FacIt: Factorizing Tensors into Interpretable and Scrutinizable Patterns

Xidao Wen*
School of Computing and Information
University of Pittsburgh

Yu-Ru Lin†
School of Computing and Information
University of Pittsburgh

Yongsu Ahn‡
School of Computing and Information
University of Pittsburgh

Konstantinos Pelechrinis§
School of Computing and Information
University of Pittsburgh

Xi Liu¶
Program of Computer Engineering
Texas A&M University

Nan Cao‖
College of Design and Innovation
TongJi University

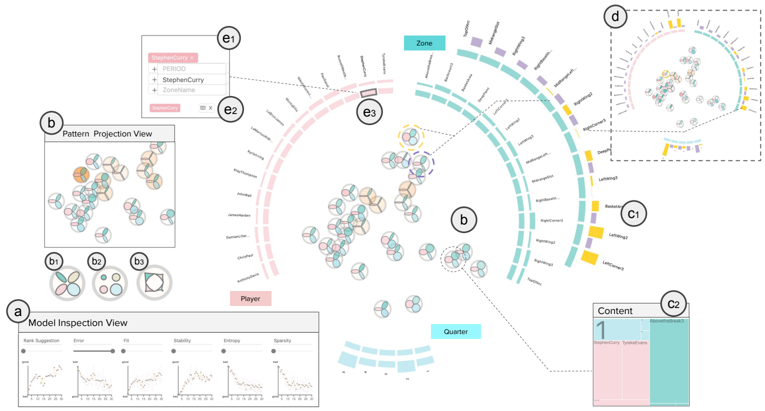

Figure 1: Using *FacIt* to interpret and scrutinize patterns based on tensor factorization from NBA shot data: (a) Model Inspection View provides various metrics of model sensitivity for selecting a desirable setting of rank from different aspects. (b) Pattern Projection View provides users high-lever overview of the entire pattern space. (c) Circular Bar Charts (c1) and Treemap view (c2) allow for examining the detailed content of patterns. (d) Pattern Comparison Mode allows users to analyze pairs of common and discriminative patterns and their associated items. (e) Pattern Query Mode enables users to retrieve most relevant patterns (e2) by query (text) input (e1) and item bars (e3).

Abstract

Tensor Factorization has been widely used in many fields to discover latent patterns from multidimensional data. Interpreting or scrutinizing the tensor factorization results are, however, by no means easy. We introduce *FacIt*, a generic visual analytic system that directly factorizes tensor-formatted data into a visual representation of patterns to facilitate result interpretation, scrutinization, information query, as well as model selection. Our design consists of (i) a suite of model scrutinizing and inspection tools that allows efficient tensor model selection (commonly known as rank selection problem) and (ii) an interactive visualization design that empowers users with both characteristics- and content-driven pattern discovery. We demonstrate the effectiveness of our system through usage scenarios with policy adoption analysis.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Visual analytics

1 Introduction

As a dimension reduction technique for the high-dimensional dataset, tensor factorization has been widely used to identify the latent patterns from multi-aspect data. Similar to other methods, such as singular value decomposition (SVD) [13] or latent Dirichlet allocation (LDA) [5], tensor factorization helps us to extract latent patterns with the noise of raw data removed. Then, such patterns tend to be more abstract, compressed, and in general, better describe the correlations and interactions among the original set of dimensions.

Tensor factorization has been used to discover multi-aspect patterns to jointly describe the underlying data phenomenons, in many real-world applications, such as social network analysis [20], web search [2],] brain data analysis [25], and health care [15,23]. Despite its wide range of applications, identifying insightful patterns from Tensor Factorization still poses the two challenges. (1) The mismatch between human information need/interest and optimization goals: Tensor Factorization is optimized towards minimizing the discrepancy between data and model. However, human information need can be more than this. For example, users may want to sacrifice its fit with the increased sparseness in the representation, for the sake of better interpretability. (2) The mismatch between factorization results and human understandability: The factorization results are not readily translated in a way that human see things clustered or close to one another. While these challenges call for user-driven pattern discovery from multi-aspect data, there have been a few studies in understanding users' information needs in the process of Tensor Factorization. Viola [7] and TPFlow [22] are among the early attempts to understand users' need. However, their primary focus is limited on applications within a spatio-temporal context.

To address the above challenges, we present *FacIt* (pronounced as *facet*), a generic visual analytic system that factorizes tensor-formatted data into a set of patterns that are visually represented, to facilitate model selection, result interpretation, and pattern scrutinization. Specifically, our work has the following key contributions:

*e-mail:xidao.wen@pitt.edu
†e-mail:yurulin@pitt.edu
‡e-mail:yongsu.ahn@pitt.edu
§e-mail:kpele@pitt.edu
¶e-mail:xiliu.tamu@gmail.com
‖e-mail:nan.cao@gmail.com

- **Task Analysis**. We conduct interviews with the users of Tensor Factorization from three different domains to understand the information need based on their experience of tensor-based analysis. We formalize a set of analytical tasks and requirements applicable to generic tensor-based analysis;
- **System Design**. We propose a system design that closely follows the design requirements distilled from the task analysis;
- **Case Study**. We demonstrate the effectiveness of the *FacIt* with detailed usage scenarios with policy adoption analysis in the United States.

2 Related Work

Tensor-based methods have become popular to discover patterns from the multi-way dataset in a variety of domains, ranging from social network analysis [20], web search [2],] brain data analysis [25], health care [23], event analytics [34, 35], sports analytics [26], etc. For example, Papalexakis and Pelechrinis construct 3-D tensor of the players, zones, and time of the game, from NBA shot dataset [26]. One of the challenges for tensor factorization technique is the rank estimation problem. Although studies have used some quality metrics (e.g., core consistency diagnostic [6] to determine the rank [19,26,28]), we argue that a single metric should not form the sole basis upon the choice of rank. Tensor Factorization serves not only to fit the tensor but also to provide a set of meaningfully interpreted patterns.

Existing high-dimensional visualization work mostly focuses on various ways of visualizing the raw data. Multivariate visualization simultaneously encodes multiple attributes in a single view with a more compact manner, such as parallel coordinates [16], space-time cube [3], ring maps [39]. On the other hand, some other visualizations provide a set of separate views, which coordinate with each other to describe the multidimensional information, such as Smartadp [21], SRVis [36], StreamExplorer [38]. However, there has not been much work in helping the analysis of the patterns from the tensor factorization. Although several recent works, e.g., Viola [7], TPFlow [22], have presented the analysis on exploring and interpreting Tensor Factorization results, their applications are limited to the spatio-temporal context.

3 Requirement Analysis

In this section, we present the preliminaries on tensor, and the procedure and data for our requirement analysis, followed by the design goals and analytical tasks to define the requirements of our system.

3.1 Tensor Preliminaries

Tensor. A tensor is a multidimensional array. Let x denote a scalar, $\mathbf{x}$ a vector, $\mathbf{X}$ a matrix, and then $\mathscr{X}$ is the extension of these concepts to higher dimensions. A tensor is called M-way tensor if it has M-dimensions or *modes*. The dimensionality of each mode is determined by the number of *items* in the corresponding mode.

CP Decomposition. The CANDECOMP/PARAFAC (CP) [8,14] is one of the most popular tensor decomposition approaches. A *CP decomposition* of an M-way tensor $\mathscr{X} \in \mathbb{R}^{I_1 \times \cdots \times I_M}$ finds a set of *factor matrices*, $\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(M)}$, that approximates the tensor as the sum of R vector outer products. It can be concisely expressed as: $\mathscr{X} \approx [\![\mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(M)}]\!] \equiv \sum_{r=1}^{R} \lambda_r \mathbf{u}_r^{(1)} \circ \ldots \circ \mathbf{u}_r^{(M)}$, where the m-th factor matrix $\mathbf{U}^m = \left[\mathbf{u}_1^{(m)} \cdots \mathbf{u}_R^{(m)}\right]$ is the combination of the vectors, $\lambda_r \in \mathbb{R}^R$ is often used to absorb the respective weights during normalization of the factor matrices' columns, "$\circ$" represents outer products, and R denotes the specified *rank* – the number of components.

Descriptors and Patterns. We refer to each entry i for $i = 1, \ldots, I_m$, as an *item* of the m-th dimension in the tensor. We denote the vector $\mathbf{u}_r^{(m)} \in \mathbb{R}^{I_m}$ as a *descriptor* consisting of entries $\langle \mathbf{u}_{ir}^{(m)} \rangle$ for $i = 1, \ldots, I_m$ from the m-th dimension that describes the contribution of the i-th *item* i to the r-th component.

Table 1: Dataset and Tensor Modes Description.

Domain	Dataset	Tensor Setting	Tensor Entry	Size
Sports	NBA shot data (2014-15)	Period, Player, Zone	#shots	$[5, 15, 14]$
Business	Ponpare Coupon Purchase	Genre,Sex,Age, Price, Period	#coupon	$[13, 2, 13, 10, 7]$
Politics	Policy Adoption in U.S.	Subjct, Year, State, Keyword	#adoption	$[16, 26, 50, 18]$

3.2 Procedure and Data

The design of *FacIt* follows the nested model described in [24], which is an iterative analytic process with one expert from each of the three different application domains – sports analytic, online purchase, and public policy analysis. Despite the diverse application domains, our experts all have extensive experience in computational data analytics in their respective domain. Our domain experts are further selected based on two criteria: 1) they have knowledge of tensor factorization; 2) they are comfortable using off-the-shelf package to run tensor factorization and have used it in their analytic tasks in the past. We use three datasets from those applications as motivating examples and use the last one to evaluate our system. The first application is the analysis of NBA shots data in season 2014-2015 [26]. The second application is the analysis of online coupon purchase [17]. The third one is the analysis of policy adoption in the United States. Table 1 summarizes the three datasets.

Over the course of six months, we held weekly meetings with our experts. During our early meetings, the system design requirements in experts' respective domains were discussed. Then, system prototypes were proposed and demonstrated to the experts for gathering feedback, and improvements were made iteratively in this process.

3.3 Design Goals

Based on a thorough literature review and interviews with the experts, we identify the following design goals for visual analytic systems to assist their tensor-based analysis:

G1. Effectiveness: How can we assist users in evaluating the effectiveness of model configuration? The result of tensor factorization is highly dependent on the configuration of the *rank*. However, there is no trivial algorithm to determine the optimal rank. Our experts suggest that the system should support the evidence on which rank leads to the effective decomposition, by providing a set of quality measurements and letting users inspect the patterns with a specific rank.

G2. Efficiency: How can we assist users in exploring the patterns more efficiently? Our experts mentioned that the process of exploring the pattern is an iterative and time-consuming process to examine all the patterns until discovering the meaningful ones. In the regard, the system should present the high-level pattern summary and allow users to locate patterns of their interests instantly.

G3. Interpretability: How can we better understand a pattern? As mentioned by our experts, it is usually a time-consuming task to interpret the pattern through examining multiple charts of descriptors all together. To complement the existing approaches, which typically present the descriptors side-by-side [7, 22, 30], the system should provide a space-efficient and well-balanced visual representation to explain multiple aspects of a pattern rather than displaying descriptors in a real-valued manner.

G4. Comparison: How can we compare patterns? While previous studies have addressed the issue of comparing multiple patterns (e.g., [1, 9–11, 18, 27, 29–32, 37]). experts were interested in selecting a pattern of their interests, and identifying other similar patterns in terms of a certain dimension or a combination of several aspects. Therefore, the system should intuitively represent patterns based on how they are similar to each other.

3.4 Analytical Tasks

To meet the design goals mentioned above, we summarize the analytical tasks as follows:

T.1: Present a comprehensive model statistics. The system should provide a comprehensive set of essential metrics (T.1.1) associated with the quality of tensor factorization. Besides, the system should allow users to configure the rank settings based on their preferences and facilitate the understanding of the trade-offs (T.1.2). Then the system should immediately present the patterns as the output of selected ranks to enable the quick fidelity check (T.1.3). (**G1**)

T.2: Present the overview of the patterns. The system should provide users with an overview of the patterns (T.2.1). Users should be able to explore patterns having a varying degree of contribution to the model (T.2.2). (**G2**).

T.3: Multi-facet pattern query. The system should provide an interaction mechanism allowing the users to query for the patterns that match with their specific items of interest (T.3.1). Upon issuing of such queries, the system should present a ranked list of patterns that are most relevant to the query (T.3.2). As a query may consist of a combination of multiple items, the system should support users to keep track of querying history (T.3.3). (**G2**)

T.4: Encode the multi-aspect characteristics of patterns in multiple scales. To deal with the multi-aspect patterns, the system should allow users to effectively grasp the overall distribution and high-level statistic of the patterns (T.4.1). The low-level displays present on-demand details of the patterns from both quantitative distributions (T.4.2) and qualitative narratives (T.4.3). (**G3**)

T.5: Encode the multi-scale comparison between patterns. The system should first summarize the similarity between each pattern's same kind of descriptors across the patterns (T.5.1). Moreover, the system should highlight the similar and discriminative items between patterns on demand so that users can immediately spot how two patterns differ from and concur to each other in each descriptor (T.5.2). (**G4**)

4 SYSTEM DESIGN

Based on the design requirements and analytical needs, we designed and developed *FacIt*, an interactive visual analytic to facilitate the interpretation and exploration of the results from tensor factorization.

4.1 Model Inspection View: Setting the Proper Rank

The model inspection view visualizes the summary statistics of a tensor factorization model varying by the selection of the rank with a set of line charts (Fig. 1a). We present five model quality measurements with respect to: the degree of fit (*normalized reconstruction error* and *model fit*), the model sensitivity to the initialization (*model stability* [37]), and the interpretability of the model (*normalized entropy* and *sparsity*). To increase the robustness of the measurements, for each rank, we performed five runs of Tensor Factorization with the mean and standard deviation reported in the line charts. In the system, users can set the weights on different metrics to enable the rank suggestion based on their interest.

4.2 Pattern Projection View: Pattern-Level Exploration

The pattern projection view provides an overview of the patterns. Each pattern is presented in a novel form of flower glyph to encapsulate key information related to the patterns.

Projection View. This view provides an overview of the relationships between the patterns in the two-dimensional space (Fig. 1(b)). Since each pattern is jointly described by multiple descriptors, we use a multi-view extension of Multi-Dimensional Scaling (MDS) [33] to map the patterns to 2-D space. As a result, the pattern projection view offers the pairwise relationship of the patterns (i.e., similar patterns are located close to each other).

Pattern Glyph. We present the design of pattern glyph that effectively summarize the following information (Fig. 1(b1)):

- Pattern Dominance. Analogous to PCA, where each component is associated with an amount of variance explained, we can also rescale the columns of each factor matrices to be unit length, and absorb the scalings into λ_r for each pattern r.
- Descriptor Informativeness. Given a descriptor $\mathbf{u_r}^{(m)}$ with m_i set of items, we first compute entropy $entr_r^m$ of $\mathbf{u}_r^{(m)}$ and use it as a proxy of its informativeness.
- Descriptor Similarity. We use the $\bar{\mathbf{u}}^{(m)}$ to denote the distribution of the m-th descriptor averaged over R components. Then, the similarity between the m-th descriptor of the r-th component to $\bar{\mathbf{u}}^{(m)}$ is calculated based on a Spearman rank.

To effectively visualize and summarize the multi-dimensional nature of the pattern, we adopt the shape of a flower (=pattern) with its petals (=descriptors). We encode each petal by using an *ellipse* and rotate the ellipses so that all petals take up the entire circle (360°) as shown in Fig. 1(b1). We also encode the pattern dominance λ_r with the saturation of the outer circle surrounding the glyph, indicating larger variance with more vivid color. The height of the ellipse represents the entropy $entr_m$ of the m-th descriptor, where a petal with narrow ellipse represents a large entropy of that descriptor (i.e., balanced distribution over the entire set of items). We encode how each descriptor is similar to the average distribution using the color saturation of the petal. In this way, a petal with darker saturation indicates its similarity to the average of all the patterns.

Design Alternatives. Over the course of the interviews with our experts, we have proposed alternative designs, such as Fig. 1(b2) and Fig. 1(b3). In Fig. 1(b2), the curvature of the petal represents the informativeness, where a curved petal indicates a more focused distribution while a round petal indicates more balanced distributions. The similarity in this design is double-coded by the size and color saturation of petal. However, it turned out that the curvature is not effectively differentiating the entropy values, according to the experts. In Fig. 1(b3), the circles represent the descriptors where the radius indicates the level of informativeness and the color saturation for the similarity. While the effectiveness of the design was appealing to our expert, the circle of descriptors with small informativeness, is likely to become extremely small for experts to read.

4.3 Pattern Detail Examination: Interpreting the pattern

Pattern Detail Examination is designed to contain two coordinated views, one for the quantitative distribution and one for the qualitative narrative of the pattern.

First, we provide a design of a circular bar chart to present the quantitative details of each pattern (Fig.1(c1)). This view consists of multiple circular bar charts, each of which indicates one descriptor. Each item in a descriptor is represented as a bar corresponding to its value. We use circular design for three considerations: 1) to enable the space-efficiency by allocating the pattern projection view inside its circular layout; 2) to match the design between the visual representation of high-level (patterns as glyph) and low-level (items as bars in the circular pattern descriptor details), where the color and orientation of each dimension's visual encoding accord to each other; 3) to expedite the exploration process, such that users can transition between the Pattern Overview and Pattern Detail View.

Second, we provide a Treemap View for users to qualitatively examine the pattern narratives (Fig. 1(c2)). Treemap provides its compact and space-filling displays of hierarchical information, which summarizes the nested nature of the patterns (patterns→ descriptors→ items). Each small rectangle of the Treemap represents an item having the value as its size, and the membership of descriptor as its color.

4.4 Pattern Query Mode

Pattern Query Mode allows the users to efficiently locate the patterns that match with the items of their explicit interest. Fig. 1(e1) and (e3) presents two alternatives venues to support the users in issuing their queries. We develop the Query Panel which contains the query input box for each of the descriptors, allowing users to input the

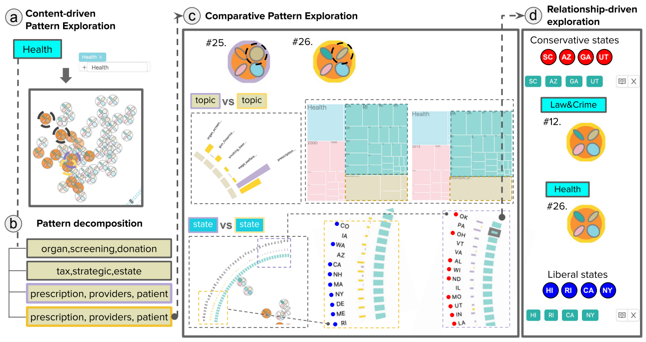

Figure 2: **Policy Adoption: Pattern Scrutinization**. ⓐ Yvonne first queries the "Health" category to detect the most relevant patterns. ⓑ She found that the topic dimension decomposes patterns health-related pattern into different political agendas, two of which fit into her interest. ⓒ When she selected and compared those two patterns, they had different distributions of topic and state. She especially noticed that states with the opposite ideology are dominant actors in the patterns. ⓓ She further explored the relationship between the most conservative and liberal states by querying each of them. The system showed her the difference in terms of most relevant categories.

item of their interest. Once a query is submitted, the most relevant patterns are retrieved and ranked based on their *relevance* to the user query. Given a query $Q = \{q_1, q_2, \dots\}$ that consists of multiple queried items, the relevance of a pattern r to the query Q is given as: $rel(r) = \prod_{m \in M_Q} \prod_{i \in I_Q^{(m)}} \mathbf{u}_{ir}^{(m)}$, where M_Q are all modes (dimensions) involved in the query, and $I_Q^{(m)}$ are the set of items involved in the query from the m-th dimension, and $\mathbf{u}_{ir}^{(m)}$ is the value for the i-th item from the m-th dimension with respect to the r-th component. We also present the query bars as the inner ring of the circular bar in alignment with item circular bar chart. Users can trigger the query by clicking on items of their interests. To keep track of queries that users have performed, we add the support of query book for users to bookmark queries that they would like to quickly retrieve later.

4.5 Pattern Comparison Mode

The system provides both the high-level comparisons as well as details-on-demand comparison in the Pattern Projection View and Detail View. Once a pattern is selected, the system updates the petals' color saturation from the rest of the patterns, according to the similarity between the selected pattern and each pattern. When users select any two patterns, the two bar charts are lined up in a way that 1) each item bar from two charts are adjacent to each other, and 2) items are re-ordered based on the difference between the two patterns by manipulating *superposition* and *explicit encoding* [12], to move discriminative items to both sides of the circular bar chart, and common item to the middle to ease the comparison.

5 Case Study: Policy Adoption Analysis

We use policy adoption analysis to showcase the effectiveness of *FacIt*, through discussing the pattern analysis of policy adoption data in US state politics since the 1990s. Yvonne is a policymaker in the department of public health in Pennsylvania state government. She is now working on proposing legislation of national health insurance, with three questions to address in the system: (1) What are the dominant adoption patterns of health-related policies? (2) Can we further decompose the dimension of those patterns into health-related specific topics? How are the dominant patterns different from each other? (3) Are the political interests of states different? How are they related to state characteristics (e.g., liberal or conservative)?

Model Selection. In Model Inspection View, she found that all rank metrics are improved as rank became larger but began to converge around $R = 50$ (T.1.1). Since she expected that it would be more insightful to have patterns to be decomposed into smaller patterns, it also fits into her interest to set the model with a larger rank (T.1.2). After clicking on the dots that represent different ranks and reviewing the corresponding outputs (T.1.3), she decided to further explore the patterns with the setting of $R = 50$.

Content-driven Pattern Exploration. She started with querying health policy. After selecting "Health" in the subject query box (T.3.1), she found that the system highlighted a set of patterns that are related to "Health" policy using the color saturation in the pattern glyph in the Pattern Projection View (T.3.2). She focused on exploring the four most relevant patterns with greater relevance scores (Fig. 2ⓐ). When she looked at the dominant topics of those patterns, she noticed that they were characterized by different sub-topics (Fig. 2ⓑ). Among them, Pattern 25 and Pattern 26 were of her interest for two reasons: (1) The dominant topic of two patterns is "prescription, patient, provider", which directly relates to her current interest (Health insurance policy), and (2) The topical distributions were different in that Pattern 25 was dominant by a few topics and pattern 26 spread out through multiple topics (T.5.1).

Comparative Pattern Exploration. She began to explore two patterns in detail (Fig. 2ⓒ). In terms of the dimension of states, she noticed that Pattern 26 was mainly driven by liberal states, as compared to Pattern 25 driven by the conservative states in the Treemap View (T.4.3 and T.5.2). She also found that PA was the second dominant in Pattern 25 among other dominant states in the conservative side. For the topical dimension, Pattern 26 was all about "prescription, provider, patient" while Pattern 25 was distributed across multiple items with two dominant topics, "prescription, provider, patient" and "organ, screening, donation". While she was exploring the patterns, those ideological difference between two health-related topics made her come up with another question: "How the liberal and conservative states concern the different topic of policy agendas?".

Relationship-driven Pattern Exploration. At this time, she was interested in querying states by their ideology and analyzing relevant patterns. First, she started by identifying the most liberal and conservative states. According to [4], the latest state ideology score in 2017 indicated that CA, RI, HI, NY were the most liberal states, and SC, AZ, GA, UT were the most conservative states. Since *FacIt* was able to issue multiple items in a query, she made two separate queries that consisted of each of four states (T.3.1) and saved them to the query book (T.3.3). Interestingly, the two queries of liberal and conservative states resulted in different policy topics from different patterns (Fig. 2ⓓ). For liberal states, health-dominant Pattern 26 was the most relevant one. On the other hand, Pattern 12, mainly driven by "Law and crime", was the most relevant pattern topic of conservative states.

Throughout all these analyses combined, she was able to learn health-related policies in detail, and figure out which states she can better communicate with.

6 Conclusion

In this paper, we present *FacIt*, a visual analytic system for Tensor Factorization. The system is built to meet the requirements, such as model selection, results scrutinization, and interpretation, in its real-world applications. We provide an interactive design that caters to experts' different exploration strategy. The effectiveness and usefulness of *FacIt* have been evaluated through usage scenarios in the application of policy adoption analysis. In our next step, we would like to invite a panel of domain experts for in-depth interviews to evaluate the design, interaction, usability, and potential improvements of *FacIt*. Besides, we also would like to involve domain experts in the factorization and have them to be able to iteratively fine-tune the patterns so that the patterns can be more aligned to their domain knowledge.

Acknowledgement

The authors would like to acknowledge the support from NSF #1637067, and #1739413, and the DARPA UGB program.

References

[1] E. Acar, E. E. Papalexakis, G. Gürdeniz, M. A. Rasmussen, A. J. Lawaetz, M. Nilsson, and R. Bro. Structure-revealing data fusion. *BMC bioinformatics*, 15(1):239, 2014.

[2] R. Agrawal, B. Golshan, and E. Papalexakis. A study of distinctiveness in web results of two search engines. In *Proceedings of the 24th International Conference on World Wide Web*, pp. 267–273. ACM, 2015.

[3] W. Aigner, S. Miksch, H. Schumann, and C. Tominski. *Visualization of time-oriented data*. Springer Science & Business Media, 2011.

[4] W. D. Berry, E. J. Ringquist, R. C. Fording, and R. L. Hanson. Measuring citizen and government ideology in the american states, 1960–93. *American Journal of Political Science*, pp. 327–348, 1998.

[5] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[6] R. Bro and H. A. Kiers. A new efficient method for determining the number of components in parafac models. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 17(5):274–286, 2003.

[7] N. Cao, C. Lin, Q. Zhu, Y.-R. Lin, X. Teng, and X. Wen. Voila: Visual anomaly detection and monitoring with streaming spatiotemporal data. *IEEE transactions on visualization and computer graphics*, 24(1):23–33, 2018.

[8] J. D. Carroll and J.-J. Chang. Analysis of individual differences in multidimensional scaling via an n-way generalization of "eckart-young" decomposition. *Psychometrika*, 35(3):283–319, 1970.

[9] L. Espín Noboa, F. Lemmerich, P. Singer, and M. Strohmaier. Discovering and characterizing mobility patterns in urban spaces: a study of manhattan taxi data. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pp. 537–542. International World Wide Web Conferences Steering Committee, 2016.

[10] Z. Fan, X. Song, and R. Shibasaki. Cityspectrum: a non-negative tensor factorization approach. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 213–223. ACM, 2014.

[11] R. Frelat, M. Lindegren, T. S. Denker, J. Floeter, H. O. Fock, C. Sguotti, M. Stäbler, S. A. Otto, and C. Möllmann. Community ecology in 3d: Tensor decomposition reveals spatio-temporal dynamics of large ecological communities. *PloS one*, 12(11):e0188205, 2017.

[12] M. Gleicher, D. Albers, R. Walker, I. Jusufi, C. D. Hansen, and J. C. Roberts. Visual comparison for information visualization. *Information Visualization*, 10(4):289–309, 2011.

[13] G. H. Golub and C. Reinsch. Singular value decomposition and least squares solutions. In *Linear Algebra*, pp. 134–151. Springer, 1971.

[14] R. Harshman. Foundations of the parafac procedure: Models and conditions for an" explanatory" multimodal factor analysis. 1970.

[15] J. C. Ho, J. Ghosh, and J. Sun. Marble: high-throughput phenotyping from electronic health records via sparse nonnegative tensor factorization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 115–124. ACM, 2014.

[16] A. Inselberg and B. Dimsdale. Parallel coordinates for visualizing multi-dimensional geometry. In *Computer Graphics 1987*, pp. 25–44. Springer, 1987.

[17] P. Japan. Coupon purchase prediction, 2015. data retrieved from Kaggle.com, `https://www.kaggle.com/c/coupon-purchase-prediction/data`.

[18] D. Koutra, E. E. Papalexakis, and C. Faloutsos. Tensorsplat: Spotting latent anomalies in time. In *Informatics (PCI), 2012 16th Panhellenic Conference on*, pp. 144–149. IEEE, 2012.

[19] S. Lee, H. Liu, M. Kim, J. T. Kim, and C. Yoo. Online monitoring and interpretation of periodic diurnal and seasonal variations of indoor air pollutants in a subway station using parallel factor analysis (parafac). *Energy and Buildings*, 68:87–98, 2014.

[20] Y.-R. Lin, J. Sun, P. Castro, R. Konuru, H. Sundaram, and A. Kelliher. Metafac: community discovery via relational hypergraph factorization. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 527–536. ACM, 2009.

[21] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, and Y. Wu. Smartadp: Visual analytics of large-scale taxi trajectories for selecting billboard locations. *IEEE transactions on visualization and computer graphics*, 23(1):1–10, 2016.

[22] D. Liu, P. Xu, and L. Ren. Tpflow: Progressive partition and multidimensional pattern extraction for large-scale spatio-temporal data analysis. *IEEE transactions on visualization and computer graphics*, 2018.

[23] S. Mohammadi, D. F. Gleich, T. G. Kolda, and A. Grama. Triangular alignment tame: A tensor-based approach for higher-order network alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 14(6):1446–1458, 2017.

[24] T. Munzner. A nested model for visualization design and validation. *IEEE transactions on visualization and computer graphics*, 15(6):921–928, 2009.

[25] E. Pantraki and C. Kotropoulos. Automatic image tagging and recommendation via parafac2. In *Machine Learning for Signal Processing (MLSP), 2015 IEEE 25th International Workshop on*, pp. 1–6. IEEE, 2015.

[26] E. Papalexakis and K. Pelechrinis. thoops: A multi-aspect analytical framework for spatio-temporal basketball data. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pp. 2223–2232. ACM, 2018.

[27] E. Papalexakis, K. Pelechrinis, and C. Faloutsos. Spotting misbehaviors in location-based social networks using tensors. In *Proceedings of the 23rd International Conference on World Wide Web*, pp. 551–552. ACM, 2014.

[28] M. A. Prada, J. Toivola, J. Kullaa, and J. HollméN. Three-way analysis of structural health monitoring data. *Neurocomputing*, 80:119–128, 2012.

[29] A. Schein, J. Paisley, D. M. Blei, and H. Wallach. Bayesian poisson tensor factorization for inferring multilateral relations from sparse dyadic event counts. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1045–1054. ACM, 2015.

[30] X. Shi, F. Lv, D. Seng, B. Xing, and J. Chen. Exploring the evolutionary patterns of urban activity areas based on origin-destination data. *IEEE Access*, 7:20416–20431, 2019.

[31] C. H. Stock, A. H. Williams, M. S. Advani, A. M. Saxe, and S. Ganguli. Learning dynamics of deep networks admit low-rank tensor descriptions. 2018.

[32] K. Takeuchi, Y. Kawahara, and T. Iwata. Structurally regularized non-negative tensor factorization for spatio-temporal pattern discoveries. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 582–598. Springer, 2017.

[33] N. T. Trendafilov. Stepwise estimation of common principal components. *Computational Statistics & Data Analysis*, 54(12):3446–3457, 2010.

[34] X. Wen, Y.-R. Lin, and K. Pelechrinis. Pairfac: Event analytics through discriminant tensor factorization. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pp. 519–528. ACM, 2016.

[35] X. Wen, Y.-R. Lin, and K. Pelechrinis. Event analytics via discriminant tensor factorization. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 12(6):72, 2018.

[36] D. Weng, R. Chen, Z. Deng, F. Wu, J. Chen, and Y. Wu. Srvis: Towards better spatial integration in ranking visualization. *IEEE transactions on visualization and computer graphics*, 25(1):459–469, 2018.

[37] A. H. Williams, T. H. Kim, F. Wang, S. Vyas, S. I. Ryu, K. V. Shenoy, M. Schnitzer, T. G. Kolda, and S. Ganguli. Unsupervised discovery of demixed, low-dimensional neural dynamics across multiple timescales through tensor component analysis. *Neuron*, 2018.

[38] Y. Wu, Z. Chen, G. Sun, X. Xie, N. Cao, S. Liu, and W. Cui. Streamexplorer: A multi-stage system for visually exploring events in social streams. *IEEE transactions on visualization and computer graphics*, 24(10):2758–2772, 2018.

[39] J. Zhao, P. Forer, and A. S. Harvey. Activities, ringmaps and geovisualization of large human movement fields. *Information visualization*, 7(3-4):198–209, 2008.