Topic modeling for analysis of big data tensor decompositions

Thomas S. Henretty, M. Harper Langston, Muthu Baskaran, James Ezick, and Richard Lethin

Reservoir Labs, 632 Broadway Suite 803, New York, NY, USA, 10012

ABSTRACT

Tensor decompositions are a class of algorithms used for unsupervised pattern discovery. Structured, multidimensional datasets are encoded as tensors and decomposed into discrete, coherent patterns captured as weighted collections of high-dimensional vectors known as components. Tensor decompositions have recently shown promising results when addressing problems related to data comprehension and anomaly discovery in cybersecurity and intelligence analysis. However, analysis of Big Data tensor decompositions is currently a critical bottleneck owing to the volume and variety of unlabeled patterns that are produced. We present an approach to automated component clustering and classification based on the Latent Dirichlet Allocation (LDA) topic modeling technique and show example applications to representative cybersecurity and geospatial datasets.

Keywords: tensor decomposition, topic modeling, clustering, classification, cybersecurity, geospatial, Big Data, LDA

1. INTRODUCTION

Tensor decompositions[1] have been shown to excel in unsupervised pattern discovery when run on large multidimensional datasets. Cybersecurity and geospatial intelligence are two application areas where results have been particularly encouraging, with coherent extraction of both expected and previously undetected patterns of activity.[2345] In these areas, datasets are frequently on the order of millions to billions of multidimensional data points. In order to produce meaningful, interpretable results, datasets at this scale typically need to be decomposed into thousands of component patterns. Manual real-time analysis of these patterns in an operational environment is not feasible. To date, research into tensor decompositions has not focused on automated analysis of results. This work presents a system for analysis automation that utilizes topic modeling, an established technique rooted in natural language processing. With this capability, it becomes feasible to construct automated workflows that leverage tensor methods to highlight specifically interesting patterns in a variety of domains.

In the cybersecurity domain, threat identification and intrusion detection from network traffic data are notoriously difficult problems to solve. Traditional signature-based approaches are often thwarted by the dynamic and evolving nature of modern cyber threats. It is nearly impossible to define signatures for what is or is not normal that generalize across many networks. Even on a given network, expected behaviors might change from day to day. Furthermore, because of the transience of identity in the IP address and port space, it might not be possible to write coherent rules that capture all activities of concern.

The application of cutting-edge data analytics to network traffic logs has struggled to surpass the shortcomings of classical signature-based systems. Supervised machine learning techniques encounter the same key problem — it is not realistic to specify, a priori, normal versus abnormal behavior. Approaches that rely on training a model based on large volumes of historical data are hindered by another issue — because of the sensitive nature of network traffic, there is very little publicly-available training data, and that data is not guaranteed to generalize in a meaningful way to the user's own network.

Ezick et al.[6] decompose a tensor into 100 components or more in order to detect threats. Typically, most, if not all, of these components will represent innocuous network traffic. Still, a trained analyst must review every component and investigate any indicators of suspicious activity. This laborious process must be repeated for every tensor decomposition and severely curtails the effectiveness of tensor decompositions for production

Further author information:
Send correspondence to Thomas Henretty: E-mail: henretty@reservoir.com, Telephone: 1 212 780 0527

network security. There is a time-labor tradeoff where more analysts could examine decomposition results faster, however analyst time is both scarce and expensive. Fewer analysts can be used however response time suffers as the analysts workload increases.

In the geospatial analytics domain, an analyst cannot just observe a known entity and/or look for a known behavior. To be effective, the modern analyst must discover previously unknown, anomalous, or emerging behavior. Further, the amount of intelligence data is overwhelming. The volume of geospatial intelligence data has exploded with the maturation of internet, mobile phone, and social media technology. Hence there is a critical need for effective analytics for geospatial Big Data. Furthermore, in order to identify the "unknown unknowns" in a large, rapidly growing body of data, breakthrough analytics must be developed. In particular, the focus must be on unsupervised techniques that provide analysis without the burden, bias, and inherent limitations of manually categorized known behavior.

Tensor decompositions provide an approach for analyzing both network traffic and geospatial data that has been demonstrated to overcome these shortcomings. Tensor decompositions reveal patterns-of-activity without upfront classification of normal versus abnormal behavior. Further, tensor analysis works in an arbitrary number of dimensions and thus can detect complicated relationships between several data attributes simultaneously.

While tensor decompositions have been proven to be a powerful tool for analyzing large, complex datasets, there remain barriers to their widespread use. Currently, an analyst who is both a domain expert and is trained in tensor mathematics must define the tensor analysis to be performed. Also, while tensor analysis has been demonstrated to be capable of detecting patterns and anomalies in large multidimensional datasets, the analysis currently requires intensive manual inspection of tensor decomposition results.

To automate this analysis we propose the use of topic modeling for clustering and classification of tensor decomposition results. Topic modeling is a widely used technique that has moved beyond its original application in natural language to perform advanced clustering and other analysis of high dimensional datasets in a variety of application domains.[7,8] When applied to tensor decompositions, topic modeling provides the ability to clearly connect tensor decomposition results to patterns automatically discovered in previously seen data.

The major contribution of this paper is the development of the first known application of topic modeling to the problem of clustering and classifying tensor decomposition results. A system for automated analysis, clustering, and classification of tensor decomposition results is presented, and results are demonstrated on two real-world datasets from computer networking and geospatial intelligence domains.

The remainder of this paper is organized as follows: Section 2 provides background on tensor decompositions and introduces the concept of topic modeling as it has been applied to collections of text documents. Our approach to adapting topic modeling to the component analysis problem is described in Section 3. Section 4 provides results from an experimental evaluation in both the cyber and geospatial domains. Conclusions and directions for future work are presented in Section 5.

2. BACKGROUND

2.1 Tensor Decomposition

Tensors (or multidimensional arrays) are a natural fit for representing data with multiple associated attributes such as network traffic logs. Consider a hypothetical log of messages leaving a network. For each message, the log records when it was sent, which IP address sent the message, and which TCP/UDP port that IP address used. This dataset can be formed into a three-dimensional tensor (a data cube) with the dimensions time, IP, and port. For each (time, IP, port) tuple, the tensor contains the count of the number of messages sent at that time, by that IP, and on that port. Similarly, in the geospatial domain, a three-dimensional tensor could be formed from transportation data by capturing source and destination locations (as linearized grid coordinates) along with departure times.

Tensor decompositions are a valuable, mathematically sound set of tools for exploratory analysis of multi-dimensional data and for capturing underlying multidimensional relationships. Tensor decompositions separate input data into patterns called components. Each component represents a latent behavior or correlation from within the dataset. This separation into components occurs without training or upfront specification. The

particular decomposition algorithm used in this paper is the CANDECOMP/PARAFAC (CP) decomposition illustrated in Fig. 1.

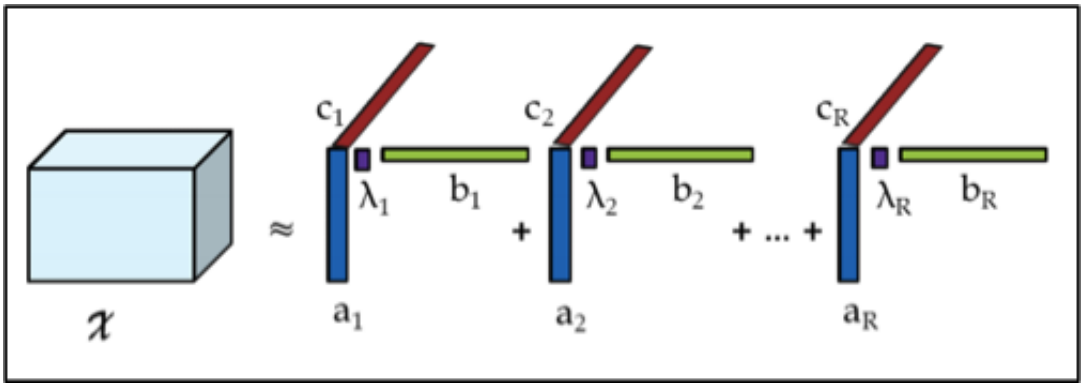


Figure 1: CP tensor decomposition. A tensor is decomposed into a non-unique weighted sum of a pre-defined number of rank-1 components.

The CP decomposition decomposes a tensor into a sum of a predefined number of component tensors. Each component consists of a vector of scores for each dimension of the original data, with one score in the vector for each element of the dimension. A scalar weight (lambda) term associated with each component captures the relative prominence of that component pattern in the original data. Entries within and across dimensions that score highly in a single component are correlated. Intuitively, a tuple comprised of a single high-scoring element from each dimension corresponds, up to a best-fit approximation, to a tuple in the original dataset. Groups of correlated entries in each component identify clusters, patterns, trends, or abnormal events in the data.

At a high level, the method of decomposition involves a constrained model-fitting algorithm that proceeds by gradient descent. In each step, readily parallelizable matrix and vector operations calculate a refinement to the model. A fitness metric tracks progress and, at termination, gives a measure of the final quality of the data approximation. In practice, even coarse approximations almost always succeed in pulling out dominant patterns in the data. For network traffic and geospatial datasets, we use a variant of CP decomposition, namely, CP Alternating Poisson Regression (CP-APR),[9] that supports non-negativity constraints. This ensures that contributing scores are positive and are thus interpretable as activities in the original data.

2.2 Topic Modeling

Topic modeling is a technique originally developed to discover the latent structure in large collections of text documents. A topic modeling algorithm processes a representation of each document in a corpus and produces a topic model that captures clusters of similar words as topics. These topics can then be used to describe collections of related documents.

A large number of algorithms for topic modeling have been proposed. We focus on the seminal Latent Dirichlet Allocation (LDA) algorithm[10] in this work. In LDA, a document is modeled as a high-dimensional vector where each entry represents the count of a particular word in the document. This is often referred to as a "bag-of-words" model. The algorithm produces a topic model with a pre-defined number of topics where each topic is a multinomial distribution over words in the original corpus. From this model the topic mixture of a previously unseen document can be inferred.

3. DESCRIPTION OF APPROACH

Our approach models each component from a tensor decomposition as a document (Section 3.2). Components from multiple decompositions are used to train an LDA model and topics from this model are interpreted as archetypical components (Section 3.3). The topic mixture of new components is inferred and used to both cluster and classify previously unseen components in terms of known patterns (Section 3.4). Finally, we discuss application of the technique in Big Data environments (Section 3.5).

3.1 Introduction to Running Examples

In this section, we describe a typical network traffic and geospatial tensor decompositions that are used to illustrate points for the remainder of Section 3. The network traffic tensor consists of four modes: *timestamp*, *source IP*, *destination IP*, and *destination port*. The mode sizes are, respectively, 1440, 6818, 44916, and 37849. Timestamps are binned by minute and one day of data is considered. Tensor values are a count of duplicate 4-tuples in the dataset after binning.

A rank 100 CP-APR decomposition is performed on this tensor. A single illustrative component, (Component 3) is shown in Fig. 2. Component 3 shows a pattern of regular DNS lookups that occur once an hour, along

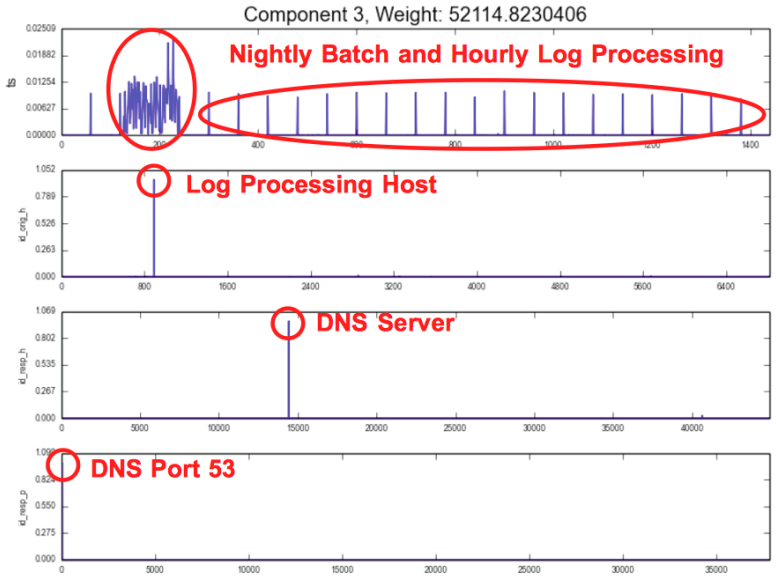


Figure 2: Example network traffic tensor decomposition component.

with a large batch of lookups early in the morning (top row). This pattern is associated with a log server (second row) connecting to a DNS server (third row) on port 53 (fourth row). The pattern represents predictable traffic that occurs every day, thus we would expect it to be represented as a topic when modeling components from many days of decompositions.

The geospatial tensor is taken from the New York City taxi dataset[11] and consists of three modes: *pickup timestamp*, *pickup location*, and *dropoff location*. The mode sizes are, respectively, 168, 35033, and 76082. Timestamps are binned by hour and one week of data is considered. Locations are binned by rounding latitude and longitude values to three decimal places. Tensor values are a count of duplicate 3-tuples in the dataset after binning.

A rank 100 CP-APR decomposition is performed on this tensor. Component 31, shown in Fig. 3, illustrates a pattern of taxi pickups (purple) and dropoffs (teal) primarily occurring on the Upper West Side of Manhattan. The timestamp mode shows the pattern to be strongly associated with weekday traffic as evidenced by the first five groups of nonzero scores in the plot of the timestamp scores beneath the map.

3.2 Component-Document Mapping

The first step in connecting tensor decompositions to topic modeling is representing tensor components as documents. We consider a rank R decomposition of an order-N tensor. Further consider an arbitrary component r from the decomposition with weight w_r and score vectors $a_0, a_1, ..., a_{N-1}$ with lengths $I_1, I_2, ..., I_{N-1}$. We use a mapping where the score vectors of r are concatenated into a single vector to represent a bag-of-words encoded

Figure 3: Example geospatial tensor decomposition component. Component weight is 32132.73. Map contains plots of pickup (purple) and dropoff (teal) locations. Point radii are proportional to score in archetype/decomposition. Bar chart is plot of scores in time mode with Monday 00:00 at the leftmost position and continuing sequentially by hour to Sunday 23:00 at rightmost.

document of length $L = I_1 + I_2 + ... + I_{N-1}$. Each index i of the concatenated vector represents a "word" and the score s_i at each index is transformed to represent an integer word count c_i.

In the running network traffic example, we concatenate the four score vectors of Component 3 (illustrated in Fig. 2) into a single vector with $L = 91023$. Every score s_i (blue lines in Fig. 2) in the vector is multiplied by the weight $w_3 = 52114.82$ and truncated. In the running geospatial example, we similarly concatenate three score vectors into a vector with $L = 111283$ and multiply by weight $w_{31} = 32132.73$ to construct a document.

Transformation of score s_i, where $0 \leq s_i \leq 1$, to nonnegative integer word count c_i is performed by some function $t(s_i)$. In this work, we scale s_i by the component weight w_r and truncate, giving $c_i = \lfloor w_r s_i \rfloor$. This approach gives word counts that are roughly proportional to the number of records in the original dataset represented by the component. An alternative approach would be to scale the scores by some constant k and truncate. This approach gives word counts that are roughly equivalent across components. For this work, we choose to scale by component weight.

An important consideration when mapping components to documents is the semantic consistency of index values across decompositions. It is imperative that each index in each decomposition represents the same data value. For instance, in our running example, if the source IP of the log host is mapped to index 5, it **must** be mapped to index 5 in all components used for topic model training (Section 3.3) and inference (Section 3.4)

3.3 Topic Model and Interpretation

To construct a topic model, we gather a collection of components for training the model. Since our goal is to automatically cluster and classify the results of a tensor decomposition, we train the model with components from multiple decompositions. In our running example, we would train with multiple decompositions each representing one day of traffic.

Given a collection of components represented as documents, we specify some number of topics T and train an LDA model. In the trained model, each of the T topics is represented by a vector where each entry represents a probability p such that the $\sum_{i=0}^{L-1} p_i = 1$.

Recall that in a tensor decomposition component, scores sum to 1 *in each mode* as opposed to *across all modes* as in a topic. We can, however, treat topics as *archetype* components by splitting the topic into N vectors corresponding to the modes of the original component. We adjust the probabilities in each mode such that relative proportions between probabilities are the same as in the topic are the probabilities for each mode sum to 1. By converting topics into archetypes, we are able to compare them directly to components.

For the network traffic example, We trained a 100 topic LDA model with multiple decompositions representing other days of data taken from the same network as our running example. We expected to see an archetype similar to Component 3 (as shown in Fig. 2) because of the predictable and consistent pattern-of-activity represented by the component. The topic model produced the archetype shown in Fig. 4. This archetype, derived from a single topic in an LDA model, matched our expectations and is a generalized version of Component 3.*

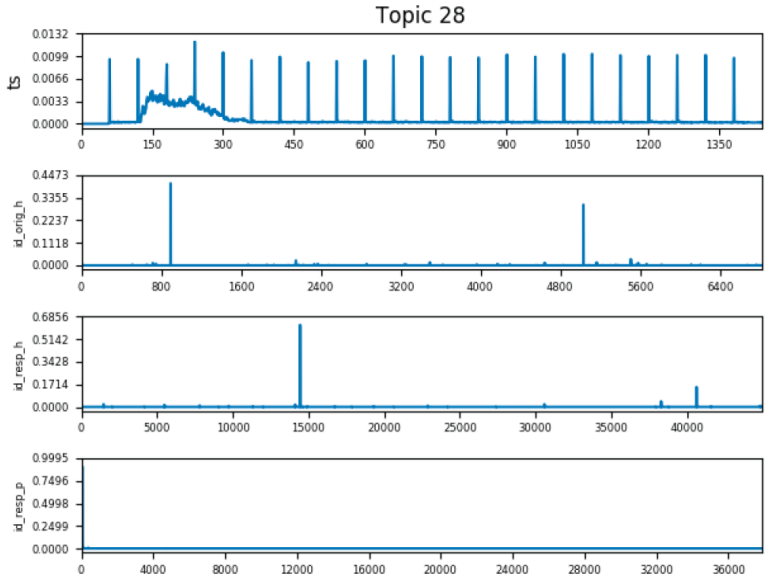


Figure 4: Example network traffic archetype.

For the geospatial example, We trained a 100 topic LDA model with multiple decompositions representing previous weeks of data taken from the taxi dataset. An archetype similar to Component 31 (as shown in Fig. 3) was produced by the topic model and is shown in Fig. 4. This archetype shows largely identical pickup locations, dropoff locations, and timestamps as Component 31 with small variations in the relative magnitude of scores/probabilities.

*The extra source IP in the topic/archetype 28 is another machine that also performs hourly log processing.

Figure 5: Example geospatial archetype.

3.4 Inference, Clustering, and Classification

We would like to automatically match previously unseen components, such as Component 3 in Figure 2, to known behavior, for example the archetype shown in 4. LDA provides inference, which is a capability that makes this automatic matching possible. After inference, it is trivial to automatically cluster previously unseen components around known patterns-of-activity as represented by topics. These topics may optionally be labeled, thus enabling classification along with clustering.

First, via LDA, we infer the topic mixture of one or more components. A topic mixture tells us which topics each component most resembles, and the relative magnitude of the resemblance. Inference also has the effect of projecting high-dimensional components into a lower dimensional topic space. In our running example, inference projects a 93,000+ dimension component into a 100-dimensional topic space. Inference on Component 3 led to a mixture that was over 99% Topic 28.

After inference, we may use traditional clustering algorithms (e.g., k-means, agglomerative hierarchical clustering) on the lower-dimensional topic space. In this work, we use a naive clustering algorithm that assigns components to a cluster corresponding to the topic with the highest proportion in the mixture. In our running example Component 3 would clearly be assigned to the Topic 28 cluster.

Finally, topics/archetypes can be manually examined and labeled. The number of topics to label will be substantially lower than the number of components used to train the model. In our running network traffic example, suppose we trained a topic model with 30 days of rank 100 decompositions. The 3,000 components produced by these decompositions, if they were to be used in an operational environment, would have to be

manually examined and labeled. Instead, a topic model with 100 topics can be trained, and the resulting 100 topics labeled. Future decompositions can then be clustered and meaningfully described in terms of those 100 labeled topics. Further, since inference on documents used to train an LDA model produces meaningful results, the original 3,000 components can also be clustered and classified based on the topic model they were used to train.

3.5 Big Data Considerations

When adapting our approach to Big Data, the scale of both the data and the algorithms used must be considered. Big datasets can exceed trillion-scale while current tensor decomposition algorithms are limited to billion-scale datasets.[12] Clearly, big datasets must be partitioned into (at most) billion-scale chunks and decomposed. Cybersecurity and geospatial intelligence applications are focused on identifying patterns-of-activity thus it makes sense to partition these datasets into manageable chunks representing a fixed time period.

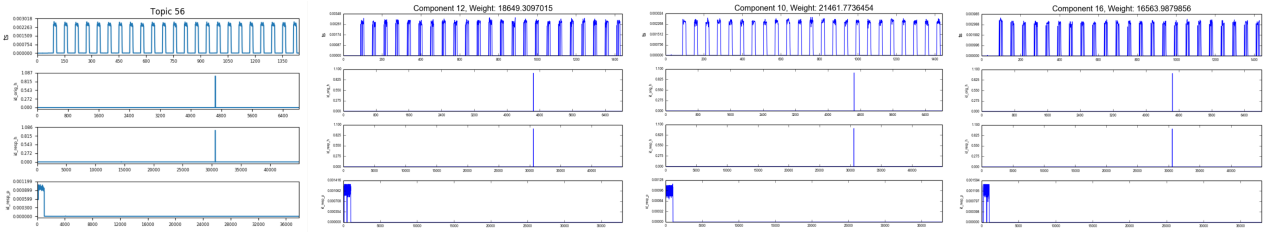
Tensor decomposition substantially reduces the scale of data used in topic modeling. For example, decomposition of a one billion entry tensor with $R = 10^4$ (number of "documents") and $L = 10^7$ (number of "words") yields a a corpus and a vocabulary that are relatively small with respect to the capabilities of modern LDA implementations, which are capable of processing over 10^7 documents with vocabularies of over 10^9 words on a commodity shared memory system.[13][14]

4. EXPERIMENTAL EVALUATION

Experiments were conducted using the CP-APR decomposition included in the ENSIGN tensor decomposition package[15] and the LDA implementation provided by the Gensim topic modeling package.[16][17] Network data was collected over a period of three months on a small office network using the R-Scope implementation of the Bro network monitoring tool.[18][19] The geospatial data used was New York City yellow cab trip report data[11] from May through July of 2015. All decompositions were performed using default parameters. All LDA models were trained for 100 topics using 100 LDA iterations and 10 passes over the corpus. All other LDA parameters were left at default values.

4.1 Network Data

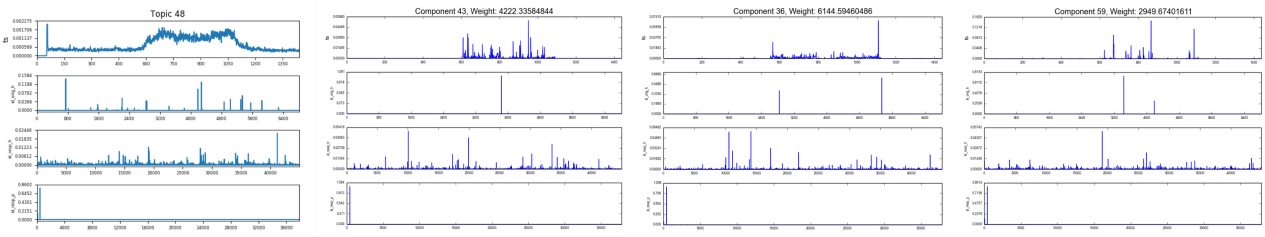
Network traffic tensors were constructed in an identical manner to the running example described in Section 3.1. Rank 100 CP-APR decompositions were performed on 56 tensors, with each tensor representing one day of traffic. LDA models were trained using 42 days of decompositions and evaluated using the remaining 14 days. The training corpus was 4,200 documents with a vocabulary of 91,023 words.



(a) (b) (c) (d)

Figure 6: Hourly port scan. Topic archetype is shown in (a). Representative components with topic at least 98% of mixture shown in (b), (c), and (d). Each chart plots decomposition score vectors of, from top to bottom, the *timestamp*, *source IP*, *destination IP*, and *destination port* modes

Figure 6 illustrates members of a cluster with Topic 56 making up more than 97% of the inferred mixture. This cluster represents an intentionally seeded port scan where one source IP connects to ports 1-1024 of one



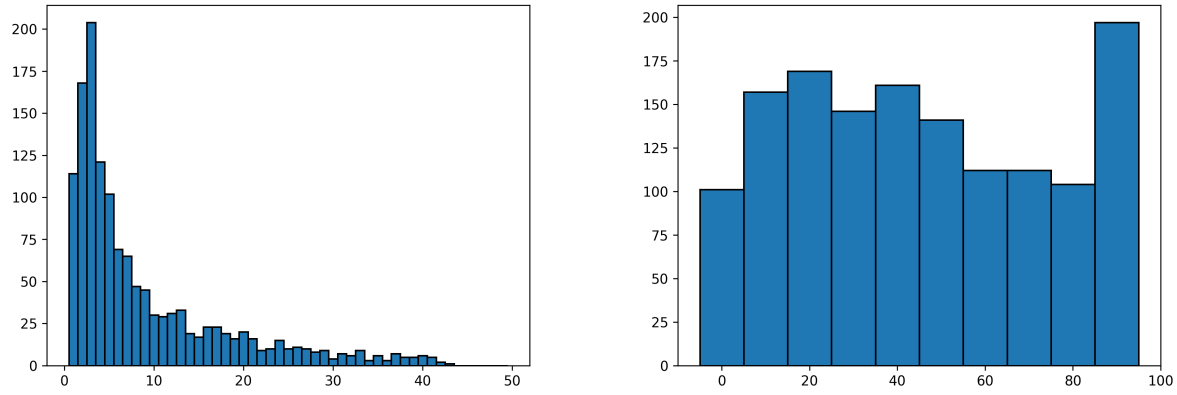
(a) (b) (c) (d)

Figure 7: Business hours web browsing. Topic archetype is shown in (a). Representative components with topic at least 90% of mixture shown in (b), (c), and (d).

destination IP once per hour. The archetype of Topic 56 is nearly identical to actual decomposition components from the test data.

Figure 7 shows another cluster where the dominant topic 48 made up greater than 98% of the inferred mixture. This cluster represents business hours internet traffic from employee machines. The archetype shows a sustained burst of high scores during business hours with traffic originating from multiple IP addresses destined for a large number of other IP addresses. This traffic is primarily to port 443 (encrypted websites) with a smaller amount to port 80 (unencrypted websites).

Figure 7 illustrates the ability of inference to and clustering to identify components that are generalized by a topic. The components shown in Fig. 7(b)-(d) do not match the archetype exactly. In all cases the components emphasize different times during business hours and highly score only a subset of the machines seen in the archetype. Despite these differences, it is reasonable to assign all components to the topic 48 cluster and further, it is reasonable to label this cluster "Business hours internet traffic."



(a) (b)

Figure 8: Network traffic topic mixture composition for all test components. (a) Nonzero topics per component histogram. (b) Mixture proportion (percent) of dominant topic histogram.

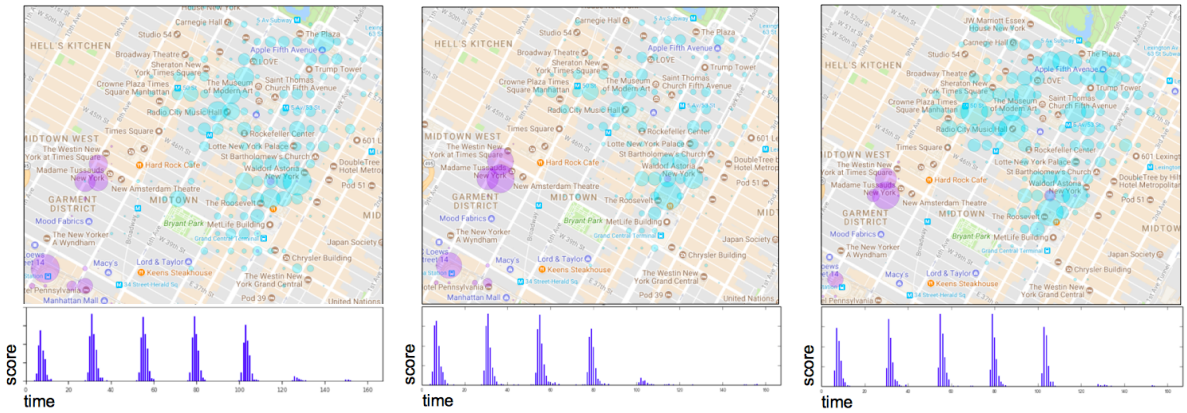
The histograms in Fig. 8 provide insight into the composition of inferred topic mixtures for all test components. Figure 8(a) shows that approximately half of topic mixtures are composed of one to five nonzero topics. Components with lower topic counts were easier to interpret in terms of known behavior, whereas components with higher topic counts generally represented patterns-of-activity that had not been previously seen. Regardless of topic count, components with a single dominant topic were readily identifiable as specific instance of the

dominant topic when it was as little as 60% of the mixture. Figure 8(b) shows the distribution of the mixture proportion for the single largest topic, or *primary topic*, of each mixture. A large number of components have primary topics accounting for over 60% of the mixture. There are, however, a substantial number of mixtures with the primary topic below 60% representation. Taken together, both of these histograms suggest that an area for future research is to maximize the number of topics that either 1) have a single topic representing a substantial proportion of the mixture (thus representing known patterns) or 2) have a large number of topics with none that are particularly strong (thus representing anomalous patterns).

4.2 Geospatial Data

Seven tensors were constructed from taxi data with each tensor representing one week of fares. Tensor modes were *pickup timestamp*, *pickup location*, and *dropoff locations*. Timestamps were binned by hour and locations were binned by rounding decimal latitude and longitude to three decimal places. Tensor values were a count of identical tuples in the dataset.

Rank 100 CP-APR decompositions were performed on each tensor. LDA models were trained using five weeks of decompositions from May and June and evaluated using the remaining two weeks from June and July. The training corpus was 500 documents with a vocabulary of 111,283 words.



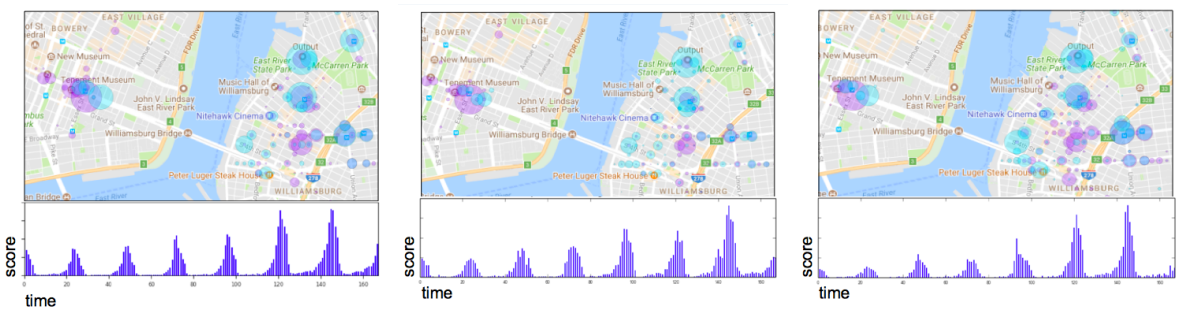
(a) (b) (c)

Figure 9: Traffic from Penn Station / Port Authority bus terminal to Midtown Manhattan. Topic archetype is shown in (a). Representative components with topic over 90% of mixture shown in (b) and (c).

Inference on test components consistently produced mixtures with reasonable clustering results. Clustering by largest topic proportion led to high-quality clusters when the topic proportion was over 90% of the mixture. This is shown in Fig. 9. This figure shows weekday traffic from Penn Station and the Port Authority bus terminal to locations in Midtown Manhattan. In Fig. 9(b) we see a lower scores than expected on Friday. This is likely due to the fact that Friday, July 3 was a United States federal holiday leading to a reduced number of taxi fares to midtown employers. Otherwise, the time modes are clearly similar. The distribution of locations in the archetype is clearly similar to those in the two test components with only minor variations in score magnitude.

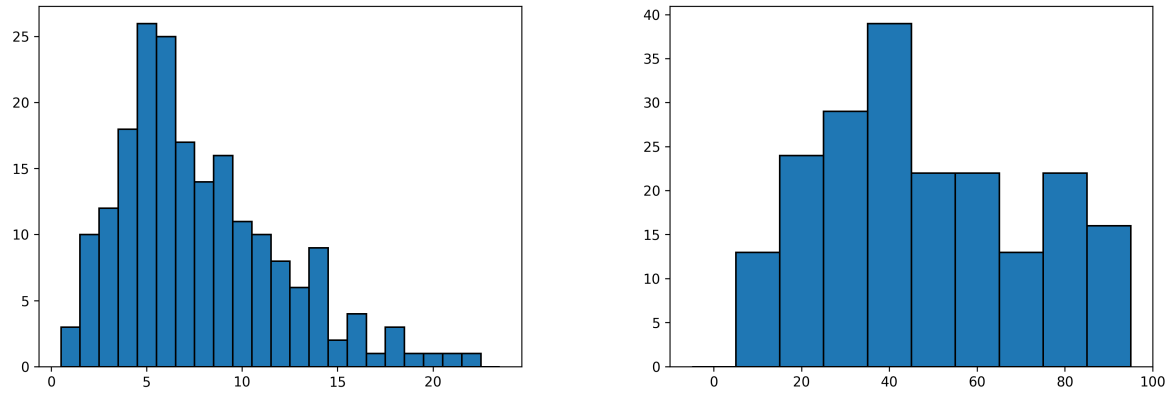
Figure 10 shows another high-quality clustering of components representing traffic between Manhattan and Brooklyn. Again, we see a reduction of time mode scores on the Friday. Otherwise, the same pattern of low scores on Monday through Thursday with higher scores on Friday and Saturday is evident. Again we see a similar distribution of locations with minor variations in score magnitude.

Topic mixture histograms for the taxi dataset, shown in Fig. 11, indicate that inferred mixtures generally consist of more topics with fewer dominant ones compared to the network traffic histograms in Figure 8. This is partially explained by the major holiday falling towards the end of the week in the first test dataset. High topic count /anomalous components were found to represent behavior around the holiday, including increased traffic near popular fireworks viewing locations on the Fourth of July.



(a) (b) (c)

Figure 10: Traffic between the Lower East Side of Manhattan and Williamsburg, Brooklyn. Topic archetype is shown in (a). Representative components with topic over 90% of mixture shown in (b) and (c).



(a) (b)

Figure 11: Taxi dataset topic mixture composition for all test components. (a) Nonzero topics per component histogram. (b) Mixture proportion (percent) of dominant topic histogram.

4.3 Discussion

As evidenced by the cluster results shown in Section 4.1 and Section 4.2 robust clusters of similar tensor decomposition components can be obtained by constructing a topic model and using a simple clustering strategy. The technique is resilient to overfitting of topics to components, as evidenced by components containing business internet traffic and holiday taxi traffic being sufficiently generalized by clearly defined topics.

Figures 6, 7, 9, and 10 all show components with single dominant topics in their inferred topic mixture, but this was not always the case. Many components were described as mixtures of multiple topics with no clear dominant topic. In such cases, there were multiple reasonable interpretations of the components. When the component was a mixture of a small number of topics (2 to 4) with none clearly dominant, it could be interpreted as a combination of known patterns-of-activity. When the component was a mixture of a large number of topics with none clearly dominant, it was reasonable to interpret the component as representing anomalous behavior.

It was clear that the most useful components were either those that had a single dominant topic in the inferred mixture or those that were mixtures of a large number of weakly represented topics. In the first case, the components could be readily identified as representing known behavior; in the second case, the components were readily identified as representing anomalous behavior.

5. CONCLUSION

The use of topic modeling to accelerate classification of patterns isolated by tensor decomposition methods represents a disruptive combination of powerful existing technologies. This combination enables unsupervised and automated analysis of high-volume high-dimensional data. The result is an effective method for clustering and classification of patterns-of-activity in high-dimensional datasets that sidesteps the "curse of dimensionality" issues encountered by traditional distance- and density- based approaches such as k-means clustering and DBSCAN. The approach enables the automation of workflows that go directly from data collection and tensor formation, through decomposition, to identification of actionable insight. Further research is needed to refine the quality of clusters produced and may focus on the topic modeling algorithm, the clustering algorithm, and post-processing of clustering output.

REFERENCES

[1] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," *SIAM Review* **51**(3), 455–500 (2009).

[2] Afshar, A., Ho, J. C., Dilkina, B., Perros, I., Khalil, E. B., Xiong, L., and Sunderam, V., "Cp-ortho: An orthogonal tensor factorization framework for spatio-temporal data," in [*Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*], *SIGSPATIAL'17*, 67:1–67:4, ACM, New York, NY, USA (2017).

[3] Mao, H.-H., Wu, C.-J., Papalexakis, E. E., Faloutsos, C., Lee, K.-C., and Kao, T.-C., "Malspot: Multi 2 malicious network behavior patterns analysis," in [*Pacific-Asia Conference on Knowledge Discovery and Data Mining*], 1–14, Springer (2014).

[4] Bruns-Smith, D., Baskaran, M. M., Ezick, J., Henretty, T., and Lethin, R., "Cyber security through multi-dimensional data decompositions," in [*2016 Cybersecurity Symposium (CYBERSEC)*], 59–67 (2016).

[5] Henretty, T., Baskaran, M., Ezick, J., Bruns-Smith, D., and Simon, T. A., "A quantitative and qualitative analysis of tensor decompositions on spatiotemporal data," in [*2017 IEEE High Performance Extreme Computing Conference (HPEC)*], 1–7 (2017).

[6] Ezick, J., Baskaran, M., Bruns-Smith, D., Commike, A., Henretty, T., Langston, M. H., Ros-Giralt, J., and Lethin, R., "Discovering deep patterns in large-scale network flows using tensor decompositions," in [*FloCon*], (2017).

[7] Liu, L., Tang, L., Dong, W., Yao, S., and Zhou, W., "An overview of topic modeling and its current applications in bioinformatics," *SpringerPlus* **5**(1), 1608 (2016).

[8] Wang, X. and Grimson, E., "Spatial latent dirichlet allocation," in [*Advances in neural information processing systems*], 1577–1584 (2008).

[9] Chi, E. C. and Kolda, T. G., "On Tensors, Sparsity, and Nonnegative Factorizations." arXiv:1304.4964 [math.NA] (December 2011).

[10] Blei, D. M., Ng, A. Y., and Jordan, M. I., "Latent dirichlet allocation," *J. Mach. Learn. Res.* **3**, 993–1022 (2003).

[11] New York City Taxi and Limousine Commission, "Trip Record Data." http://www.nyc.gov/html/tlc/html/about/trip_record_data.shtml. (Accessed: 28 March 2018).

[12] Jeon, I., Papalexakis, E. E., Kang, U., and Faloutsos, C., "Haten2: Billion-scale tensor decompositions," in [*IEEE 31st International Conference on Data Engineering (ICDE)*], 1047–1058 (2015).

[13] Yu, H.-F., Hsieh, C.-J., Yun, H., Vishwanathan, S., and Dhillon, I. S., "A scalable asynchronous distributed algorithm for topic modeling," in [*International World Wide Web Conference (WWW)*], (2015).

[14] Yuan, J., Gao, F., Ho, Q., Dai, W., Wei, J., Zheng, X., Xing, E. P., Liu, T.-Y., and Ma, W.-Y., "Lightlda: Big topic models on modest computer clusters," in [*Proceedings of the 24th International Conference on World Wide Web*], 1351–1361, International World Wide Web Conferences Steering Committee (2015).

[15] Reservoir Labs, "ENSIGN." https://www.reservoir.com/research/tech/tensor-analysis/. (Accessed: 28 March 2018).

[16] Řehůřek, R. and Sojka, P., "Software Framework for Topic Modelling with Large Corpora," in [*Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*], 45–50 (2010).

[17] Hoffman, M., Bach, F. R., and Blei, D. M., "Online learning for latent dirichlet allocation," in [*Advances in neural information processing systems*], 856–864 (2010).

[18] Reservoir Labs, "R-Scope Advanced Threat Detection." https://www.reservoir.com/product/r-scope/. (Accessed: 28 March 2018).

[19] Paxson, V., "Bro: a System for Detecting Network Intruders in Real-Time," *Computer Networks* **31**(23-24), 2435–2463 (1999).