f^{(v)}$->$X_i^{(v)}$, h_j->$f^{(v)}$->$X_j^{(v)}$. Moreover, subsets with more similar features will have more similar embeddings, as we use the sum of losses of all subnets (Equation 1) for updating embeddings (R2).

We implement each subnet as a fully-connected neural network containing a single hidden layer. We fix the length of embeddings to 30, which achieves relatively good performance in most cases. We can conveniently increase the size to reduce the information loss

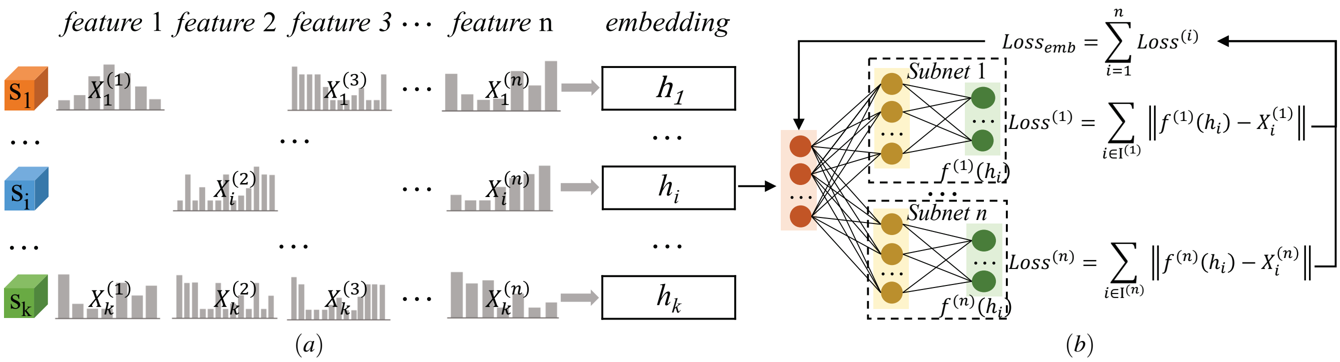


Figure 4: *Subset embedding network. (a) Subsets to be embedded, which are allowed to have different numbers of features. (b) The multi-subnet structure of the SEN. All the subnets share embeddings of subsets as their inputs.*

of embedding. Training optimizations, such as learning rate decay, early stop, etc., can be used optionally. The simple structure and few parameters achieve high training efficiency. The network can handle a large number of subsets with multiple features in real-time (Section 5.3), enabling easy integration with visualization systems (R3).

Algorithm 1 shows the training process of the SEN. The network takes a group of subsets represented as feature vectors as inputs and outputs uniformly-formatted embeddings for them. We randomly initialize embeddings and parameters of subnets (line 1), train subnets with separate losses using Equation 2 (lines 3-5) and update embeddings using Equation 1 (lines 6). We will terminate the training when $Loss_{emb}$ (Equation 1) no longer significantly decreases over multiple consecutive epochs.

Algorithm 1: Subset Embedding Network Training

```
  Input: subsets $\{S_1,...,S_k\}$, $S_i = \left\{X_i^{(1)},...,X_i^{(n)}\right\}$
1 Randomly initialize embeddings $\{h_1,...,h_k\}$ and parameters
    of subnets $\left\{\theta^{(1)},...,\theta^{(n)}\right\}$
2 while not converged do
3     for $v$ = $1 : n$ do
4         Update $\theta^{(v)}$ with $Loss^{(v)} = \sum_{i \in I^{(v)}} \left\| f^{(v)}(h_i) - X_i^{(v)} \right\|$
5     end
6     Update $\{h_1,...,h_k\}$ with $Loss_{emb} = \sum_{v=1}^{n} Loss^{(v)}$
7 end
  Output: embeddings $\{h_1,...,h_k\}$
```

4. Visualization System

We develop a visualization system based on the SEN for exploring multi-dimensional data, detailed below.

4.1. System Overview

Following the line of "overview->details->patterns" [Shn96], we propose an explorative workflow, as in Figure 5. Analysts first partition the given multi-dimensional dataset into subsets (**Progressive Data Partition**). They then project the subsets according to their embeddings obtained from a SEN trained on-the-fly, and select subsets with specific distributions (e.g. outliers or clusters) in the projection (**Latent Space Creation & Exploration**). Analysts finally visualize feature vectors of the selected subsets and observe whether they have consistent values on any features. The more consistent feature values a group of selected subsets have, the more significant patterns they involve (**Visual Pattern Discovery**).

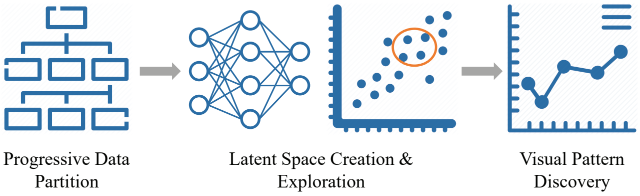

Progressive Data Partition
Latent Space Creation & Exploration
Visual Pattern Discovery

Figure 5: *The workflow utilized by the visualization system.*

Figure 6 shows the interface of the visualization system that integrates three components, i.e. Exploration Manager (EM), Subset Projector (SP), and Pattern Explainer (PE), to achieve the above workflow, as follows:

EM implements the progressive data partition that enables analysts to generate a variety of subsets with few operations (step 1), as in Figure 6a. Its main body is a tree. Each node contains subsets selected together from those sliced on its parent.

SP is responsible for creating the projected latent space, in which analysts select subsets that may involve patterns (step 2), as in Figure 6b. It integrates rich interactions and automatic methods to assist in selecting subsets.

PE consists of a group of feature cards implemented as classic visualization techniques to show and compare feature values of selected subsets (step 3), as in Figure 6c. It uses a consistency measure (Section 4.4) to quantify the significance of the patterns.

4.2. Progressive Data Partition

To enable analysts' exploration, we should generate a large number of subsets at the beginning of the exploration. For flexibility,

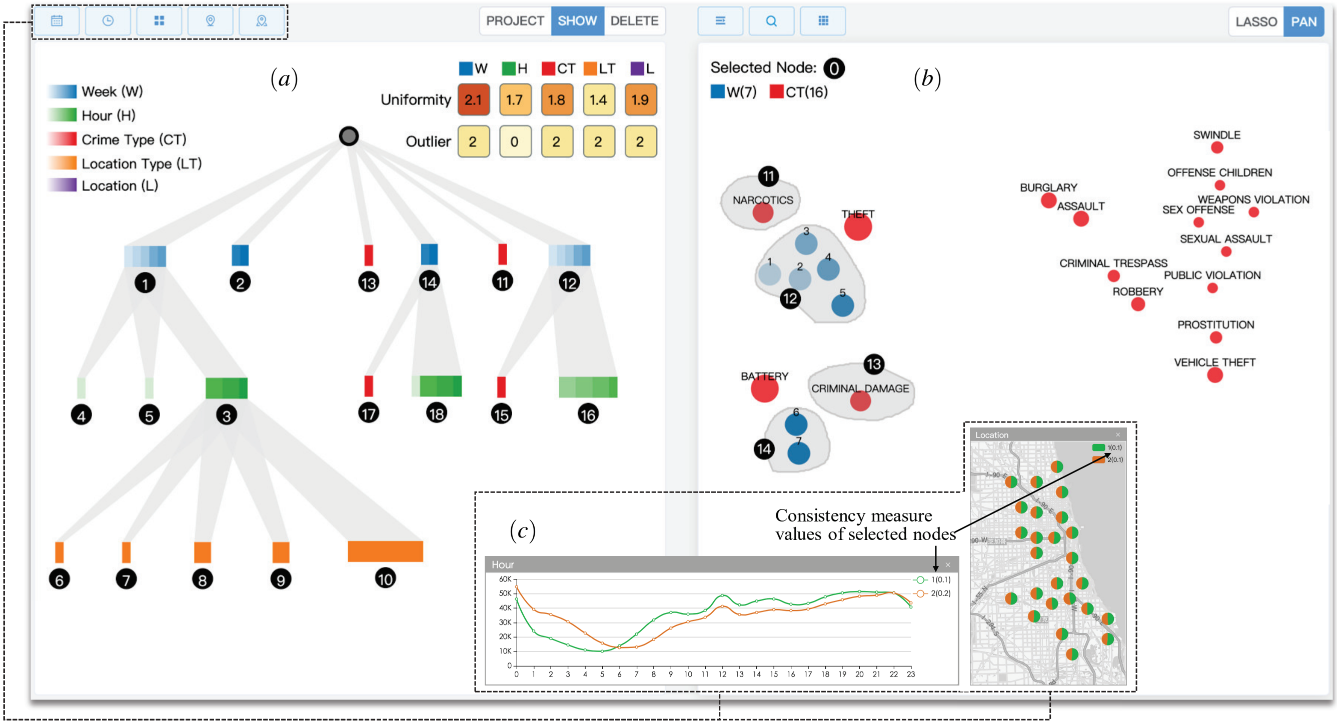


Figure 6: *The interface of the visualization system that integrates three components, i.e. (a) Exploration Manager (EM), (b) Subset Projector (SP), and (c) Pattern Explainer (PE).*

the visualization system should support to generate subsets with any slicing attributes covering any attribute ranges. For efficiency, it is impossible to set slicing attributes and attribute ranges for each subset individually, which will introduce a large number of repetitive operations. We thus propose the progressive data partition to balance the two aspects:

First, we limit to slicing data on an attribute. Therefore, each generated subset has a single slicing attribute. Moreover, we consider subsets selected at a time as a new dataset and allow for partitioning it into subsets again, which actually add a slicing attribute.

Second, we discretize the value range of the slicing attribute. Specifically, geographical locations are grouped into nominal units, such as states, cities or zones; temporal attributes are discretized on their natural intervals, such as hours, days, or weeks; numerical attributes are divided into equal intervals. We make each generated subset cover a minimum value interval.

The visualization system utilizes EM to implement the progressive data partition. The main body of EM is a tree, as in Figure 6a. We utilize Figure 7 as an example to illustrate how analysts use EM to generate subsets with different slicing attributes. At the beginning of the exploration, there is only the root in the tree, which represents the original dataset. Analysts can select the root and specify "Week" as the slicing attribute to partition it into 7 subsets and project all the generated subsets, as in Figure 7a. They then select two groups of subsets corresponding to the weekday and the weekend respectively, which form separate child nodes of the root, as in Figure 7b-c. Analysts further select the newly added node that contains subsets of weekend and partition it into 24 subsets on Hour, as in Figure 7d. Each subset thus represents an hour on weekends. They also select two groups of subsets to form two child nodes of the weekend node, as in Figure 7e-f.

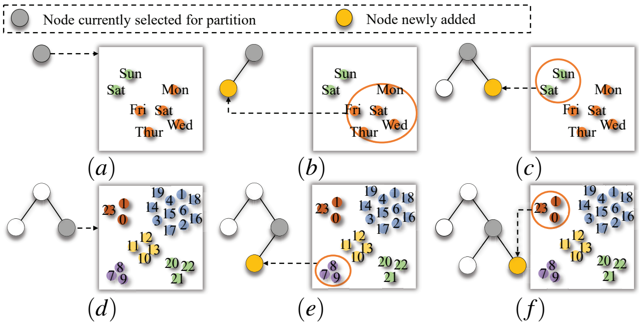


Figure 7: *An example of the progressive data partition. Subsets in a cluster are marked with the same color.*

Analysts can generate arbitrary subsets they want by repeating the above process. Moreover, the progressive process balances the flexibility and efficiency of the subset generation. First, it simplifies interactive operations. Analysts can simply choose a slicing attribute to generate subsets without the needs of setting the attribute range for each subset individually. Second, it makes the implementation of the visualization system easy. The system only needs to support a type of operations, i.e. slicing a dataset into subsets on a user-specified attribute. Finally, each projected subset covers a unit of the value range of the slicing attribute, which avoids generating a large number of logically-unrelated subsets, making identified patterns more explainable.

In EM, each tree node consists of multiple colored rectangles, as in Figure 6a. Each rectangle represents a subset that covers a single unit of the value range of the slicing attribute. The more subsets a node contains, the longer the node will be. We assign a globally unique color to an attribute, see the legend in Figure 6a. The color of a rectangle indicates the slicing attribute selected at the current round. We can understand all the slicing attributes of subsets of a node by tracing colors of nodes along the path from the root to the node. For example, in Figure 6a, #1 has a single slicing attribute, i.e. Week, while #3 has two slicing attributes, i.e. Week and Hour. For attributes with continuous attribute ranges (e.g. Week and Hour), the transparency of rectangles (subsets) gradually decreases as the attribute value increases. Rectangles of categorical attributes have the same transparency.

EM will display two measure values, i.e. uniformity and the number of outliers [SS04] for each marginal distribution of the selected node (see the colorful rectangles in the upper right corner of Figure 6a, which is showing ten measure values of the root node on five attributes). The two measures provide important guidance for selecting slicing attributes, especially helpful when the dataset contains many attributes. A larger or smaller value may relate to potential patterns. Therefore, analysts can slice the node on the corresponding attribute.

4.3. Latent Space Creation & Exploration

We extract features for sliced subsets, project them according to their embeddings obtained from a SEN trained on-the-fly, and select parts of subsets that may involve patterns, as follows:

Feature Extraction. We calculate the distribution of the number of records on an attribute as a feature for a subset, as in Figure 2. It is also possible to add/remove features, or use other features for each subset. We exclude features of slicing attributes (the reason has been explained in Section 2.2). Using the five-dimensional Chicago crime dataset as an example (Figure 1), each 1-dimensional subset has four features, each 2-dimensional subset has three features, and so on.

Representation Learning & Projection. We train a SEN to obtain uniformly-formatted embeddings of the subsets. We then project the subsets onto a 2-dimensional plane according to their embeddings. We choose t-SNE [MH08] to generate the projection. Other dimensionality reduction techniques can also be used.

Subset Selection. Each point in the generated projection represents a subset, as in Figure 6b. The size of a point encodes the number of records contained in the subset. Analysts select subsets in the projection, which are mapped together as a child of the selected node in the tree. Common interactions, such as zoom, pan, lasso, etc., are supported by SP.

There are three ways to select subsets in the SP, as follows:

1) Analysts can freely select subsets according to the distribution of the projected subsets. Significant clusters or outliers are possible candidates for selection.

2) Analysts can highlight projected subsets within specified attribute ranges and select the highlighted subsets exclusively. For example, in Figure 9c, we highlight seven subsets (sliced on Hour). A common explorative strategy is to change the queried attribute range and observe the distribution of the highlighted subsets in the projection before selecting subsets. Having found any interesting distributions (e.g. clusters or outliers), analysts can select them for further in-depth analysis.

3) Analysts can use the clustering function to divide projected subsets into groups according to their embeddings automatically. Many clustering algorithms, such as K-means, hierarchical clustering, density-based clustering (no need to specify the number of clusters), etc., are integrated. Analysts can set parameters, such as the number of clusters, according to their prior assumptions. For example, we can divide seven subsets sliced on Week into two clusters, considering weekday and weekend are naturally different on many features. They can also interactively adjust the parameter during the exploration. Clustering results, whether expected or not, provide cues for the subset selection. A group of links will appear on the right of the projection after clustering. Each link corresponds to a cluster of subsets, as in Figure 9d. We can click a link to highlight the corresponding subsets.

4.4. Visual Pattern Discovery

We use PE to visualize the feature vectors of subsets of different nodes in the tree, as in Figure 6c. Feature vectors of subsets in a node are aggregated and visualized as a whole in each feature card. Each feature card corresponds to a feature and is implemented as a classical visualization technique suitable for the feature. For example, line chart is for showing temporal patterns; bar chart is for showing categorical patterns; thematic map (pie chart + map) is for showing spatial patterns. Analysts can select multiple nodes and add them into a feature card for comparison. Figure 6c contain two nodes respectively, in which each visual item (line/bar/sector) reflects the aggregated feature vector of all the subsets in a node. A feature card will appear when analysts click on the corresponding button in the upper left corner of the system, marked with dashed borders in Figure 6.

We propose a measure to describe the consistency of feature vectors of subsets contained in a node using $\frac{1}{D}\sum_{d=1}^{D}\sigma(d)$, where D represents the size of the feature vector, and $\sigma(d)$ represents the standard deviation of feature vector values at the dth positions of all subsets. A lower measure value indicates more consistent feature vectors (as in Figure 8), i.e. more significant subset patterns. We visually encode the measure value for each node in feature cards (see numbers in legends in Figure 6c).

4.5. Implementation

The visualization system has a client-server architecture. The SEN is at the backend, which receives requests from the frontend (written in JavaScript using D3 library [BOH11]) and sends back embeddings. The Flask is used to transfer parameters between the

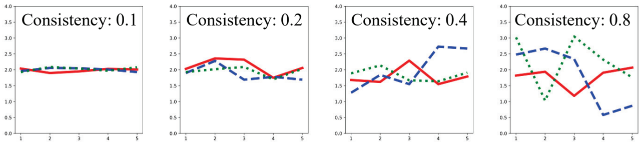


Figure 8: *Four groups randomly-generated vectors with different consistency measure values.*

front and back ends. We also build an OLAP index, i.e. data cube, to speed up the feature extraction from the original data.

5. Evaluation

We conduct case studies and quantitative experiments to demonstrate the applicability of our approach in actual scenarios.

5.1. Case Studies

We develop a visualization system on the Chicago crime dataset [chi] (about 6M records) and perform two categories of tasks to identify patterns at group and individual levels respectively:

Group Segmentation. We partition data into subsets on gradually-increasing slicing attributes. For each partition, we select groups of subsets with similar feature values. Each group thus indicates a kind of crime patterns. The exploration process is shown in Figure 9a and detailed below.

We choose Week with the highest uniformity value as the slicing attribute to partition the data into 7 subsets. We project the subsets (Figure 9b) using t-SNE (perplexity is set to 20, while other hyperparameters are kept as default settings) and find two obvious clusters, i.e. weekday (#Mon- #Fri) and weekend (#Sat-#Sun). We select them as separate tree nodes (#1 and #2) and determine patterns using feature cards. We find crime patterns in a feature card (Figure 9b1), i.e. more crimes occur in the daytime on weekdays (green line), while more occur at night on weekends (orange line).

We further partition #1 into 24 hour subsets to understand finer crime patterns of weekdays. We focus on subsets of night (19:00-1:00), when crimes occur intensively. We highlight the corresponding subsets, which form 3 clusters in the projection, as in Figure 9c. We select them as three separate tree nodes (#3-#5) and find several interesting patterns. For example, some kinds of crimes (marked with dashed borders in Figure 9c1) occur more often at midnight (0:00) than in the evening (19:00-23:00), while residences are the main locations (marked with dashed borders in Figure 9c2).

We finally partition #3 on Location-type. Each subset thus represents a location-type of weekday evening. The subsets are automatically divided into 5 clusters, as in Figure 9d. An interesting finding is two roadway-related subsets, i.e. sidewalk and street, are in two clusters. By selecting them and visualizing their features, we find they are drastically different on features of crime-type and location. Two criminal types (marked with dashed borders in Figure 9d1) and three locations (marked with dashed borders in Figure 9d2) are related to sidewalk, while crimes occurring on street are more diverse and evenly distributed throughout the city.

Individual Relationship Identification. We project subsets of crime types and those sliced on different temporal attributes (Week and Hour) together. The purpose is to know when specific kinds of crimes intensively occur, as follows:

We first project the 16 crime-type subsets with those sliced on Week, as in Figure 10a. We find weekday subsets and weekend subsets are close to different crime-type subsets and select them as tree nodes. The finding indicates that narcotics and criminal damage crimes occur more often in different periods of week, as in Figure 10a1. Their similar feature values (as in Figure 10a2) demonstrate the correctness of the projection. We also find similar patterns on theft and battery (omitted for limited space).

We partition #12 (weekday) and #14 (weekend) on Hour respectively to form two groups of hour subsets. We project 16 crime-type subsets with each group of hour subsets together, as in Figure 10b-c. We find assault subsets (#15 and #17) in the two projections are adjacent to afternoon subsets (12:00–18:00) and evening subsets (19:00–0:00) respectively. Figure 10b1 proves the correctness of the projection, in which the assault subset (#15) and the selected afternoon subsets (#16) have similar values on a feature. The observation indicates that assault crimes occur more often in the afternoon on weekdays, while assault crimes occur more often in the evening on weekends, as in Figure 10c1.

5.2. Quantitative Experiments

We assess whether SEN can accurately encode single-feature similarity of subsets through two experiments.

5.2.1. Experiment Design

We collect 5 open multi-view datasets for the two experiments, including Handwritten Digits [han], ORL [orl], PIE [pie], Caltech 101-7 [cal] and BBCSport [bbc]. We treat each record of the datasets as a subset, which has multiple features and a label that indicates its category. The records of the same category have similar values on corresponding features.

For each dataset, we replace several features of records with random numbers. This step makes parts of features that are originally similar different, thus simulating the cases that subsets (records) have different numbers of similar features. We train SENs to obtain 30-dimensional emebddings of the records and evaluate the accuracy from the following two aspects:

Pattern Encoding Accuracy. We use K-means to divide the records of each dataset into clusters according to their embeddings. The number of clusters is set to the actual number of categories of the dataset. We quantify the differences between the actual clusters and predicted clusters using three indicators, i.e. **(i)** Accuracy (ACC) **(ii)** Normalized Mutual Information (NMI) **(iii)** Adjusted Rand Index (ARI) (we match a predicted cluster and a actual one if they have a large number of common records). The value ranges of the three indicators are [0, 1], with 0 and 1 representing the lowest and highest accuracy. A larger value indicates a better match of predicted and actual clusters.

Visual Perception Accuracy. We use t-SNE [MH08] to project records according to their embeddings and evaluate the visual separability of records of different categories in the projection. Two

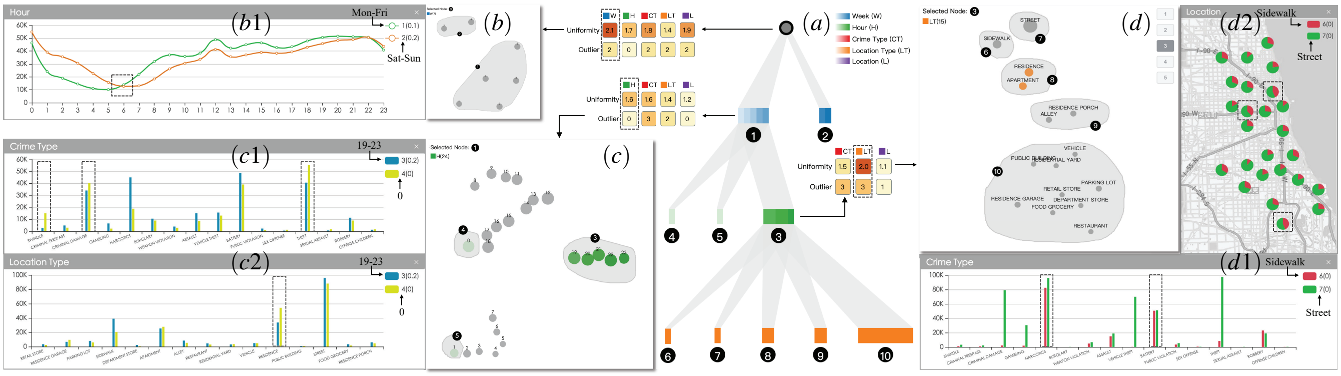


Figure 9: *Explorative process of the first category of tasks. The exploration begins at the root node (a), then we gradually add slicing attributes to generate projections (b-d) and identify patterns (b1-d2).*

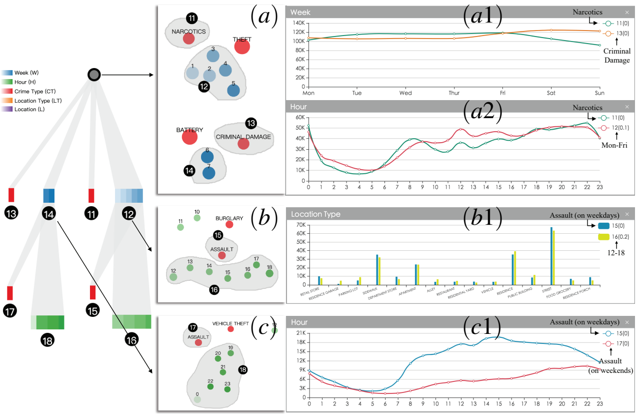


Figure 10: *Explorative process of the second category of tasks. (a) Projecting 16 crime-type subsets and 7 week subsets together. (b-c) Projecting 16 crime-type subsets and 24 hour subsets sliced on weekday and weekend selected in (a) respectively. (a1-c1) Feature cards for determining patterns.*

indicators are used: **(i)** Silhouette Coefficient (SC) and **(ii)** Calinski-Harabasz Index (CHI). A larger value indicates higher separability, i.e. higher visual perception accuracy.

We compare SEN with t-SNE [MH08] and m-SNE [XMTH11]. We connect all features (vectors) of a record as the input of t-SNE, while m-SNE, as a multi-view dimensionality reduction technique, has the same input format as SEN. For pattern encoding accuracy, all the three techniques output 30-dimensional embedding vectors. For visual perception accuracy, t-SNE and m-SNE outputs 2-dimensional vectors for projection directly, while we project 30-dimensional embedding vectors of SEN through dimensionality reduction (consistent with the actual "first embedding then projection" workflow of SEN). We set perplexity to 20 and keep other hyperparameters as default settings for t-SNE and m-SNE.

5.2.2. Experiment 1

We first choose four datasets, replace half of features for each dataset, and calculate indicator values for embeddings obtained using the three techniques. As in Table 1, we find SEN has the highest values in most cases (marked in red), which indicates general higher pattern encoding accuracy and visual perception accuracy.

Table 1: *Experiment results on four datasets with half of features replaced. The highest indicators are marked in red. We repeat the experiment five times, and numbers in parentheses are variances.*

Method	Measure	BBCSport	Caltech-7	ORL	PIE
SEN	ACC	0.719 (0.097)	0.599 (0.059)	0.630 (0.036)	0.470 (0.144)
	NMI	0.535 (0.057)	0.429 (0.065)	0.612 (0.043)	0.439 (0.188)
	ARI	0.497 (0.087)	0.354 (0.068)	0.407 (0.055)	0.231 (0.161)
	SC	0.214 (0.010)	0.034 (0.055)	0.009 (0.039)	-0.078 (0.090)
	CHI	158.788 (31.466)	38.825 (16.290)	7.792 (3.143)	5.117 (3.266)
m-SNE	ACC	0.437 (0.079)	0.537 (0.179)	0.349 (0.009)	0.303 (0.006)
	NMI	0.173 (0.086)	0.395 (0.250)	0.240 (0.045)	0.183 (0.039)
	ARI	0.142 (0.087)	0.320 (0.225)	0.083 (0.007)	0.037 (0.010)
	SC	-0.014 (0.027)	0.064 (0.154)	-0.191 (0.010)	-0.185 (0.012)
	CHI	15.154 (17.449)	48.553 (51.837)	1.285 (0.683)	0.987 (0.233)
t-SNE	ACC	0.562 (0.016)	0.528 (0.097)	0.508 (0.092)	0.346 (0.071)
	NMI	0.374 (0.022)	0.359 (0.131)	0.434 (0.158)	0.342 (0.168)
	ARI	0.308 (0.021)	0.282 (0.119)	0.252 (0.117)	0.122 (0.095)
	SC	0.150 (0.022)	-0.007 (0.104)	-0.074 (0.119)	-0.146 (0.034)
	CHI	112.595 (11.819)	48.904 (27.649)	6.809 (5.213)	1.229 (0.392)

5.2.3. Experiment 2

We conduct the experiment using the Handwritten Digits dataset that contains ten categories. We split each record's 649 attributes into 30 features (29 20-dimensional features and 1 69-dimensional feature). Figure 11 shows the experiment results. We find indicator values decrease as the numbers of replaced features increase. However, the downward trend of SEN (red lines) is slower than those of m-SNE (blue lines) and t-SNE (green lines). Moreover, SEN has higher values on all the five indicators at any number of replaced features.

We further visualize projections obtained in the above experiment, as in Figure 12. Each point represents a record (subset) with the color indicating the category. We find when only a small number of features are replaced, most categories can be identified in the projections (see the two leftmost columns). However, as the number of replaced features increases, many categories are mixed in the projections of m-SNE and t-SNE (see the two rightmost columns in the last two rows). In contrast, we can identify more categories

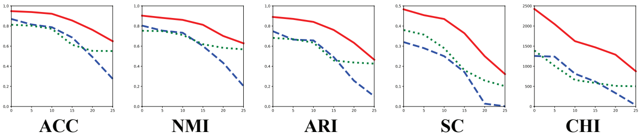
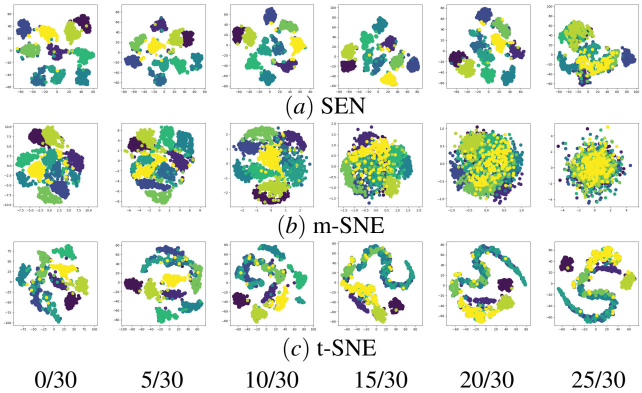


Figure 11: *Indicator values of three techniques, i.e. SEN (red lines), m-SNE (blue lines), tSNE (green lines), under different numbers of replaced features.*

in the projections of SEN, even when few features are retained (see the two rightmost projections in the first row).


(*a*) SEN

(*b*) m-SNE

(*c*) t-SNE

Figure 12: *Projections of Handwritten Digits dataset with 0-25 replaced features. From left to right, numbers of replaced features gradually increase.*

5.3. Efficiency Assessment

We finally test the training speed of SEN using randomly-generated records (subsets). We choose two independent variables, i.e. $|subsets|$ and $|features|$. The experiment is conducted on a GPU server (XEON E5-2680, 196G, 2080Ti).

As in Table 2, the SEN can handle hundreds of subsets with multiple features in a short time. Moreover, the longest training time is 7.5s, which is still an acceptable time cost for most visualization systems. Experiment results show high efficiency of the SEN.

Table 2: *Training time (seconds) of the SEN under different numbers of features and subsets. We repeat the experiment five times and numbers in parentheses are variances.*

$\|features\|$ / $\|subsets\|$	10	15	20	25	30
100	1.9 (0.18)	2.9 (0.25)	4.5 (0.36)	5.8 (0.43)	7.2 (0.52)
300	1.9 (0.25)	3.0 (0.27)	4.7 (0.32)	6.1 (0.41)	7.3 (0.54)
500	2.0 (0.21)	3.1 (0.31)	4.9 (0.41)	6.2 (0.45)	7.5 (0.55)

6. Discussion and Limitations

We discuss two important but easily overlooked aspects, which may affect the application of our approach in actual scenarios.

There is inevitably uncertainty in the projection. To explain this point, consider three subsets A, B, and C. Specifically, A and B have similar values on a feature, A and C have similar values on another one, while B and C are different on all features. In that case, the embedding similarity between B and C is uncertain. They can be similar (as B and C are similar to A on single features) or not (as B and C are completely different). We hope the visualization system can eliminate the uncertainty. The system provides rich interactions and automatic methods to assist in selecting subsets in the projection, and provides feature cards and a consistency measure to visually determine the existence of patterns within selected subsets, thus improving the interpretability and faithfulness of the identified patterns.

As an AI model, the SEN inevitably involves hyperparameters, such as the size of embeddings, the number of neurons in different layers, etc. The black-box nature makes the optimization of these hyperparameters difficult. The positive aspect is we only use ordinary fully-connected neural network with common hyperparameters. The simple structure reduces the tuning difficulty. We only assume developers of the SEN have basic knowledge about neural network. They can always obtain a well-performing model through a small number of trials. The simple network structure and few parameters also achieve fast training.

7. Related Work

We discuss related works from the following two aspects that are related to the two contributions of the paper.

7.1. Data Embedding

DR is the most common embedding technique for multi-dimensional data. PCA [WEG87] projects data along the directions with maximum variance. MDS [BG05] preserves pairwise Euclidean distances during the projection. ISOMAP [TDSL00] changes the Euclidean distance of MDS to geodesic distance, thus enabling to capture nonlinear manifold structures in high-dimensional space. t-SNE [MH08] allows objects that are close in high-dimensional space to be projected together with high probability in low-dimensional projections. Other important DR techniques include LLE [RS00], LE [BN02], LTSA [ZZ04], etc. Many works exist for explaining the DR results [FKM19, LWCC17, FGS18, CMK20]. All these techniques, however, are to maintain global relationships [SPT19] and cannot be used to explore subset patterns that are often associated with few features.

There are many DR techniques with special objective functions [LMW*16, SZS*16, EMK*19]. Wang et al. [WFC*17] propose a DR algorithm that maximizes inter-class distances. Zhang et al. [CZC*15] offer a technique that can reflect the similarity of target objects on both statistic metrics and distributions. Fujiwara et al. [FCS*19] propose a DR technique for streaming data, which can maintain the mental map of analysts for record or dimension changes. Liu et al. [LATB20] propose a response function preserving algorithm that generates projections showing patterns related to single response functions. SEN is of this category with maintaining the similarity of objects on single features as the objective function.

Applying the neural network in data embedding is a recent trend.

Hinton and Salakhutdinov [HS06] implement an autoencoder-based embedding technique that achieves better performance than PCA. Mikolov et al. [MSC*13] propose the famous word2vec that embeds words according to their co-occurrence in a document set. Many similar techniques, such as cite2vec [BMS16], location2vec [ZCX*19], poi2vec [FCAC17], etc. have been proposed. These techniques, however embed objects using their co-occurring frequency in the dataset, unable to be used to explore subset patterns.

The SEN is inspired by multi-view learning, as in Figure 3. Canonical Correlation Analysis (CCA) is a representative technique [Hot36] that can find two linear projections making the multi-view data maximally correlated. We then obtain embeddings using the basis vectors of the two projections. Many techniques extend CCA to capture nonlinear inter-view relationships [Aka06, AABL13]. These methods, however, only support two-view data. Multi-view representation fusion can exploit the complementary information of multiple views to generate the required embeddings [BJ03, CZX10, SS12]. The principle is to determine the posterior probability p(h|x,y) of the probabilistic model p(x,y,z) over the joint space of the shared latent variables z and the observed two-view data x,y. Applying the neural network in multi-view representation learning is a recent trend. Representative examples are multi-modal autoencoder [NKK*11], multi-view convolutional neural network [FPZ16], and multi-modal recurrent neural network [KFF15]. Literature surveys [LYZ18, ZXXS17, JY17] include most recent works. Existing techniques require objects to have the same number of views. They thus cannot be used for arbitrary subsets with different numbers of features.

7.2. Multi-dimensional Data Exploration

Many visualization techniques aim at finding subsets where patterns exist [Shn96]. A common strategy is projecting the data into a plane [JFSK15, LT16, BSH*15, EDF08]. The projection works as an overview, in which analysts manipulate data and filter parts of interest. Many works combine visualization and automatic algorithms to form a generic tool [FSN*20, WCR*17, SZS*16, LPK*15, Gle13]. These methods, however, have done good jobs in searching for subsets where patterns exist, but they cannot find patterns among a group of subsets by considering their feature similarity.

Subspace analysis is a kind of technique to find patterns in dimensional subspaces to overcome the problem of the Curse of Dimensionality [BGRS99]. A common strategy is to design a measure that describes the possibility that patterns exist within a subspace. Ferdosi [FR11] proposes a measure for reordering axes of parallel coordinates to identify high-dimensional structures. Tukey et al. [TT88] propose Scagnostics to identify anomalous scatterplots in the scatterplot matrix (SPLOM). The idea is to reduce the original SPLOM with $O(n^2)$ cells (n is the number of attributes) to a scagnostics SPLOM with $O(k^2)$ cells, where k is the number of measures that describe the distribution of points of an original SPLOM cell. Seo et al. [SS04] propose a rank-by-feature framework, in which users sort views according to a ranking measure. There are many other measures to find views of interest [TAE*09, AEL*10, WMA*15]. They, however, are for evaluating views with 1D or 2D axes, i.e. low-dimensional subspaces.

Many approaches assist in exploring patterns in high-dimensional subspaces. Zhou et al. [ZLH*16] propose a method to preserve clusters by reconstructing dimensions of subspaces. Wang and Mueller [WM17] decompose a high-dimensional space into a series of 3D subspaces to facilitate pattern exploration. Yuan [YRWG13] proposes Dimension Projection Matrix/Tree that enables to explore record- and dimension-related patterns at the same time. Pagliosa et al. [PPM*15] design an interactive tool for comparing different multi-dimensional projections. Xia et al. [XYC*17] design the LTSD-GD view that can represent latent low-dimensional structures within multi-dimensional data. Many methods design measures to evaluate how much insights are provided by a multi-dimensional projection [MMdALO15, LT15, AWD12]. All these methods, however, work at the record level without the ability to find patterns associated with subsets.

Shadoan and Weaver [SW13] propose a visualization system for analyzing relationships between subsets. The relationship, however, reflects the common records between subsets, not as general as SEN encoding arbitrary subset features. Gratzl et al. [GGL*14] propose Domino that supports the flexible exploration of subsets and their relationships. Borland et al. [BWZ*19] propose a visual analytics method to unbiasedly selecting a representative subset for a large dataset. Gotz et al. [GZW*19] propose a method for interactively determining the most informative event subset in a specific analysis context. These methods, however, are different from our approach that aims at providing a generic tool to explore patterns among a large number of subsets.

8. Conclusion and Future Work

We have presented an approach to exploring multi-dimensional data at the subset level. The core of the approach is a subset embedding network that has three characteristics compared with existing embedding techniques. First, it supports arbitrary subsets with different numbers of features. Second, it captures the similarity of subsets on single features. Third, it has high efficiency by using a simple structure with few parameters. The network has been integrated into a visualization system that integrates three components to achieve a flexible and efficient workflow. We present example usage scenarios with real-world data and conduct multiple quantitative experiments to demonstrate the general applicability and effectiveness of our approach.

In the future, we plan to make two improvements. First, we will apply the approach in more fields to thoroughly test its applicability. Second, we will further enrich the functions of the visualization system. For example, we would like to support customizing subset features or integrate more intelligent and automatic techniques to assist in selecting subsets in the projection.

Acknowledgements

This work is supported by National Key R&D Program of China (2018YFC0831700), NSFC project (61972278), Natural Science Foundation of Tianjin (20JCQNJC01620), and the Browser Project (CEIEC-2020-ZM02-0132). This work is also supported by Shanghai Municipal Science and Technology Major Project (2021SHZDZX0103, 2018SHZDZX01) and ZJLab.

References

[AABL13] ANDREW G., ARORA R., BILMES J., LIVESCU K.: Deep canonical correlation analysis. In *Proceedings of the 30th International Conference on Machine Learning* (2013), vol. 28, pp. 1247–1255. 10

[AEL*10] ALBUQUERQUE G., EISEMANN M., LEHMANN D. J., THEISEL H., MAGNOR M.: Improving the visual analysis of high-dimensional datasets using quality measures. In *5th IEEE Conference on Visual Analytics Science and Technology* (2010), pp. 19–26. 10

[Aka06] AKAHO S.: A kernel method for canonical correlation analysis. *CoRR abs/cs/0609071* (2006). 10

[AWD12] ANAND A., WILKINSON L., DANG T. N.: Visual pattern discovery using random projections. In *2012 IEEE Conference on Visual Analytics Science and Technology (VAST)* (2012), IEEE, pp. 43–52. 10

[bbc] BBCSport. http://mlg.ucd.ie/datasets/ Accessed: 2020-09-25. 7

[BCV13] BENGIO Y., COURVILLE A., VINCENT P.: Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 8 (2013), 1798–1828. 1

[BG05] BORG I., GROENEN P. J.: *Modern multidimensional scaling: Theory and applications*. Springer Science & Business Media, 2005. 9

[BGRS99] BEYER K., GOLDSTEIN J., RAMAKRISHNAN R., SHAFT U.: When is "nearest neighbor" meaningful? In *7th International Conference on Database Theory* (1999), Springer, pp. 217–235. 10

[BJ03] BLEI D. M., JORDAN M. I.: Modeling annotated data. In *Proceedings of the 26th International Conference on Research and Development in Information Retrieval* (2003), pp. 127–134. 10

[BMS16] BERGER M., MCDONOUGH K., SEVERSKY L. M.: cite2vec: Citation-driven document exploration via word embeddings. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2016), 691–700. 10

[BN02] BELKIN M., NIYOGI P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems* (2002), pp. 585–591. 9

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D^3 data-driven documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309. 6

[BSH*15] BACH B., SHI C., HEULOT N., MADHYASTHA T., GRABOWSKI T., DRAGICEVIC P.: Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 559–568. 10

[BWZ*19] BORLAND D., WANG W., ZHANG J., SHRESTHA J., GOTZ D.: Selection bias tracking and detailed subset comparison for high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 429–439. 10

[cal] Caltech101-7. http://www.vision.caltech.edu/ImageDatasets/Caltech101/ Accessed: 2020-09-25. 7

[chi] Chicago crime dataset. https://www.kaggle.com/currie32/crimes-in-chicago Accessed: 2020-07-10. 1, 7

[CMK20] CHATZIMPARMPAS A., MARTINS R. M., KERREN A.: t-viSNE: Interactive assessment and interpretation of t-SNE projections. *IEEE Transactions on Visualization and Computer Graphics* (2020). 9

[CZC*15] CHEN H., ZHANG S., CHEN W., MEI H., ZHANG J., MERCER A., LIANG R., QU H.: Uncertainty-aware multidimensional ensemble data visualization and exploration. *IEEE Transactions on Visualization and Computer Graphics 21*, 9 (2015), 1072–1086. 9

[CZX10] CHEN N., ZHU J., XING E. P.: Predictive subspace learning for multi-view data: a large margin approach. In *Advances in Neural Information Processing Systems* (2010), pp. 361–369. 10

[EDF08] ELMQVIST N., DRAGICEVIC P., FEKETE J.-D.: Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1539–1148. 10

[EMK*19] ESPADOTO M., MARTINS R. M., KERREN A., HIRATA N. S., TELEA A. C.: Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics* (2019). 9

[FCAC17] FENG S., CONG G., AN B., CHEE Y. M.: Poi2vec: Geographical latent representation for predicting future visitors. In *Proceedings of the Thirty-First Conference on Artificial Intelligence* (2017), pp. 102–108. 10

[FCS*19] FUJIWARA T., CHOU J.-K., SHILPIKA S., XU P., REN L., MA K.-L.: An incremental dimensionality reduction method for visualizing streaming multidimensional data. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 418–428. 9

[FGS18] FAUST R., GLICKENSTEIN D., SCHEIDEGGER C.: Dimreader: Axis lines that explain non-linear projections. *IEEE Transactions on Visualization and Computer Graphics 25*, 1 (2018), 481–490. 9

[FKM19] FUJIWARA T., KWON O.-H., MA K.-L.: Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 45–55. 9

[FPZ16] FEICHTENHOFER C., PINZ A., ZISSERMAN A.: Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 1933–1941. 10

[FR11] FERDOSI B. J., ROERDINK J. B.: Visualizing high-dimensional structures by dimension ordering and filtering using subspace analysis. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1121–1130. 10

[FSN*20] FUJIWARA T., SAKAMOTO N., NONAKA J., YAMAMOTO K., MA K.-L.: A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction. 10

[GGL*14] GRATZL S., GEHLENBORG N., LEX A., PFISTER H., STREIT M.: Domino: Extracting, comparing, and manipulating subsets across multiple tabular datasets. *IEEE Transactions on Visualization and Computer Graphics 20*, 12 (2014), 2023–2032. 10

[Gle13] GLEICHER M.: Explainers: Expert explorations with crafted projections. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2042–2051. 10

[GZW*19] GOTZ D., ZHANG J., WANG W., SHRESTHA J., BORLAND D.: Visual analysis of high-dimensional event sequence data via dynamic hierarchical aggregation. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 440–450. 10

[han] Handwritten Digits. https://archive.ics.uci.edu/ml/datasets/Multiple+Features Accessed: 2020-09-20. 7

[Hot36] HOTELLING H.: Relations between two sets of variates. *Biometrika 28* (1936), 321–377. 10

[HS06] HINTON G. E., SALAKHUTDINOV R. R.: Reducing the dimensionality of data with neural networks. *Science 313*, 5786 (2006), 504–507. 10

[JFSK15] JÄCKLE D., FISCHER F., SCHRECK T., KEIM D. A.: Temporal mds plots for analysis of multivariate data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 141–150. 10

[JY17] JINGJING T., YINGJIE T.: A survey on multi-view learning. *Mathematical Modeling and Its Applications* (2017). 3, 10

[KFF15] KARPATHY A., FEI-FEI L.: Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 3128–3137. 10

[LATB20] LIU S., ANIRUDH R., THIAGARAJAN J. J., BREMER P.-T.: Uncovering interpretable relationships in high-dimensional scientific data through function preserving projections. *Machine Learning: Science and Technology* (2020). 9

[LMW*16] LIU S., MALJOVEC D., WANG B., BREMER P.-T., PASCUCCI V.: Visualizing high-dimensional data: Advances in the past decade. *IEEE Transactions on Visualization and Computer Graphics 23*, 3 (2016), 1249–1268. 9

[LPK*15] LOORAK M. H., PERIN C., KAMAL N., HILL M., CARPENDALE S.: Timespan: Using visualization to explore temporal multi-dimensional data of stroke patients. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 409–418. 10

[LT15] LEHMANN D. J., THEISEL H.: Optimal sets of projections of high-dimensional data. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 609–618. 10

[LT16] LEHMANN D. J., THEISEL H.: General projective maps for multidimensional data projection. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 443–453. 10

[LWCC17] LIAO H., WU Y., CHEN L., CHEN W.: Cluster-based visual abstraction for multivariate scatterplots. *IEEE Transactions on Visualization and Computer Graphics 24*, 9 (2017), 2531–2545. 9

[LYZ18] LI Y., YANG M., ZHANG Z.: A survey of multi-view representation learning. *IEEE Transactions on Knowledge and Data Engineering 31*, 10 (2018), 1863–1883. 2, 3, 10

[MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-SNE. *Journal of Machine Learning Research 9*, Nov (2008), 2579–2605. 2, 6, 7, 8, 9

[MMdALO15] MOTTA R., MINGHIM R., DE ANDRADE LOPES A., OLIVEIRA M. C. F.: Graph-based measures to assist user assessment of multidimensional projections. *Neurocomputing 150* (2015), 583–598. 10

[MSC*13] MIKOLOV T., SUTSKEVER I., CHEN K., CORRADO G. S., DEAN J.: Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems* (2013), pp. 3111–3119. 10

[NKK*11] NGIAM J., KHOSLA A., KIM M., NAM J., LEE H., NG A. Y.: Multimodal deep learning. In *Proceedings of the 28th International Conference on Machine Learning* (2011), pp. 689–696. 10

[orl] ORL. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html Accessed: 2020-09-25. 7

[pie] PIE. http://www.cs.cmu.edu/afs/cs/project/PIE/MultiPie/Multi-Pie/Home.html Accessed: 2020-09-25. 7

[PPM*15] PAGLIOSA P., PAULOVICH F. V., MINGHIM R., LEVKOWITZ H., NONATO L. G.: Projection inspector: Assessment and synthesis of multidimensional projections. *Neurocomputing 150* (2015), 599–610. 1, 10

[RS00] ROWEIS S. T., SAUL L. K.: Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323–2326. 9

[Shn96] SHNEIDERMAN B.: The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages* (1996), IEEE, pp. 336–343. 2, 4, 10

[SPT19] SPATHIS D., PASSALIS N., TEFAS A.: Interactive dimensionality reduction using similarity projections. *Knowledge-Based Systems 165* (2019), 77–91. 9

[SS04] SEO J., SHNEIDERMAN B.: A rank-by-feature framework for unsupervised multidimensional data exploration using low dimensional projections. In *IEEE Symposium on Information Visualization* (2004), IEEE, pp. 65–72. 6, 10

[SS12] SRIVASTAVA N., SALAKHUTDINOV R. R.: Multimodal learning with deep boltzmann machines. In *Advances in Neural Information Processing Systems* (2012), pp. 2222–2230. 10

[SW13] SHADOAN R., WEAVER C.: Visual analysis of higher-order conjunctive relationships in multidimensional data using a hypergraph query system. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2070–2079. 10

[SZS*16] SACHA D., ZHANG L., SEDLMAIR M., LEE J. A., PELTONEN J., WEISKOPF D., NORTH S. C., KEIM D. A.: Visual interaction with dimensionality reduction: A structured literature analysis. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2016), 241–250. 9, 10

[TAE*09] TATU A., ALBUQUERQUE G., EISEMANN M., SCHNEIDEWIND J., THEISEL H., MAGNORK M., KEIM D.: Combining automated analysis and visualization techniques for effective exploration of high-dimensional data. In *IEEE Symposium on Visual Analytics Science and Technology* (2009), IEEE, pp. 59–66. 10

[TDSL00] TENENBAUM J. B., DE SILVA V., LANGFORD J. C.: A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323. 9

[TT88] TUKEY J. W., TUKEY P. A.: Computer graphics and exploratory data analysis: An introduction. *The Collected Works of John W. Tukey: Graphics: 1965-1985 5* (1988), 419. 10

[WCR*17] WENSKOVITCH J., CRANDELL I., RAMAKRISHNAN N., HOUSE L., NORTH C.: Towards a systematic combination of dimension reduction and clustering in visual analytics. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 131–141. 10

[WEG87] WOLD S., ESBENSEN K., GELADI P.: Principal component analysis. *Chemometrics and Intelligent Laboratory Systems 2*, 1-3 (1987), 37–52. 9

[WFC*17] WANG Y., FENG K., CHU X., ZHANG J., FU C.-W., SEDLMAIR M., YU X., CHEN B.: A perception-driven approach to supervised dimensionality reduction for visualization. *IEEE Transactions on Visualization and Computer Graphics 24*, 5 (2017), 1828–1840. 9

[WM17] WANG B., MUELLER K.: The subspace voyager: exploring high-dimensional data along a continuum of salient 3d subspaces. *IEEE Transactions on Visualization and Computer Graphics 24*, 2 (2017), 1204–1222. 10

[WMA*15] WONGSUPHASAWAT K., MORITZ D., ANAND A., MACKINLAY J., HOWE B., HEER J.: Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 649–658. 10

[XMTH11] XIE B., MU Y., TAO D., HUANG K.: m-SNE: Multiview stochastic neighbor embedding. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 41*, 4 (2011), 1088–1096. 2, 8

[XYC*17] XIA J., YE F., CHEN W., WANG Y., CHEN W., MA Y., TUNG A. K.: Ldsscanner: Exploratory analysis of low-dimensional structures in high-dimensional datasets. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2017), 236–245. 1, 10

[YRWG13] YUAN X., REN D., WANG Z., GUO C.: Dimension projection matrix/tree: Interactive subspace visual exploration and analysis of high dimensional data. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2625–2633. 1, 10

[ZCX*19] ZHU M., CHEN W., XIA J., MA Y., ZHANG Y., LUO Y., HUANG Z., LIU L.: Location2vec: a situation-aware representation for visual exploration of urban locations. *IEEE Transactions on Intelligent Transportation Systems 20*, 10 (2019), 3981–3990. 10

[ZLH*16] ZHOU F., LI J., HUANG W., ZHAO Y., YUAN X., LIANG X., SHI Y.: Dimension reconstruction for visual exploration of subspace clusters in high-dimensional data. In *2016 IEEE Pacific Visualization Symposium (PacificVis)* (2016), IEEE, pp. 128–135. 10

[ZXXS17] ZHAO J., XIE X., XU X., SUN S.: Multi-view learning overview: Recent progress and new challenges. *Information Fusion 38* (2017), 43–54. 3, 10

[ZZ04] ZHANG Z., ZHA H.: Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM Journal on Scientific Computing 26*, 1 (2004), 313–338. 9