

Visualizing Temporal Patterns in Representation Data

Figure 1: A temporal visualization of 80 widely studied diseases, chemicals, and drugs in the scientific community of biomedicine from 2006 to 2015.

ABSTRACT

Representation learning techniques, particularly applied deep architectures, can learn effective representations of data, which is important for the performance of machine learning and is also valuable for us to understand large-scale data. However, few visualization techniques were developed for visualizing patterns, especially temporal patterns, in high-dimensional representation data. We introduce a general approach for visualizing temporal patterns in representation data based on t-SNE. We adapted t-SNE for reducing the dimensionality of temporal representation data and proposed a visualization framework to display the temporal data. We also illustrated a case study of a set representation data produced by word embedding technique to demonstrate our proposed visualization.

Index Terms: Temporal visualization; t-SNE; Information visualization

1 INTRODUCTION

Representation learning is important for the performance of machine learning methods as well as understanding large-scale data. It has become a rapidly developing field in the machine learning community [3]. It is widely used in speech recognition [8], vision recognition [20] and natural language processing [17]. The high dimensions of representation data obtained from these representation learning techniques prevent the direct display and instantaneous recognition of structures and patterns of the data. Various approaches have been proposed to visualize high-dimensional data. Some visualization techniques display more than three data dimensions, such as parallel coordinates [10] and pixel-based visualizations [13]. On the contrary, visualization techniques [16] based on dimensionality reduction convert the high-dimensional data into two or three dimensions such that data can be displayed directly.

Despite these techniques of visual representations, it is still challenging for users to identify critical information for their analysis task from a large-scale data set because of the diversity of their information needs. To resolve this issue, many user-driven visualization techniques and interaction methods were introduced, such as user-defined representative projections [14, 15], flexible zoom mechanism [2, 4] and showing different levels of detail [9]. These features are desirable for various task-oriented data analysis and exploration. However, few techniques explicitly visualize the temporal patterns of representation data.

Temporal data is common in the real world such as scientific publications, video recordings, server log and any datasets where entities have timestamps. Temporal patterns in such datasets are valuable to characterize critical dynamics at both individual-level and aggregate-level. For examples, a dramatic semantic change of a term describing a research topic in scientific publication data may indicate radical novelty and rapid increase of new scientific knowledge in scientific activities related to this research topic; a dramatically changing pattern shared by multiple research topics may indicate an underlying change of the intellectual structure of a certain research community. A temporal visualization at aggregate-level can provide an overview of an overall pattern for identifying an individual or a group of interest for further exploration and also gives rich contextual information for investigating individuals and their interactions. In this paper, we present a temporal visualization method for highdimensional representation data based on t-Distribution Stochas-

^{*}e-mail: jiangen.he@drexel.edu

[†]e-mail: chaomei.chen@drexel.edu

tic Neighbor Embedding (t-SNE) [16], which aims at providing a generic approach for visualizing temporal changes of individuals as well as temporal patterns in a large dataset. Compared to recent work on temporal high-dimensional data visualization also based on dimension reduction techniques [11, 19], our proposed method provides a visualization in a single view for a temporal overview instead of using different time snapshots.

2 METHODS

In this section, we describe how we align representation data learned through representation learning techniques at different time points; Then, we introduced temporal t-SNE for converting highdimensional representation data into two-dimensional data across time points; Lastly, the visualization pipeline and methods are described.

2.1 Data Alignment

Supposing we obtained a matrix of representations $\chi^t = \{x_1^t, x_2^t, ..., x_n^t\}$ for each time point t = 1, 2, ..., T where $\chi_t \in \mathbb{R}^{d \times n}$ and x_i^y is the representation vector of entity *i* at time point *t*. The representation vectors for different time points produced separately through most of representation learning techniques (e.g. word2vec [17]) are in different vector spaces. The different vector spaces preclude the comparison of entities across time points. In order to compare vectors at different time points, vectors from different time points need to be aligned into a same coordinate axes. We use orthogonal Procrustes to align the learned high-dimensional representations [5]. We align across time points while preserving cosine similarities by optimizing

$$\mathbf{R}^{t} = \arg\min_{\mathbf{Q}^{\mathsf{T}}\mathbf{Q}=\mathbf{I}} \left\| \mathbf{Q}\chi^{t} - \chi^{t+1} \right\|_{F}$$

with $\mathbf{R}^{(t)} \in \mathbb{R}^{d \times d}$.

2.2 Temporal t-SNE

We employ t-SNE to convert high-dimensional representation vectors into low dimensionality. The conversion goal is to convert a series of high-dimensional representation dataset $\{\chi^1, \chi^2, ..., \chi^T\}$ into low-dimensional map point set $\{\mathcal{Y}^1, \mathcal{Y}^2, ..., \mathcal{Y}^T\}$ where $\mathcal{Y}^t = \{y_1^t, y_2^t, ..., y_n^t\}$ (n = 2 in this paper). Although converting all the entities from different time points together is the most straightforward way, it may not be an efficient and effective way. If converting through t-SNE without distinguishing time points t, the matrix of similarities between high-dimensional data points would be $T * n \times T * n$. Modeling the pairwise similarities for mapping data points into two dimensions will lead to high computing costs if n or T is large. Actually, many similarities are not necessarily needed to be modeled for visualization, such as the similarity between x_i^t and $x_j^{t-\Delta t}$ where $i \neq j, \Delta t > 1$. Additionally, modeling the unnecessary similarities may weaken representing the temporal changes of individuals.

We propose a temporal t-SNE to convert high-dimensional data points at different time points t separately in a reverse chronological order (i.e. T, T - 1, ..., 1). The conversion starts from t = T to t = 1 and the converted results at t + 1 is used as input for the one at t. For t = T, original t-SNE is applied and the map points \mathcal{Y}^T produced by t-SNE will work as input for the conversion of data points at T - 1. For t < T, we model the pairwise similarities between data points in $\{\chi^t, \chi^{t+1}\}$, i.e. we are only interested in modeling the pairwise similarities between high-dimensional data points at t and its counterparts at t + 1. Although the map points \mathcal{Y}^{t+1} are used as input for converting data points at t, \mathcal{Y}^{t+1} are immutable in the conversion at t, i.e. our method only change \mathcal{Y}^t to minimize the cost function. Pseudocode for temporal t-SNE is presented in Algorithm 1. The technical details of t-SNE we didn't describe can be found in [16].

Algorithm 1: Temporal t-SNE. **Data:** Data set $\{\chi^1, \chi^2, ..., \chi^T\}$, cost function parameters: perplexity *Perp*, optimization parameters: number of iterations R, learning rate η , momentum $\alpha(r)$. Result: Low-dimensional data representation $\{\mathcal{Y}^1(R), \mathcal{Y}^2(R), ..., \mathcal{Y}^T(R)\}$ 1 $\mathcal{Y}^{T}(R) = \text{Original-tSNE}(\chi^{T}, Perp, R, \eta, \alpha(r));$ 2 $\mathcal{V}^{previous}(R) = \mathcal{V}^T(R)$: 3 for t=T-1 to 0 do compute pairwise affinities $p_{i|j}$ between x_i and x_j with 4 Prep where $x_i, x_j \in \chi^t \cup \chi^{t+1}$; for r = 1 to R do 5 compute low-dimensional affinities q_{ij} between y_i and 6 y_j where $y_i, y_j \in \mathcal{Y}^{t+1}(R) \cup \mathcal{Y}^t(r)$; compute gradient $\frac{\delta C}{\delta \mathcal{V}^t}$; 7 $\mathcal{Y}^{t}(r) = \mathcal{Y}^{t}(r-1) + \eta \frac{\delta C}{\delta \mathcal{Y}^{t}} + \alpha(\mathcal{Y}^{t}(r-1) - \mathcal{Y}^{t}(r-2))$ 8 9 end

10 end

2.3 Visualization

In a similar way to Time Curves [1], we employ a curve to represent the temporal changes of an individual. Each node on the curve is an observed time point and the curve connected nodes according to their temporal ordering. Time Curves was designed to visualize the temporal change of an individual; its visualization is based on self-similarity, i.e., the distance between two nodes is determined by how similar between corresponding two time points of the visualized individual. However, the distance between two nodes in our visualization is determined by both self-similarity and similarities to other individuals. For example, in the Figure 3, the distance between the node 5 of A and the node 3 of A represent the temporal change of A within a certain time-period, but the distance between node 5 of A and node 9 of B represents the difference between them. Based on this basic design rationale, we proposed several methods of visual encoding and layout optimization of visual elements to obtain a visualization showing clear temporal patterns. Visualization pipeline is described through a real-world example ¹ in Figure 2.

Visual Encoding. Since the curves are used to convey temporal information, it should be intuitive to follow the ordering of time points. We apply several visual encoding strategies in the visualization. Each segment of a curve is color coded, starting in orange and ending in dark purple (see the bottom of Figure 3) such that the more orange part of a specific curve depicts the temporal change of the corresponding entity at earlier stages of the observed period. We also encode the nodes representing the time points on a curve by the same way. In our design, the duration between two time points remains same for all the curves, so the same color of the nodes on different curves indicates that information they display was observed at the same time point. A similar design is proposed in [6].

As we use the map points $\{y_i^T\}$ as benchmarks for obtaining map points at t where t < T, we set the nodes at time point T with a slightly larger size than others such that the nodes would be visual benchmarks.

If the representations of a particular entity at more than two adjacent time points are same, we use a light gray ring with different radius to indicate the number of time points where the representations are same. Rings with a larger radius means more time points. For example, the node 3 and node 5 of A in Figure 3 represent more than one time points.

¹A web version of this example can be accessed through this link: https://www.cs.drexel.edu/jh3328/vis/02/dy_tsne_top50.html.



Figure 2: Visualization pipeline. Example: a temporal visualization of representations of 50 widely studied chemicals and drugs in biomedical science community from 2006 to 2016.



Figure 3: Temporal visualization examples of two entities A and B.

Node clustering. In many scenarios, the temporal changes of entities might be small. Minor temporal changes of entities may be less important information, and they may reduce the overall clarity of an overview showing numerous time curves. To increase the saliency of important patterns, we cluster nodes by employing a one-dimensional clustering algorithm proposed in [18] which was originally introduced for clustering edges in graphs. The algorithm is more effective than clustering algorithms making use of pairwise similarities for clustering, such as k-means. The process of the algorithm is straightforward: given a specific entity i, each representation vector x_i^t is scanned according to increasing order of t; the scanning starts from x_i^0 , which forms the first cluster; if the distance between the new x_i^t and the previously scanned x_i^{t-1} doesn't exceed a predefined value *diff* and adding the distance to the sum of previously scanned distances in this cluster doesn't exceed a predefined value *limit*, the new x_i^t is added to the cluster. Otherwise, the new x_i^t forms a new cluster. The Figure 4 describes the process where the parameters diff and limit can be tuned to decide how smooth the curves are, which is helpful for task-oriented analysis. The default value of *limit* is the median of the sum of distances of each entities across t(t = 1, 2, ..., T); the default value of *diff* is the default *limit* divided T. Based on the results of node clustering, we merge all the nodes in each cluster into a new supernode. A real-world example of node clustering is shown in Figure 2(b) where supernodes are annotated with gray rings.

It is worth noting that we use the original high-dimensional representation data rather than the low-dimensional map points to compute the distances because the map points highly depend on what entities are included in visualization and their distribution. Observing the differences between Figure 2(a) where no node clustering is applied and Figure 2(b) where node clustering is applied, we can see that some of the curves with crowding nodes were not merged, but some of the curves with spaced node placement were merged. A pattern of interest may be identified after node clustering. The curves still with crowding nodes after node clustering may indicate that the temporal changes of corresponding entities are caused or can be explained by entities beyond the entities get involved in the visualization. This feature may provide valuable hints for further



Figure 4: Visualization examples of two entities A and B.

exploration.

Overlap removal. Since t-SNE place similar nodes close to each other, we still can see many overlaps and overly crowd node placement in the Figure 2(b) after node clustering. We apply collision detection based on force-directed algorithm to remove the overlaps. The visualization with overlap removal can be seen in Figure 2(c) where the legibility has been improved.

Labeling The labels of entities are useful for characterizing the entities, but directly placing labels will cause overlaps due to varying widths of labels and possible crowding placement of nodes. We applied simulated annealing to produce a label placement which places labels around the node representing most recent time point (t = T) with minimum overlaps between labels. An example of label placement produced by simulated annealing is shown in Figure 2(d).

3 CASE STUDY

In this section, we describe a case study where representations of widely studied disease, chemicals, and drugs in the biomedical science community were visualized by our proposed visualization method. The representations are produced by a commonly used a representation learning technique named word2vec [17] with a large-scale scientific publication data set MEDLINE/PubMed. MED-LINE/PubMed data contains over 26 million journal citations and abstracts for biomedical literature from around the world which is often cited as the largest database of biomedical publications. We use the baseline set of MEDLINE/PubMed abstract records released in December 2016 for training word embedding models. Each publication in the MEDLINE/PubMed is indexed by a set of descriptors from the Medical Subject Headings (MeSH). We selected 80 descriptors within categories of 'Diseases[C]' and 'Chemical and Drugs[D]' ² that were most frequently used to index publications for this case study. The visualization of these 80 descriptor representation data from trained word embedding model is shown in Figure 1³.

At first glance, we can see the visualized MeSH descriptors were roughly grouped into two clusters. The cluster in the top half of

³A web version of this example can be accessed through this link: https://www.cs.drexel.edu/jh3328/vis/01/dy_tsne_top80.html.

²MeSH Tree View, https://meshb.nlm.nih.gov/treeView

the figure mainly includes descriptors of diseases and the one in the bottom half mainly includes descriptors of chemicals and drugs. Comparing between these two clusters, the pattern that temporal changes of disease descriptors are more dramatic than chemical and drug descriptors. Many curves representing chemicals in the bottom cluster were merged into one node due to few temporal differences, such as 'Nitrogen', 'Zinc' and so on. These patterns may indicate that scientific research on diseases has more radical novelty, uncertainty, or controversies than research on chemicals.

Some interesting pairwise relations can be seen in the visualization. For example, 'DNA, Complementary' in the very left of the visualization was constantly moving toward 'Genetic Makers' in recent ten years; 'Colonic Neoplasms' and 'Colorectal Neoplasms' in the upper-right-hand corner were always twisted together; 'Cyclic AMP' and 'Membrane Glycoprotein' in the lower-right-hand corner were gradually moving toward each other along 'zigzag' paths.

The evolving patterns of individuals are also easy to be identified. For example, we can see research on intellectual disability changed dramatically in the first three years (from 2006 to 2009) but has become relatively 'stable' in recent years; 'Stroke' has a dramatic change in 2012 but the change only lasted for one year.

In this case study, we can quickly identify many interesting patterns. However, identifying more critical information for reasoning and analysis needs to be supported by visual exploration features of analytics systems as well as essential domain knowledge.

4 CONCLUSIONS AND FUTURE WORK

In this work-in-progress paper, we present a visualization technique that can reveal temporal patterns of an individual as well as a set of representation data. We introduced data processing methods to aligning representation data in different vector spaces and reducing the dimensionality of temporal high-dimensional representation data; We also proposed a series of visualization techniques for obtaining a visualization with clear patterns; At last, we illustrated a case study on representation data from embedding models trained by word2vec.

Future work mainly includes improving the scalability of the visualization, adding visual exploration features, exploring a broad range of representation data examples, quantitative evaluation, and user studies.

Scalability. Although various visualization techniques applied for improving the legibility, clutters are hard to be removed as the number of entities increase. There two possible approaches to resolve this issue: glyph design [7] and curve bundling [21]. Using glyphs instead of curves may provide a clear overview; curve bundling may reduce clutters greatly.

Visual exploration. Many patterns can easily to be identified in our proposed visualization, but more critical and contextual information is needed for further reasoning and analysis. Interaction techniques, such as brushing techniques [12] and projection controlling methods [14, 15] may be effective for visual exploration.

Various examples. In this paper, we only explored an example of word embedding a representation learning technique from the field of natural language processing. In the future, we will explore examples of various representation learning techniques from different fields.

ACKNOWLEDGMENTS

The work is in part supported by the National Science Foundation (Award Number: 1633286).

REFERENCES

 B. Bach, C. Shi, N. Heulot, T. Madhyastha, T. Grabowski, and P. Dragicevic. Time curves: Folding time to visualize patterns of temporal evolution in data. *IEEE Transactions on Visualization and Computer Graphics*, PP(99):1–1, 2015. doi: 10.1109/TVCG.2015.2467851

- [2] M. Behrisch, J. Davey, F. Fischer, O. Thonnard, T. Schreck, D. Keim, and J. Kohlhammer. Visual analysis of sets of heterogeneous matrices using projection-based distance functions and semantic zoom. In *Computer Graphics Forum*, vol. 33, pp. 411–420. Wiley Online Library, 2014.
- [3] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis* and Machine Intelligence, 35(8):1798–1828, Aug 2013. doi: 10.1109/ TPAMI.2013.50
- [4] J. Choo, C. Lee, C. K. Reddy, and H. Park. Utopian: User-driven topic modeling based on interactive nonnegative matrix factorization. *IEEE transactions on visualization and computer graphics*, 19(12):1992– 2001, 2013.
- [5] W. L. Hamilton, J. Leskovec, and D. Jurafsky. Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proc. Assoc. Comput. Ling. (ACL)*, 2016.
- [6] J. He and C. Chen. Spatiotemporal analytics of topic trajectory. In Proceedings of the 9th International Symposium on Visual Information Communication and Interaction, pp. 112–116. ACM, 2016.
- [7] C. Healey, S. Kocherlakota, V. Rao, R. Mehta, and R. S. Amant. Visual perception and mixed-initiative interaction for assisted visualization design. *IEEE transactions on visualization and computer graphics*, 14(2):396–411, 2008.
- [8] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, Nov 2012. doi: 10.1109/MSP.2012.2205597
- [9] T. Höllt, N. Pezzotti, V. van Unen, F. Koning, E. Eisemann, B. Lelieveldt, and A. Vilanova. Cytosplore: Interactive immune cell phenotyping for large single-cell datasets. In *Computer Graphics Forum*, vol. 35, pp. 171–180. Wiley Online Library, 2016.
- [10] A. Inselberg and B. Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics 1987*, pp. 25–44. Springer, 1987.
- [11] D. Jäckle, F. Fischer, T. Schreck, and D. A. Keim. Temporal mds plots for analysis of multivariate data. *IEEE transactions on visualization and computer graphics*, 22(1):141–150, 2016.
- [12] H. Jänicke, M. Böttinger, and G. Scheuermann. Brushing of attribute clouds for the visualization of multivariate data. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 2008.
- [13] D. A. Keim, M. C. Hao, J. Ladisch, M. Hsu, and U. Dayal. Pixel bar charts: A new technique for visualizing large multi-attribute data sets without aggregation. In *IEEE Symposium on Information Visualization*, 2001. INFOVIS 2001., pp. 113–120, 2001.
- [14] B. C. Kwon, H. Kim, E. Wall, J. Choo, H. Park, and A. Endert. Axisketcher: Interactive nonlinear axis mapping of visualizations through user drawings. *IEEE transactions on visualization and computer graphics*, 23(1):221–230, 2017.
- [15] S. Liu, P.-T. Bremer, J. Jayaraman, B. Wang, B. Summa, and V. Pascucci. The grassmannian atlas: A general framework for exploring linear projections of high-dimensional data. In *Computer Graphics Forum*, vol. 35, pp. 1–10. Wiley Online Library, 2016.
- [16] L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. Journal of Machine Learning Research, 9(Nov):2579–2605, 2008.
- [17] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In Advances in neural information processing systems, pp. 3111– 3119, 2013.
- [18] D. Peng, N. Lu, W. Chen, and Q. Peng. Sideknot: Revealing relation patterns for graph visualization. In *Pacific Visualization Symposium* (*PacificVis*), 2012 IEEE, pp. 65–72. IEEE, 2012.
- [19] P. E. Rauber, A. X. Falcão, and A. C. Telea. Visualizing time-dependent data using dynamic t-sne. *Proc. EuroVis Short Papers*, 2(5), 2016.
- [20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
- [21] H. Yu, C. Wang, C.-K. Shene, and J. H. Chen. Hierarchical streamline bundles. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1353–1367, 2012.