A Quantitative and Qualitative Analysis of Tensor Decompositions on Spatiotemporal Data

Tom Henretty, Muthu Baskaran, James Ezick, David Bruns-Smith
Reservoir Labs, Inc.
New York, NY 10012
{henretty, baskaran, ezick, bruns-smith}@reservoir.com

Tyler A. Simon
Laboratory for Physical Sciences
University of Maryland
College Park, MD 20740
tasimon@lps.umd.edu

***Abstract*—With the recent explosion of systems capable of generating and storing large quantities of GPS data, there is an opportunity to develop novel techniques for analyzing and gaining meaningful insights into this spatiotemporal data. In this paper we examine the application of *tensor decompositions*, a high-dimensional data analysis technique, to georeferenced data sets. Guidance is provided on fitting spatiotemporal data into the tensor model and analyzing the results. We find that tensor decompositions provide insight and that future research into spatiotemporal tensor decompositions for pattern detection, clustering, and anomaly detection is warranted.**

I. INTRODUCTION

A constant stream of georeferenced data is being generated from now-ubiquitous GPS-equipped devices including mobile phones, automobiles, and wearable technologies. Tensor decompositions provide a proven, powerful, mathematically sound method of detecting patterns and anomalies in this data. Further, tensor decompositions have been executed on extremely large datasets using shared memory and distributed memory HPC systems. Still, investigations into the applicability of tensor decompositions to spatiotemporal data have thus far been limited. In this work we apply two well-known tensor decompositions to publicly available georeferenced data sets in order to quantitatively and qualitatively assess the results. This works makes the following contributions:

- We define a workflow for the analysis of spatiotemporal data using tensor decompositions;
- We provide guidance, through examples, on key decisions affecting the quality of decompositions;
- We establish the suitability of tensor decompositions for spatiotemporal data analysis;
- We characterize the behavior of common tensor decompositions on spatiotemporal datasets.

II. BACKGROUND

A. Tensor Decompositions

Advanced linear algebra, including methods for decomposing and analyzing data are well established in the realms of two-dimensional data and matrices [9] and graph analysis problems [8]. Real-world data, however, is often multidimensional with multiple aspects and features. In such cases, hypergraph analysis and multi-linear algebra (a generalization of linear algebra to higher dimensions) are often more applicable to real-world applications. For extracting and explaining the properties of multi-attribute data, tensors (i.e., multidimensional arrays) are a natural representation of data with multiple aspects and dimensionality [1]. Decomposing and analyzing tensors has applications in a range of domains such as cyber security, signal processing, data mining, computer vision, numerical linear algebra, numerical analysis, and graph analysis [10].

One well-known tensor decomposition model is CANDECOMP-PARAFAC (CP). A CP decomposition is similar to a higher-order Singular Value Decomposition (SVD), where a tensor is broken into a weighted sum of *components*, each representing a multidimensional pattern produced by the decomposition algorithm.

Fig. 1 illustrates a CP decomposition of a three-dimensional tensor, alternatively referred to as a tensor with *order* equal to three. The tensor has three *modes*, *a, b*, and *c*. Each mode is of size M_i and is indexed by integers from *1* to M_i. Each index represents a data point in a multidimensional space and a tuple of indices maps to some value *v*.

The decomposition in Fig. 1 produces *R components*, where *R* is referred to as the *rank*. Each component contains a *weight* λ and a vector for each mode of the tensor. The vectors contain a *score* for each index of the mode. Outer products are computed and summed to provide an approximation of the original tensor.

An example component from a decomposition of the New York City Taxi dataset (described in Section II.B) is shown in Fig. 2(a). The three modes represented are *pickup time*, *pickup location,* and *dropoff location.*

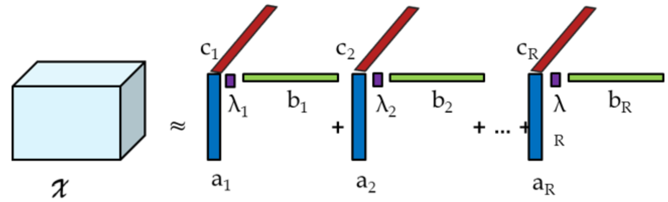
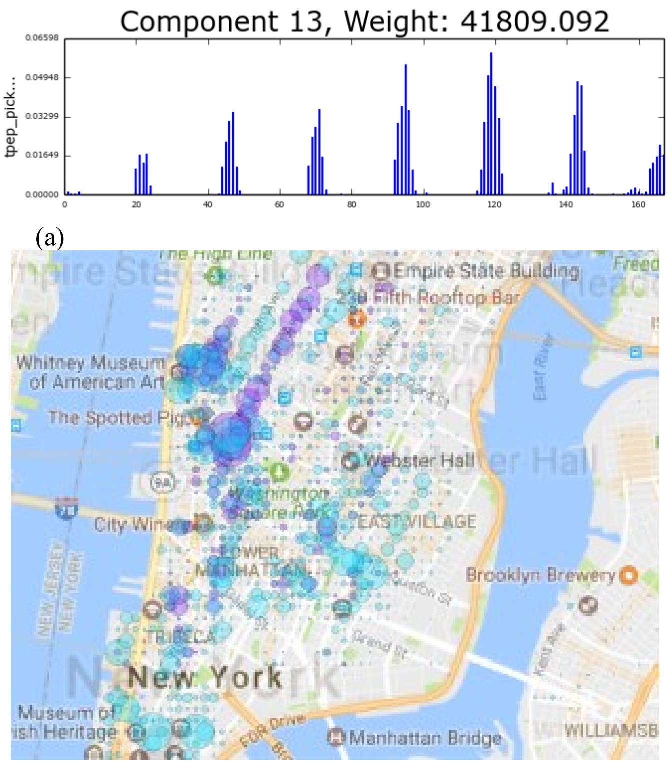


Fig. 1. An example CP decomposition of a three-dimensional tensor.

978-1-5386-3472-1/17/$31.00 ©2017 IEEE



(a)

(b)

Fig. 2. Tensor decomposition component visualization. (a) Score vector as bar chart. (b) Pickup and dropoff score vectors plotted. Map points represent score by size and mode by color (pickups are purple, dropoffs are turquoise).

The time vector is represented by a bar chart, with the mode index on the x-axis and the score for that index on the y-axis. Time runs from Sunday 12AM to Saturday 11PM. Since geospatial data is not easily interpretable in chart form, we plot scores from the location modes in Fig. 2(b). Here, the magnitude of the score is represented by the size of the point. The mode is represented by color.

We evaluated two tensor decompositions: Non-negative CP-ALS (CP-ALS-NN) [15] and CP-APR [16]. CP-ALS-NN using an alternating least squares regression technique to solve for components. This decomposition assumes an underlying Gaussian distribution to the tensor values. CP-APR uses an alternating Poisson regression and assumes an underlying Poisson distribution for the tensor values. In general, CP-APR is best used to decompose data generated by Poisson processes.

B. Datasets

Two datasets were used in this work. Both of these datasets contain real world georeferenced and timestamped multidimensional event data.

New York City Taxi (NYCT): Under the Taxicab & Livery Passenger Enhancement Programs (TPEP/LPEP), the New York City Taxi and Limousine Commission collects and provides data for yellow taxis operating in the greater New York City area [7]. Examples of data features include records of pickup and drop-off times and latitude/longitude, distances and fares. The data is provided as a CSV file; in this paper we focused on yellow cab trip data for the week of June 13-20, 2016. This dataset contains approximately 2.5 million records.

Montgomery County Traffic Violations (MCTV): As part of the openMontgomery initiative, the government of Montgomery County, Maryland collects and provides data on traffic violations issued electronically within the county [11]. Examples of data features include descriptions and latitude/longitude of violations, issuing agencies, and limited demographic information describing the offender. The data is provided as a CSV file; in this paper we focus on violations occurring between January 1, 2013 and September 22, 2016. This dataset contains approximately 815 000 records.

III. Methodology

In this section, we describe a seven-step workflow used to construct, decompose, and analyze spatiotemporal tensors. At each step, we examine the key decisions that need to be made, and provide examples from the decompositions and datasets described in Section II.

A. Feature Selection

The first step of our workflow is feature selection. Here, we choose the features from our dataset that we wish to examine for multidimensional patterns. For our example datasets, each feature is a column in the CSV file containing the data such as *timestamp*, *latitude*, and *longitude*. When selecting features for spatiotemporal tensors, we find it best to start with a small number of features (e.g., three), and add features one at a time for reasons described below.

The outer product model of tensor reconstruction used by CP decompositions (described in Section II.A) can lead to a combinatorial explosion of tensor entries represented by a component. As an example, consider a component with four nonzero scores in each mode. In a 3-mode tensor, the outer product computation produces $4*4*4 = 4^3$ tensor entries. As modes are added, the number of tensor entries represented grows exponentially.

In general, when decomposing an N-mode tensor, each component of the decomposition has a count of nonzero scores in each mode $[a_1, a_2, \ldots, a_N]$. Computing the outer product produces $N^{(a1*a2*\ldots*aN)}$ entries in the reconstructed tensor. In many cases, especially when the number of nonzero scores in a mode is high, computing the outer product introduces a significant amount of error since a tensor entry is created at every combination of nonzero score indices. As the number of features increase, at a fixed rank, the ability of the decomposition to accurately represent the original data decreases.

KEY DECISIONS: Which features to choose?

For the NYCT dataset: *tpep_pickup_datetime*, *pickup_longitude*, *pickup_latitude*, *dropoff_longitude*, *dropoff_latitude*

978-1-5386-3472-1/17/$31.00 ©2017 IEEE

For the MCTV dataset: *Date Of Stop*, *Time Of Stop*, *SubAgency*, *Description*, *Latitude*, *Longitude*, *Race*, *Gender*

B. Data Cleaning

Frequently, datasets contain corrupted feature values or are missing data. This is a particular problem with geospatial data since GPS receivers may lose the ability to accurately record location in obstructed environments. This can also be a problem in datasets that have been through automated formatting or datasets with manually input feature values.

When cleaning spatiotemporal data for tensor decomposition one can (a) remove the missing/corrupted data point (e.g., remove the row in the CSV file), (b) use a dummy value such as "*UNKNOWN,*" or (c) repair the data if possible. Ideally, data can be repaired. If looking for patterns in missing/corrupted values (e.g., GPS system failures) then use dummy values, otherwise remove the data point.

KEY DECISIONS: How to handle missing/corrupt data?

For the NYCT dataset: Entries with missing/corrupt location or time data were removed.

For the MCTV dataset: Entries with missing/corrupt location, gender, and race data were removed in most cases. Location data with latitude recorded in the longitude feature and vice versa were repaired.

C. Feature Engineering and Binning

After features have been selected and cleaned, they may be further transformed for tensor construction and decomposition. This process is referred to in the data science literature as *feature engineering*. In the context of tensor decompositions, feature engineering may include mathematical, syntactical, or structural transformations.

Mathematical transformations may include normalization of values. For spatial data a mathematical transformation may be used to round decimal latitude/longitude features or convert the values to a different coordinate reference system.

Syntactical transformations may include elimination of unnecessary detail (e.g., "*Speeding: 53 MPH in a 45 MPH zone*" to "*Speeding*"). For temporal data, date and time formats may be changed for readability. For spatial data, geolocation or conversion of coordinates to points of interest (POI) may be performed.

Structural transformations create new features by fusion, fission, or derivation of new features. This is common in georeferenced datasets where latitude and longitude are separate features. For tensor decompositions, it is better to fuse latitude and longitude into a single feature; this fused feature is then treated as a single location during decomposition.

How the transformations divide features into bins is a critical consideration for tensor decompositions. For example, rounding decimal latitude/longitude features determines the spatial resolution of the subsequent tensor decomposition. At New York City latitudes, rounding decimal latitude/longitude to three decimal places puts all coordinates into approximately city-block-sized-bins. The granularity of time binning determines the temporal resolution of the subsequent decomposition. Time may be binned into days, hours, minutes, seconds, etc. If we are looking for patterns in long distance (100+ miles) movements over long timescales (1-2 years), coarser bins for time and space are chosen (bin time by day, space by rounding to one decimal place).

Binning also has a direct effect on the computational effort required to build, decompose, and analyze a tensor. Coarser binning leads to smaller mode sizes. The algorithms and data structures used for tensor decompositions are sensitive to mode size. Smaller modes lead to faster runtimes, and less memory/disk consumption.

KEY DECISIONS: What spatial bin resolution? What temporal bin resolution? How to bin and transform other features?

For the NYCT dataset: Latitude/longitude for pickup/dropoff locations was rounded to three decimal places and fused into pickup and dropoff locations. Time was binned by hour.

For the MCTV dataset: Latitude/longitude for violation locations was rounded to three decimal places and fused into a single location feature. Date and time were left as two separate features. Date was binned by day. Time was binned by minute.

D. Sparse Tensor Construction

Next, a sparse tensor must be constructed. The main decision to be made is what values to use as tensor entries. Tensors are analogous to multidimensional arrays, and features are mapped to nonnegative integer indices of the array. In the NYCT tensor, for example, we map timestamp, pickup location, and dropoff location to indices. For each (timestamp, pickup, dropoff) tuple we must decide the value of the tensor at that entry.

Common tensor entries are occurrence, count, or feature values. Continuing the NYCT example, an occurrence entry would simply be *1* if *any* corresponding (timestamp, pickup, dropoff) existed in the (feature engineered and binned) dataset, and *0* otherwise. A count tensor entry is the number of identical tuples in the dataset. For example, if there were three trips from (12.34, 56.78) to (87.65, 43.21) at 9AM the count entry would be three (*3)* for that tuple. Feature values are taken from the engineered dataset; an example would be the fare of a taxi trip.

Tensor entries can be given negative values, although neither of the decompositions examined in this paper support them. Negative tensor entries, in general, make interpreting a decomposition much more difficult.

The other consideration when building a sparse tensor is the mapping of engineered features to nonnegative integer indices. Sorting the values can make the decomposition results easier to visualize. In the case of time, sorting such that the earliest time bin maps to the first index allows the creation of a simple decomposition component graph where time increases from left to right. With latitude/longitude features sorting is less important but can be useful depending on the visualization.

KEY DECISIONS: Occurrence, count, or feature value for tensor entry? Which feature value? Sort modes? Which ones?

For the NYCT and MCTV datasets: Tensor entries are counts and all modes are sorted.

E. Parameter Space Selection

Most tensor decompositions contain tunable parameters. Some of these parameters include the type of decomposition, rank of the decomposition, maximum number of iterations, and convergence criteria.

Each type of decomposition, for example CP-ALS-NN or CP-APR, have built-in assumptions about the distribution of data entries in the tensor and can be more or less difficult to interpret. Papalexakis et al. [3] provide summaries of many different tensor decompositions beyond the two evaluated in this work.

The rank of the decomposition determines how many components (i.e., multidimensional patterns) are produced by the decomposition. The best rank for a decomposition is highly data dependent. A rank that is too low will force unrelated data points into the same component and lower fit to the original tensor. A rank that is too high will separate related data points into different components and fragment patterns across the results.

Determining the *true rank* of a tensor, that is the rank that allows for an exact reproduction of the original tensor, is NP hard. Typically, it is not necessary to decompose to true rank for the results to be useful. Thus, it is important to select a range of ranks when first examining a dataset in order to determine a rank that produces meaningful patterns.

Maximum iterations and convergence criteria determine when an iterative tensor decomposition algorithm will terminate. Higher values of maximum iterations and/or lower values of convergence criteria will improve fit at the cost of run time.

KEY DECISIONS: Which parameters to vary and what values to give those parameters?

For the NYCT and MCTV datasets: CP-ALS-NN and CP-APR decompositions. Ranks of 10, 25, 50, 100, and 250. Maximum iterations at 250, and convergence criteria of 0.0000001.

F. Decompositions

In this step, we simply run decompositions for all of combinations of parameters selected above. Decompositions are a compute-intensive task. Higher ranks, larger mode sizes, and larger modes all lead to longer run times. Many implementations of tensor decompositions are available. For all experiments, we used decomposition implementations provided with ENSIGN [17].

KEY DECISIONS: Which implementation of the chosen tensor decompositions to use?
For the NYCT and MCTV datasets: ENSIGN

G. Plotting and Analysis

After running all decompositions, we analyze the results. For geospatial and spatiotemporal tensor decompositions, visualization is a critical tool for understanding and gaining insights from the results. For analysis, it is helpful to plot nonzero scores of each mode of each component in a bar graph. This enables rapid scanning of components for interesting patterns, especially in temporal and non-spatial modes.

Spatial modes can be visualized using geographic information systems (GIS) [12][13]. Nonzero score indices in spatial modes are converted into a representation amenable to plotting on the system of choice. For a GIS system, latitude/longitude bins and POI can be converted to points or polygons, roads can be converted to polylines, and areas of interest such as states, regions, or nations can be converted to polygons.

For GIS systems, spatial modes can be assigned to layers and grouped by component. This enables easy toggling and overlapping for cross-component analysis.

Properties of the decomposition should be mapped to properties of the visualized points, polylines, and polygons. For points, size can be made proportional to decomposition score or weight-score product. Color can be mapped to mode or component. Similarly for polylines, thickness can be mapped to score or weight-score product and color to mode or component. For polygons, border thickness and color can be mapped similarly.

Semi-transparent polylines and polygons enable rapid discernment of patterns when plotting spatial modes with many nonzero scores and when layering modes.

A text representation of decomposition components, modes, indices (with corresponding feature values), and scores can also be useful. By sorting each mode of each decomposition by score, it is easy to discern which features values are strongly represented in a component.

ENSIGN is distributed with tools for visualizing spatiotemporal tensor decompositions and integrating with GIS systems. GIS compatibility is achieved through the generation of ESRI Shapefiles [14] from geospatial modes in decomposition data. Other visualization tools including bar charts, graphs, scatter plots, parallel coordinates charts, and text tables are produced by the ThermalVision application included with ENSIGN.

KEY DECISIONS: Which plotting systems to use? Which visualizations to create? How to map decomposition and feature attributes to elements of the visualization? How to convert data from tensor and decomposition representations to visualizations of choice?

For the NYCT and MCTV datasets: Convert decomposition spatial modes to ESRI Shapefiles using command line tools included with ENSIGN. Use the open-source QGIS tool to visualize. Scale point size by score in QGIS. Produce layer(s) for each component in QGIS. Visualize time and other modes using the ThermalVision application included with ENSIGN.

IV. Experiments

All experiments were performed on a four socket, 32-core machine running at 2.2 GHz with 128 GB of RAM. Decomposition parameters are given in Section III.E.

A. *Qualitative Analysis*

CP-ALS-NN and CP-APR decompositions produced meaningful patterns and provided insight into the underlying structure of both datasets.

NYCT: Fig. 2 in Section II.A illustrates one of many easily interpretable patterns found in our experiments. This component isolated traffic to and from popular Manhattan nightlife areas with both pickups and dropoffs concentrated near popular bars on the west side of Manhattan and dropoffs fanning out to the Financial District and residential areas in the Lower East Side. In this component we see nonzero pickup time scores peaking between 10PM and 1AM and increasing from smaller values Sunday –Wednesday to peak values Thursday – Saturday nights. A similar "nightlife" pattern appeared across multiple decomposition ranks using both decompositions.

Fig. 3 shows six components that isolated areas with dense networks of pickups and dropoffs. These areas roughly correspond to well-known neighborhoods including the Upper East Side (blue), Upper West Side (green), Midtown Manhattan (magenta), Downtown Manhattan (orange), Downtown Brooklyn (turquoise), and Long Island City/Astoria (red). Times represented by these components clearly reflected daily cyclical activity, with scores peaking in mid-day, reaching minimums in early morning hours, and reaching lower peaks on weekends. Again, similar "neighborhood" patterns appeared across multiple decomposition ranks using both decompositions.

Other components clearly isolated traffic to the region's three major airports and a major train hub. Some showed activity on weekends only, others on weekdays only, and others on all seven days of the week. Higher rank decompositions led to more granular patterns, such as patterns that broke airport traffic down to specific terminals.

CP-ALS-NN and CP-APR decompositions produced similar patterns in higher weight components but lower weight components differed. In the lower weight components CP-ALS-NN isolated point-to-point traffic at very specific times while CP-APR seemed to collect widely disparate traffic.

Fig. 3. Rank 100 CP-APR decomposition of NYCT. Six components are plotted in separate colors. Pickups and dropoffs are the same color.

MCTV: Since the tensor decomposed was a 7^{th}-order tensor, much more information was conveyed in a single component. A representative pattern is shown in Fig. 4. This component shows activity that was consistent across all dates, peaked between 10PM and 1AM, was associated with the Montgomery County Police Third District, mainly involved nine violations, occurred in a wide range of locations with two distinct clusters, and affected male black, Hispanic, and white drivers.

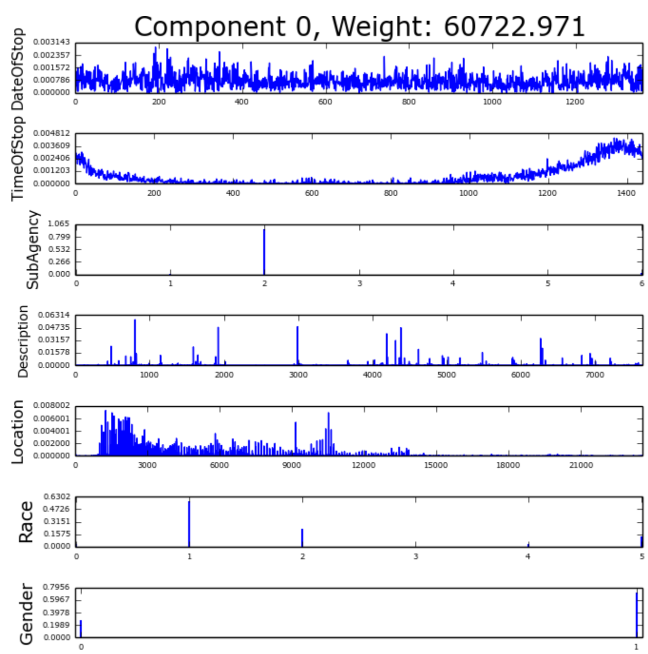
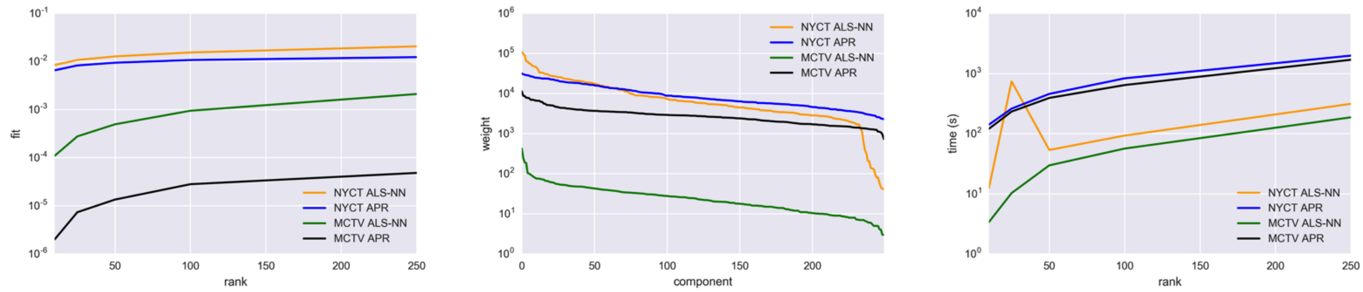


Fig. 4. Representative component of MCTV data set.



(a) (b) (c)

Fig. 5. (a) Final fit vs. rank (b) weight vs. component number (rank 250 decomposition) (c) decomposition time vs. rank

Other components isolated traffic violations during different time periods such as rush hours and business hours. Specific locations were isolated to specific offenses and time periods, such as particular red lights that were frequently run late at night.

MCTV components isolated multidimensional hotspots of space, time, and violations that were associated with one specific police department and affected different subsets of race and gender.

As in NYCT, CP-ALS-NN showed similar patterns on highly weighted components but tended towards much more specific activity than CP-APR in the lowest weighted components. Additionally, CP-ALS-NN components were much noisier than those from CP-APR.

B. *Quantitative Analysis*

Fig. 5(a) shows the final fit of a decomposition with respect to rank. As expected, fit increases with rank for all decompositions. CP-ALS-NN produces higher fit for both datasets. In absolute terms, the fit is low and ranges between 10^{-6} and 10^{-2}.

Fig. 5(b) shows the value component weights from largest to smallest on a rank-250 decomposition. There are two important features in this plot.

First, CP-ALS-NN decompositions have very high weight components, followed by a steep drop, followed by steadily declining weights, followed by another sharp drop at the lowest weight components. The rapid drop at the lowest scoring components qualitatively manifests itself as components focused on fewer, specific entries. High-weight components in CP-ALS-NN represented more tensor entries than their counterparts in CP-APR. In NYCT decompositions this did not alter the fundamental appearance of high-weight components, although it altered the ordering. CP-APR weights, on the other hand, steadily decreased with much smaller and shorter rises and dips in the weights of the largest and smallest weight components, respectively.

The pattern described above was seen in decompositions at all ranks. Further investigation is required to understand exactly why the extreme weights of CP-ALS-NN increase and decrease so much.

Second, CP-ALS-NN produced weights two orders of magnitude smaller than CP-APR on the MCTV dataset. This behavior was consistent across ranks. CP-ALS-NN produced decompositions with more nonzero scores in components. Outer products of these dense components across seven modes likely led to a combinatorial explosion of tensor entries represented by each component. Higher weights magnify this effect and would lead to a poorer fit to the original tensor. When CP-ALS-NN was rerun on the same dataset with the *DateOfStop* and *SubAgency* modes removed, the weights increased by an order of magnitude.

Finally, Fig. 5(c) shows run times for all decompositions. We observe that CP-ALS-NN was significantly faster on both datasets and run times for higher-rank decompositions were significantly higher. Run time could be significantly decreased due to the embarrassingly parallel nature of decompositions [2].

V. RELATED WORK

In a recent survey Papalexakis, Faloutsos, and Sidiropoulos [3] identify space and time modeling in tensor decompositions as a future research challenge. Haass et al. [4] specifically investigated tensors with respect to spatiotemporal data, however, they do not provide a general approach and limit analysis to 3rd-order tensors. They also examine a taxi dataset but consider trajectory data and restrict location to a single tensor mode. Zheng et al. [5] summarize a number of works that use tensor decompositions for spatiotemporal analysis in the context of urban computing. These approaches primarily use tensor decomposition to infer missing values in spatiotemporal data sets related to urban traffic, noise modeling, and activity recommendation. All approaches are restricted to 3rd-order tensors and are reliant on tensor-matrix coupling.

VI. RELATED WORK

Tensor decompositions are a versatile tool when used on spatiotemporal data. Evidence was presented showing capabilities for pattern detection ("nightlife" components), clustering ("neighborhood") components, and anomaly detection (CP-ALS-NN low-weight components). Given the paucity of research on applying tensor decompositions to spatiotemporal data with respect to these three areas it is clear that research must be performed to clarify and bound capabilities.

978-1-5386-3472-1/17/$31.00 ©2017 IEEE

REFERENCES

[1] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," SIAM Review, vol. 51 no. 3, pp. 455-500, September 2009.

[2] M. Baskaran, B. Meister, N. Vasilache and R. Lethin, "Efficient and scalable computations with sparse tensors," 2012 IEEE Conference on High Performance Extreme Computing, Waltham, MA, 2012.

[3] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos. 2016. Tensors for Data Mining and Data Fusion: Models, Applications, and Scalable Algorithms. ACM Trans. Intell. Syst. Technol. 8, 2, Article 16 October 2016.

[4] M. J. Haass, M. H. Van Benthem, E. M. Ochoa, "Tensor Analysis Methods for Activity Characterization in Spatiotemporal Data," Sandia Tech. Report SAND2014-1825, March 2014.

[5] Y. Zheng, L. Capra, O. Wolfson, and H. Yang. "Urban Computing: Concepts, Methodologies, and Applications.,"ACM Trans. Intell. Syst. Technol. 5, 3, September 2014.

[6] M. Baskaran, M. H. Langston, T. Ramananandro, D. Bruns-Smith, T. Henretty, J. Ezick, R. Lethin, "Accelerated Low-Rank Updates to Tensor Decompositions," 2016 IEEE Conference on High Performance Extreme Computing (HPEC), Waltham, MA, 2016.

[7] New York City Taxi and Limousine Commision (TLC) Trip Data, http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml

[8] J. Kepner and J. Gilbert, Eds., Graph Algorithms in the Language of Linear Algebra. SIAM, 2011.

[9] Strang, Gilbert. *Introduction to Linear Algebra*. Fourth Wellesley, MA: Wellesley-Cambridge Press, 2009.

[10] D. Dunlavy, T. Kolda, and W. P. Kegelmeyer, "Multilinear Algebra for Analyzing Data with Multiple Linkages," in *Graph Algorithms in the Language of Linear Algebra*, J. Kepner and J. Gilbert, Eds. SIAM, 2011.

[11] Montgomery County Traffic Violations, https://data.montgomerycountymd.gov/Public-Safety/Traffic-Violations/4mse-ku6q

[12] ArcGIS. http://desktop.arcgis.com/en/.

[13] QGIS. http://www.qgis.org/.

[14] ESRI Shapefile Technical Description. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf.

[15] A. Cichocki, R. Zdunek, A. H. Phan, S. I. Amari. *Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*. John Wiley & Sons, 2009.

[16] E. Chi, T. Kolda, "On Tensors, Sparsity, and Nonnegative Factorizations," in *SIAM Journal on Matrix Analysis and Applications 33.4* (2012): 1272-1299.

[17] ENSIGN. https://www.reservoir.com/research/tech/hypergraphtensor-analysis/

978-1-5386-3472-1/17/$31.00 ©2017 IEEE