MultiProjector: Temporal Projection for Multivariates Time Series

Tommy Dang(✉) and Ngan V. T. Nguyen

Texas Tech University, Lubbock, TX 79409, USA
{tommy.dang,ngan.v.t.nguyen}@ttu.edu

Abstract. This paper applies the recent advances of visual analytics, which combine computers' and humans' strengths to the data exploration process, to alleviate the scalability and overplotting issues of dimensional projection techniques for high-dimensional temporal datasets. Our approach first uses clustering algorithms to select the representative data points at each time step for each data profile. We then apply dimension reduction techniques to visualize the temporal relationships via connecting lines. Finally, we propose a couple of different underlying models to treat time steps and the time dimension to mitigate the final projections' visual clutter. We built a web-based prototype, called *MultiProjector*, to integrate these components into a unified data exploration process. The prototype is validated on several high-dimensional temporal datasets in various application domains to demonstrate our approach's benefits.

Keywords: HPC monitoring · Projections · Graph visualization

1 Introduction

Temporal datasets are increasing in size and complexity due to the growth of many fields such as scientific applications, economics, and finance. A time series is a chronological collection of observations throughout time [10]. Temporal datasets may have one variable (univariate time series) or many variables (multivariate time series). The latter is more complicated in terms of the analysis as relations between variables play a fundamental role in analyzing this type of time series [27]. An example of the multivariate time series is the US employment data. The monthly statistics of employees in various economic sectors (such as Education, Finance, or Construction) form a multivariate time series collection. In this example, each sector is a variable, and the state is an individual observation. In this paper, we consider the temporal dependencies between variables and inter-relationships between individuals over time.

There are many efforts to integrate temporal information into common visual presentations of cross-sectional datasets, or high-dimensional non-temporal datasets, such as parallel coordinates [8,14], radar charts [26], and hierarchical layouts [13]. Ali et al. [2] introduce the application of sliding window and dimension reduction techniques in visualizing long multivariate time series. Their approach helps to display the similarity of chronological sliding windows of the

G. Bebis et al. (Eds.): ISVC 2022, LNCS 13598, pp. 91–102, 2022.
https://doi.org/10.1007/978-3-031-20713-6_7

multivariate time series, enabling the detections of repetitive patterns or interesting anomalies. This paper considers each instance in the multivariate time series as a data point in the high-dimensional space. Similar data points are grouped based on their multivariate values to provide a compressed summary of the data profile. The projected positions of the remaining data points represent the interrelationships of individuals and the evolution of these individuals via connecting lines. By marrying clustering methods and dimension reduction techniques into a unified framework, we provide scalable multidimensional projections for large temporal data. The contribution of this paper is listed as the following.

- We discuss, compare and summarize the pros and cons of various dimensional reduction techniques in the context of temporal data.
- We propose a couple of different underlying models to treat time steps and the time dimension to reduce the number of projected data points without affecting the global structure and mitigate the final projections' overplotting issues.
- We implement an interactive web-based prototype to visualize high-dimensional temporal datasets. Our approach and prototype are demonstrated on real-world datasets in various domains to illustrate its benefits.

2 Related Work

2.1 Visualizing High Dimensional Temporal Datasets

Many works have been carried out to provide visualizations for high-dimensional time series. Specifically, there are many efforts to add time dimensions into common visual presentations of cross-sectional datasets, or high-dimensional non-temporal datasets, such as matrix [3], parallel coordinates [5], and circular layouts [9]. We firstly consider the temporal extension of the scatterplot. TimeSeer [6] transforms the collection of time series in the datasets into time series of Scagnostics, which are metrics for visual features of the scatterplots for each pair of variables. It uses these Scagnostics as a signal to identify unusual events. Congnostics [22] proposes a list of eight metrics for connected scatterplots' visual features and helps to visualize the dynamic correlation between variables of an individual.

TimeCluster [2] proposes the use of dimension reduction techniques to visualize long multivariate time series. It considers each sliding window as a point in a high-dimensional space, whose number of dimensions equals the time series values in the window. For example, an individual has three variables, and the sliding window has a size of sixty. In this case, the high-dimensional space has 180 dimensions. After reducing the dimensions by deep convolutional auto-encoder, the authors continue to apply other dimension reduction methods such as PCA [31], t-SNE [19], and UMAP [20]. Their approach helps to reconstruct the whole temporal dataset to only one view to observe some interesting patterns like clusters or abnormalities.

2.2 Dimension Reduction

Principal Component Analysis, or PCA, is one of the most popular linear dimension reduction techniques. It projects the original data to a lower-dimensional space, such that the variance of the projected data is maximized [31]. In addition to the linear projections, many nonlinear dimension reduction techniques have been developed. The t-Distributed Stochastic Neighbor Embedding, or t-SNE, is a frequently used nonlinear projection. It computes the similarities between data in the high-dimensional space by Gaussian distribution before reconstructing these similarities by Student t-distribution in a low dimensional space [19]. This method requires both time and memory complexity up to $O(N^2)$, which may not be efficient for large datasets. The acceleration of this technique using the Barnes-Hut algorithm can reduce the time complexity to $O(Nlog(N))$ and the memory complexity to $O(N)$ [29]. Uniform Manifold Approximation and Projection, or UMAP, is recently introduced to the literature [20]. It has been proved to be comparable to t-SNE in the visualization of large datasets. Becht et al. [4] provide a comparison for the running times of some popular projection methods, including t-SNE and UMAP. To stabilize the projection results for streaming multidimensional data, Fujiwara et al. [12] propose geometric transformation and animation methods. However, the approach does not aim to resolve the scalability issues of the multidimensional projection techniques [11]. This paper utilizes and expands the three projection methods mentioned in this section to various multivariate temporal datasets. We will discuss in detail our visual methodology in the next section.

3 Methodology

Our research problem is projected onto the three dimensions: individual data entries, variables of these individuals, and time. An example of this data structure is the monthly US employment rates. This dataset has 53 states and territories in the US as 53 individuals. Each state has many economic sectors such as Good Producing, Manufacturing, Financial Activities, etc., and they are considered the variables of each individual. The net change in the number of employees per month of a specific sector of a particular state form a time series in this collection. Before any computations and visualizations, we apply the min-max normalization for every variable in the dataset to scale them to the unit range.

3.1 Clusterings

To handle large multivariate time series, not all data points join the dimension reduction computation. Instead, we first perform clustering across all snapshots to abstract a large number of data points into the major groups and focus on data instances at the group changes. Our approach is based on the observation that stable profiles may not contain much insight when analyzing time series, but they consume the computational resources for rendering the projections and

causing overplotting issues. In particular, our *MultiProjector* web-based prototype supports two clustering algorithms: *k-means* and *leader bin*. The former requires a given number of groups and a convergence criterion such as the minimal decrease in squared error [15]. Users can also set the maximum number of iterations to stop the *k-means* computation. The latter allows a flexible range of leaders with a consideration: it is inefficient if there are too many leaders, while it tends to over-summarize the dataset if there are too few ones [7]. *MultiProjector* uses *leader bin* as the default multivariate clustering method since it provides the representative instances (leaders) and more stable clustering outcomes.

3.2 Multidimensional Projections

We consider three popular classes: PCA, t-SNE, and UMAP. PCA projects data points into a few orthogonal or uncorrelated principal components, which retain the whole data maximum variance. Usually, the first two components retain most information about the dataset, so it is reasonable to use PCA to project the data points in high-dimensional space to two-dimensional space. However, this method has two main disadvantages [30]. The first one is that it is inappropriate for embedding extremely high-dimensional space due to the overlapping problem or the curse of dimensionality. The second drawback is that it favors the large pairwise distances, not the small ones. The nonlinear methods (t-SNE vs. UMAP) can avoid the overlapping issue of distinct clusters. While t-SNE focuses on preserving the local structure of the dataset, UMAP can reconstruct the global structure.

3.3 Visualizing the Time Dimension

A straightforward approach for plotting temporal domain is using the connected lines. To enforce the time dimension in the computation, we integrate time as a new dimension (increasing from min to max) along with variables for computing the projection. This method allows time to contribute to the projection of data points and to distinguish any individual at different time points. Additionally, we introduce the use of the third axis along with the 2D space to display time. In other words, this approach projects all individuals at the same time point into a 2D layer before aligning them onto the layers in chronological order on the third axis to illustrate the temporal evolution. This third dimension enforces the contribution of time to the final projection of the dataset. The summary of the idea of integrating the time domain into the 2D projection is depicted in Fig. 1.

3.4 Multivariate Representations

Each individual at a specific time point is defined by its multivariate metrics. As we aim to plot the multivariate metrics directly on the projected space, circular representations are more appropriate for a large number of variables [21]. An intuitive presentation for an individual at a time point is a radar chart that

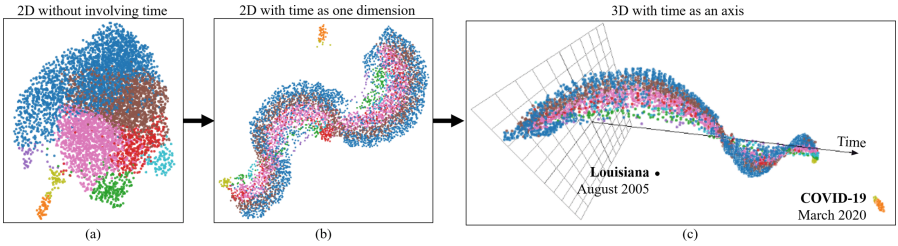

Fig. 1. Visualizing the US monthly employment data in 22 years: (a) 2D UMAP projection (b) 2D UPMAP projection considering time as an additional variable in the multidimensional project, and (c) Integrating the time domain into the 3D projection.

shows its multivariate values [17]. The position of each data point is determined by its multivariate values. Then, the Euclidean distance between any pair of data points measures how similar they are. Before applying projections, we reduce the number of input data points by compressing similar timestamps of the same data profile together. In other words, we care about the changes while discarding the static points in the high-dimensional time series data.

4 Use Cases

4.1 Use Case 1: Monthly US Employment Rate

The US employment dataset contains 53 states and territories [1]. Each state is considered as an individual profile that is recorded on 15 economic sectors. In particular, the monthly net change of the number of employees in every economic sector of each state is retrieved from January 1999 to May 2020. Totally, there are 12,495 data points in this dataset to be considered in the final projection.

In this use case, we focus on the 2D UMAP projection and its 3D variances, as depicted in Fig. 1. Different from the incremental approach discussed by Fujiwara et al. [12], we consider data points in all time steps as a whole in the projection. This allows us to avoid the unstable layouts (such as flipped or rotated) generated by independent projections for each time step. Figure 1(b) depicts the chronological sequence when we consider time as an additional variable for the UMAP projection. In Fig. 1(c), time is used as the third axis (from left to right), the 53 multivariate data points representing the economic status of states and territories in a given month are scattered on a plane orthogonal to the time axis. We can easily notice the interesting spiral pattern from the point of view of how the points are arranged throughout the 3D space in Fig. 1(c). This can be explained as the US economy is completing a circle after the 2008 Great recession. The orange points at the rear of the spiral region are states in March 2020. These points are most dissimilar to most of the points in the spiral region, which means the US experienced a significant drop in the number of employees in March 2020 when Covid-19 started wreaking havoc on the US

economy. Moreover, the outlier below the Spiral represents the Louisiana economy in August 2005 due to hurricane Katrina. In this use case, the data points are color-coded by the k-means clusters that they belong to. The cluster colors are only there for visual inspection and have no impact on the actual projection.

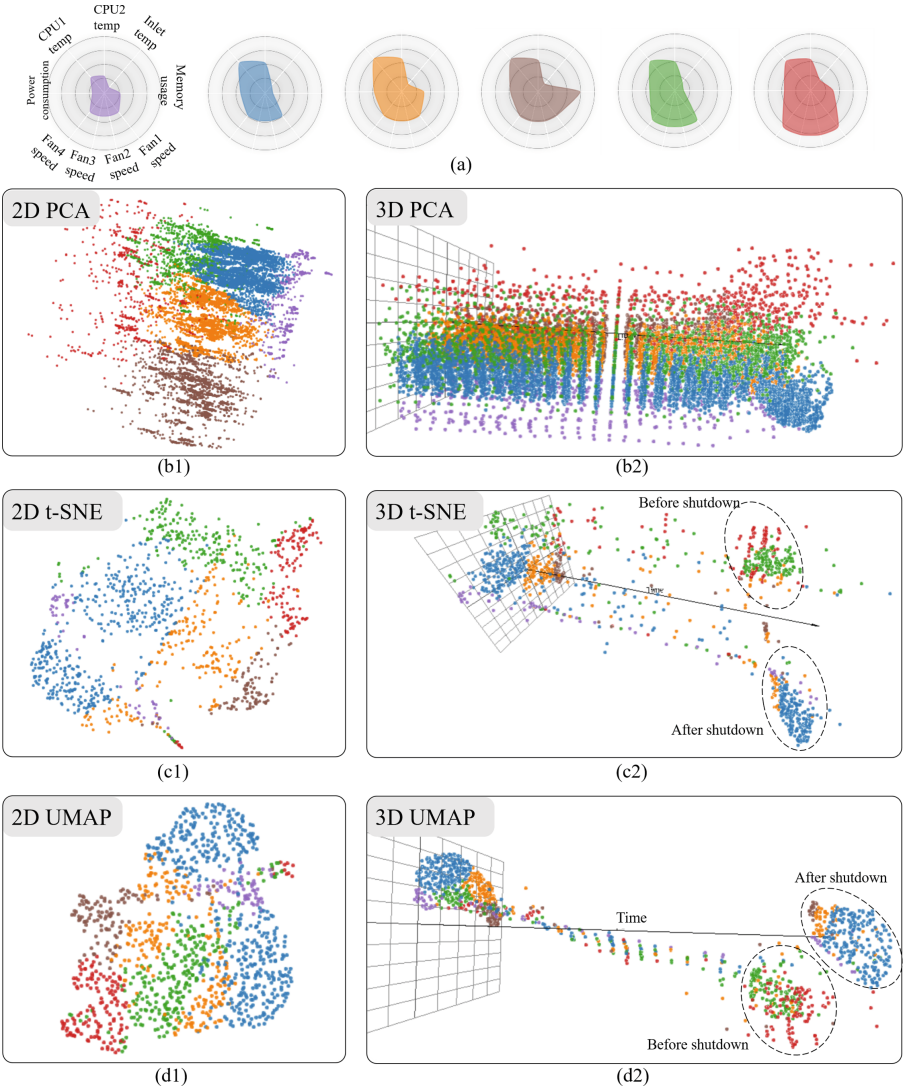

Fig. 2. Multidimensional projections of the computer health metrics: (a) The multivariate data is first classified into six groups. (b) PCA projection of 12,609 operating statuses, (c) t-SNE, and (d) UMAP projection of 1,225 operating statuses. The data points are colored by their multivariate statuses as defined in (a) (Color figure online).

4.2 Use Case 2: Monitoring Computer Metrics

The second use case considers the health metrics of a High-Performance Computing system at a university [28]. The system has 467 nodes, and thus they are 467 individuals in the high dimensional time series associating to nine health metrics, such as CPU temperatures, fans speeds, memory usage, and power consumption [18]. In other words, they are nine variables in the temporal dataset. The metrics are recorded at 5 min frequency. In particular, the dataset that we use in this use case is on March 21, 2019.

The multivariate operating statuses of computing nodes in the High-Performance Computing system are first classified into six major groups using the k-means algorithm. Users can select different clustering methods as well as the number of clusters on their choices. As depicted in Fig. 2(a), radar charts are used to represent the multivariate status of the computing nodes as they can quickly capture the *morphology* of the computing statuses [17]. The PCA projection in Fig. 2(b) takes 876 ms. The PCA projections are pretty uniform, and no visual pattern can be easily discerned.

Based on the observation that system administrators care more about the significant changes rather than the static computing nodes [23], we propose to reduce the number of static operating statuses and only focus on the dynamic behaviors of the system (when the group switchings happen). Therefore, we reduce the number of multivariate data points ten times from 12,609 down to 1,225. This allows our approach scaling well with the large time-dependent multivariate datasets. Figure 2(c) shows 2D t-SNE projection and our modified 3D temporal projection. Notice that the 3D projection, with time as the third axis, displays the three dense regions at the beginning and the end of the observed period. The first region on the grids is the first time step, and therefore, the operating statuses of all 476 computing nodes are recorded. The middle region is sparse since we only plot the significant changes on the metrics, such as CPU and memory usage, most probably associating to the HPC scheduler events (a new user is allocated the computing resources or a new job is dispatched). Toward the end of the observed period, there are separated into two groups: green and red vs. orange and blue. As shown in the radars in Fig. 2(a), the green and red groups have high *CPU temperatures* and high *fan speeds* while the orange and blue groups are normal operating statuses. In particular, the chill water for the HPC center was accidentally disconnected at around 2 pm on March 21, 2019, leading to the overheat issues on all computing nodes (green and red nodes). At 4 pm, the system had been automatically shut down and then returned to the normal operations (orange and blue groups). Regarding UMAP in Fig. 2(d), the 2D projection is quite uniform and has no visible cluster or outlier. In the 3D UMAP projection, the similar dynamic behaviors of the system are also captured on the temporal domain. We can also notice that our data reduction technique has also mitigated the serious overplotting issues in Fig. 2(b2). Our *MultiProjector* also supports embedding the multidimensional representation of computing nodes directly in the projection for visual inspections.

Our *MultiProjector* also supports embedding the multidimensional representation of computing nodes directly in the projection for visual inspections. Figure 3 depicts the same example in Fig. 2(d1) in a compressed honeycomb layout. In particular, *MultiProjector* initializes a force layout from the UMAP configuration. The data points automatically resolve collisions before projected onto a regular honeycomb layout. Specifically, each bee cell in Fig. 3 contains a representative operating status of a node. The saturation of the radar indicates how long the computing node stays on that status (no significant changes on the health metrics). In this example, we draw a trajectory of a sample profile, *compute-3-41*. We can visualize the chill water impacts on this computing node: The node started with the normal operating status at 14:00, then traveled through overheat states in green and red at 15:45 and 15:50, and finally ended up with a blue state after the HPC system reset at 16:00.

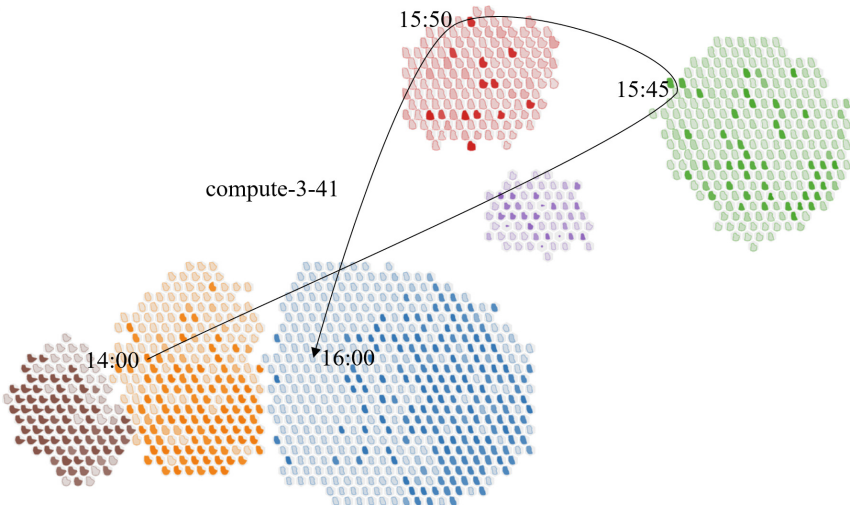

Fig. 3. Visualizing 1,225 operating statuses in our non-overlapped honeycomb layout: The six color-coded clusters are produced by the k-means algorithm on nine health metrics, such as *CPU temperatures*, *memory usage*, and *fan speeds*. The arrow connects various operational status of compute-3-41 in 2 h.

4.3 Use Case 3: Plant Genetics

In this use case, we target the visual clutter issue of multidimensional projections. The data was retrieved from the *Center for Functional Genomics of Abiotic Stress* [16]. In particular, we need to consider 20,450 plant genes experimented under 12 tested conditions, with *STOP1* mutant for the last 6 conditions. These experimented conditions are abbreviated as *wt* for wild type, *stop1* for knock-out mutant background for the transcription factor, *hp* for high phosphate supply (1 mM), *lp* for low phosphate supply (0 mM), *Al* for Al stress pH 5, and *Fe*

for Fe excess supplied to the medium pH 5. For example, nametags for the conditions composed as *wthp6* means wild-type/high Pi supply/pH 6 and *s1hp6* means stop1 ko/high Pi supply/pH 6. *Al* and *Fe* are only tested conditions under low Pi and pH 5, and hence there are two library replicates for Al and Fe for each genotype and toxicity. In the input data, the first column contains gene names, and the next six columns are the wild type conditions, including the base condition, *wthp6*. The last six columns are the corresponding *STOP1* mutant conditions.

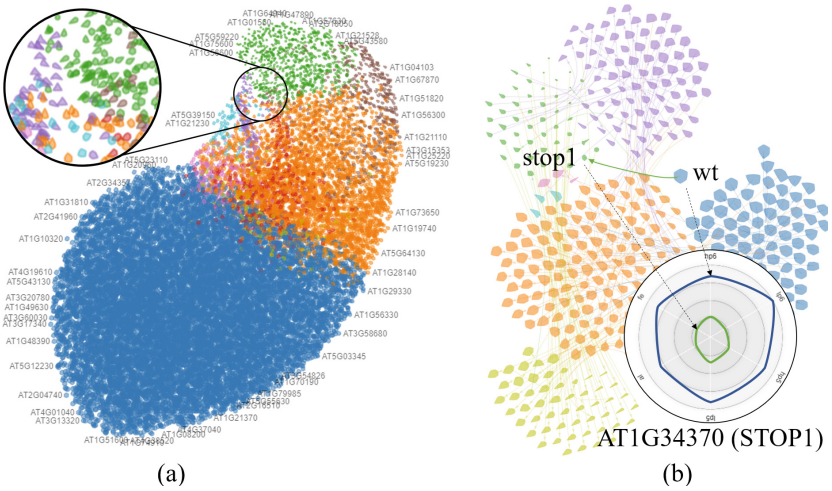

Fig. 4. Visualizing gene expressions using our *MultiProjector* prototype: (a) 20,450 plant genes (b) 210 transcription factors (Color figure online).

Figure 4 shows the expression levels of 20,450 genes under six controlled conditions through two time steps: before and after the application of *STOP1* mutant. Therefore, we have 40,900 data points in this projection. Figure 4(a) shows overplotting issue of 2D UMAP projection. Notice that low expressed genes tend to locate on the top while highly-expressed genes flow down the bottom (the blue region). To alleviate the visual clutter issue, we first reduce the number of projected genes by focusing on transcription factors (the genes that change their expression behaviors significantly), which are identified by the Euclidean distance of the multivariate values before vs. after the injection of *STOP1* mutant. Figure 4(b) shows our non-overlapped honeycomb layout of the 210 transcription factors. The arrows in the background highlight the group transitions of these 210 genes. We can notice the major group changes are between green to yellow and purple to orange. We have annotated the special gene *STOP1* and its rare transition from the most active group (blue) to an inactive one (green), as depicted in the enlarged radar view. In this example, *MultiProjector* provided a compressed projection view of gene expression data that allows

biologists to visualize and identify the behaviors of the leading factors under the tested conditions. This type of analysis is important for plant treatments and drug designs.

4.4 Discussion

PCA is a linear projection and hence is the fastest method with about one second for thousands of data points in the web-based environment. However, it has an issue of overlapping data points, especially when there are outliers. UMAP preserves pairwise Euclidean distances significantly better than t-SNE [25], and thus UMAP preserves more of the global structure. It runs much faster than another nonlinear method, the t-SNE, especially as the size of data points is significantly large. Because t-SNE focuses on reconstructing the dataset's local structure, it cannot perform well in clustering data points for finding dissimilar groups [24]. The same groups' points tend to pull each other, so the density of the t-SNE projection may not be uniform. Figure 5 gives a comparison between UMAP and t-SNE in terms of running time (in *log* scale) via our web-based prototype. All tests were performed on a computer with 2.9 GHz Intel Core i5, macOS Sierra Version 10.12.1, 8 GB RAM. The introduction video and online demo of our web-based prototype can be accessed at https://git.io/JLppG.

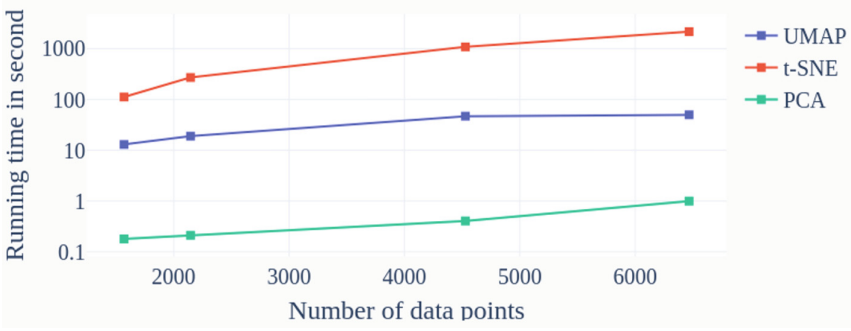

Fig. 5. Running time comparisons of PCA, UMAP, and t-SNE in our web-based application using Google Chrome.

5 Conclusion

Multidimensional projections are popular methods for reducing high-dimensional data onto lower-dimensional planes. However, the importance of the time element is not always considered properly. In this paper, we investigate the temporal domain as one of the dimensions in multidimensional projections. This allows us to impose the temporal changes onto the lower-dimensional space (such as 2D or 3D). We project different time steps as a whole and align them over

the 3rd axis in order to keep the spatial coherence between them. To project a large number of input data points, we focused on the significant time steps for each data profile where multivariate variances occur. Our temporal data reduction technique also helps to mitigate the overplotting issues generated by multidimensional projections. We experiment our approach on various existing dimensional reduction methods and demonstrate them on different domains.

References

1. U.S. bureau of labor statistics databases. http://www.bls.gov/data/. Accessed 08 Jan 2021
2. Ali, M., Jones, M.W., Xie, X., Williams, M.: TimeCluster: dimension reduction applied to temporal data for visual analytics. Vis. Comput. **35**(6), 1013–1026 (2019). https://doi.org/10.1007/s00371-019-01673-y
3. Bach, B., Pietriga, E., Fekete, J.D.: Visualizing dynamic networks with matrix cubes. In: Proceedings of ACM Conference on Human Factors in Computing Systems, pp. 877–886 (2014)
4. Becht, E., et al.: Dimensionality reduction for visualizing single-cell data using UMAP (2019)
5. Burch, M., Vehlow, C., Beck, F., Diehl, S., Weiskopf, D.: Parallel edge splatting for scalable dynamic graph visualization (2011). https://doi.org/10.1109/TVCG.2011.226
6. Dang, T.N., Anand, A., Wilkinson, L.: Timeseer: scagnostics for high-dimensional time series (2012)
7. Dang, T.N., Wilkinson, L.: ScagExplorer: exploring scatterplots by their scagnostics (2014). https://doi.org/10.1109/PacificVis.2014.42
8. Dasgupta, A., Kosara, R., Gosink, L.: Meta parallel coordinates for visualizing features in large, high-dimensional, time-varying data. In: IEEE Symposium on Large Data Analysis and Visualization (LDAV), pp. 85–89. IEEE (2012)
9. Fischer, F., Fuchs, J., Mansmann, F.: ClockMap: enhancing circular treemaps with temporal glyphs for time-series data. In: Meyer, M., Weinkaufs, T. (eds.) EuroVis - Short Papers (2012). https://doi.org/10.2312/PE/EuroVisShort/EuroVisShort2012/097-101
10. Fu, T.C.: A review on time series data mining (2011)
11. Fujiwara, T., Kwon, O.H., Ma, K.L.: Supporting analysis of dimensionality reduction results with contrastive learning (2020). https://doi.org/10.1109/TVCG.2019.2934251
12. Fujiwara, T., Chou, J.K., Shilpika, Xu, P., Ren, L., Ma, K.L.: An incremental dimensionality reduction method for visualizing streaming multidimensional data (2020). https://doi.org/10.1109/tvcg.2019.2934433
13. Greilich, M., Burch, M., Diehl, S.: Visualizing the evolution of compound digraphs with timearctrees. In: Proceedings of Eurographics Conference on Visualization, pp. 975–990 (2009). https://doi.org/10.1111/j.1467-8659.2009.01451.x
14. Gruendl, H., Riehmann, P., Pausch, Y., Froehlich, B.: Time-series plots integrated in parallel-coordinates displays. In: Computer Graphics Forum, vol. 35, pp. 321–330. Wiley Online Library (2016)
15. Hartigan, J.A.: Clustering Algorithms, 99th edn. Wiley, New York (1975)
16. Herrera-Estrella, L.: My journey into the birth of plant transgenesis and its impact on modern plant biology (2020). https://doi.org/10.1111/pbi.13319,, https://onlinelibrary.wiley.com/doi/abs/10.1111/pbi.13319

17. Kammer, D., et al.: Glyphboard: visual exploration of high-dimensional data combining glyphs with dimensionality reduction (2020). https://doi.org/10.1109/TVCG.2020.2969060
18. Li, J., et al.: Monster: an out-of-the-box monitoring tool for high performance computing systems. In: 2020 IEEE International Conference on Cluster Computing (CLUSTER), pp. 119–129 (2020). https://doi.org/10.1109/CLUSTER49012.2020.00022
19. Maaten, L.v.d., Hinton, G.: Visualizing data using t-SNE (2008)
20. McInnes, L., Healy, J., Melville, J.: UMAP: uniform manifold approximation and projection for dimension reduction (2018)
21. Meyer, M., Munzner, T., Pfister, H.: MizBee: a multiscale synteny browser (2009)
22. Nguyen, B.D.Q., Hewett, R., Dang, T.: Congnostics: visual features for doubly time series plots. In: Turkay, C., Vrotsou, K. (eds.) EuroVis Workshop on Visual Analytics (EuroVA). The Eurographics Association (2020). https://doi.org/10.2312/eurova.20201086
23. Nguyen, N., Hass, J., Chen, Y., Li, J., Sill, A., Dang, T.: Radarviewer: visualizing the dynamics of multivariate data. In: Practice and Experience in Advanced Research Computing, pp. 555–556. PEARC 2020, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3311790.3404538
24. Nguyen, N.V.T., Dang, T.: Ant-SNE: tracking community evolution via animated t-SNE. In: Bebis, G., et al. (eds.) ISVC 2019. LNCS, vol. 11844, pp. 330–341. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33720-9_25
25. Oskolkov, N.: tSNE vs. UMAP: global structure. http://towardsdatascience.com/tsne-vs-umap-global-structure-4d8045acba17. Accessed 08 Jan 2021
26. Pham, V., Nguyen, N., Li, J., Hass, J., Chen, Y., Dang, T.: MTSAD: multivariate time series abnormality detection and visualization. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 3267–3276 (2019)
27. Tsay, R.S.: Multivariate Time Series Analysis: With R and Financial Applications. Wiley, Hoboken (2013)
28. TTU: high performance computing center (HPCC) at Texas tech university. website (2020). https://www.depts.ttu.edu/hpcc/ Accessed 6 July 2020
29. Van Der Maaten, L.: Accelerating t-SNE using tree-based algorithms (2014)
30. Van Der Maaten, L., Postma, E., Van den Herik, J.: Dimensionality reduction: a comparative (2009)
31. Wold, S., Esbensen, K., Geladi, P.: Principal component analysis (1987)