

A constrained sparse non-negative matrix factorization algorithm for detecting traffic signatures in large mobility datasets

Vaibhav Karve, Derrek Yager, Marzieh Abolhelm, Dan Work *Member, IEEE*, Richard B. Sowers *Member, IEEE*

Abstract—This article proposes a data-driven method to study large-scale matrices of urban traffic information. The input to our model is a traffic data matrix where rows correspond to times and columns correspond to road segments, and the entries are traffic counts. A constrained sparse non-negative matrix factorization (constrained-SNMF) problem is formulated in order to identify a collection of traffic behavioral “signatures” which describe various traffic conditions. To allow for a more temporally scale-free interpretation of the signatures, a constraint normalizes the signatures, and the matrix factorization cost function is augmented by a term which rewards sparsity. As a result, traffic behavior on each road is expressed with a small number of signatures. Nonnegative matrix factorization is known to be NP-hard. We propose here a numerical procedure to solve the resulting constrained-SNMF problem. The technique is then applied to a 2011 New York City traffic data set, and anomalous traffic patterns are interpreted from the results.

The contributions of this are threefold. We introduce and study the normalization constraint. We secondly introduce a convenient scale-invariant metric with which to measure sparsity. Secondly, we demonstrate an implementation of the measure of sparsity originally formulated by [1] into a multiplicative algorithm. Thirdly, we propose a scale-free framework for studying the trade-off between the error of the factorization, the sparsity, and the dimension of the proposed low-rank factorization. This may be useful in city-to-city comparisons.

Index Terms—Traffic, Normalization, Sparse Non-negative Matrix Factorization

I. INTRODUCTION

A. Motivation

TRAFFIC management is one of the persistent challenges of the modern industrialized world. It simultaneously reflects both a critical infrastructural necessity and a problem with a wide range of scales and interactions. Over the past two decades, floating-car data (e.g., from GPS-equipped taxis and other vehicles) has become an important source to obtain real-time and large-scale city traffic information [2]. These data streams can be used to manage traffic signals, influence equilibrium traffic states [3]–[14], and optimally route traffic, in order to make effective and informed decision-making

possible. The challenge however, is the missing values in the data obtained by the floating-car technologies. We use non-negative matrix factorization to overcome the imperfection of the data, and analyze the underlying characterization of city traffic.

B. Problem Statement and Contribution

City traffic maintains structural and behavioral complexities. We investigate matrix factorization [15] to identify robust behavioral signatures in traffic, and describe various traffic conditions. Our starting point is a traffic matrix D , where $D_{t,\ell}$ represents traffic count on link ℓ (i.e., road segment) at time t . We could also consider speeds instead of counts with some modifications. Our algorithm outputs two matrices W and H . The columns of W represent *behavioral signatures*, and the rows of H are *coefficients*. More precisely, if we fix a link ℓ , the column $D_{\cdot,\ell}$ denotes the behavior of the traffic count on ℓ over time (i.e., $D_{t,\ell}$ is the traffic count at time t on link ℓ). We want to write any $D_{\cdot,\ell}$ as a linear combination of N signatures (i.e., column vectors indexed by time). In other words, we want to approximate

$$D_{\cdot,\ell} = H_{1,\ell}W_{\cdot,1} + H_{2,\ell}W_{\cdot,2} + \cdots + H_{N,\ell}W_{\cdot,N}. \quad (1)$$

The challenge, of course, is to optimally identify this decomposition over all W 's and H 's.

A signature can, for example, represent heavy traffic volumes during the morning rush hour, medium traffic for the rest of the workday, and light traffic in the evening. This would mean a column which has large entries in the rows corresponding to morning rush hour, moderate entries in the rows corresponding to the rest of the workday, and small entries for the remaining rows. Each signature is a time-series for the entire year, and hence, we do not need to rely on weekly periodicity. For example, the signatures can capture traffic anomalies during holidays, extreme weather events, or infrastructure failures, such as blackouts.

The columns of H , i.e., the coefficients, represent the way in which the links are decomposed into distinct signatures. Ideally, we would like to decompose the traffic structure on all links as a linear sum of only a small number of signatures, resulting in a sparse H . There are subtle issues involved in imposing sparsity vs. measuring it, and these are explored further in Section II-D.

Motivated by the complexity of traffic data, we address several structural issues:

The authors acknowledge the Program for Interdisciplinary and Industrial Internships at Illinois (PI4) and the Illinois Geometry Laboratory (IGL). The many IGL students who have made invaluable contributions to this work are: Raghav Bakshi, James Kerns, Xinyi Li, Xinyu Liu, Yicheng Pu, Gabriel Shindnes, Haozhe Wang, Jing Wang, Ziyang Wang, Yu Wu, Zeyu Wu, Bin Xu, and Dajun Xu. The authors would also like to thank the Siebel Energy Institute for its support of this work. This work was also supported by a grant from the Siebel Energy Institute. This material is based upon work supported by the National Science Foundation under Grant Number CMMI 1727785.

- Fidelity of the approximation; how does one measure the error, and how does the error depend on the rank of the approximation?
- Missing data values; data is necessarily incomplete since data is not perfectly collected all of the time.

The results can be useful in structuring analyses of traffic data in other cities.

C. Related Works

A review of the literature suggests three main challenges in the analysis of urban traffic:

- Traffic data compression
- Traffic data description
- Traffic data prediction

For the scope of this article, we will not visit traffic prediction models; however, the literature on the combination of matrix factorization and location recommendation algorithms, maintains helpful insights on this subject [16]. Matrix factorization is a well-established tool for breaking data into patterns. Non-negative matrix factorization has been applied to a wide range of problems, like text data mining [17], [18], gene expression [19]–[23], micro-array comparative genomics hybridization [24], functional characterization of gene lists [25], facial images [1], [26], and urban and network traffic analysis [2], [27]. Various extensions of non-negative matrix factorization have been made to impose sparsity, either on both W and H [1], [28]–[31], or only on H [18], [21]. Our work follows some of the thoughts of non-negative matrix factorization by Lee and Seung in [32] and the sparse non-negative matrix factorization by Kim and Park in [31].

Different matrix factorization techniques work well toward different intentions. *Principal Component Analysis* (PCA) [33], *Independent Component Analysis* (ICA) [34], and *Non-negative Matrix Factorization* (NMF) [35] are some of the most eminently used algorithms to compress and describe traffic data. PCA provides an effective low-rank representation of data, by letting each principal component be approximated by a linear combination of all basis components. However, PCA allows the entries of the factorized components to be of arbitrary sign. Since we are dealing with traffic, we require positive components in order for the decomposition to remain interpretable. Unlike PCA, which assumes Gaussianity of the data, ICA works well for finding independent sources, given that the subcomponents of data are non-Gaussian and contain statistically independent signals. This assumption is ill-suited for learning component-based representations, since various components are likely to occur together. NMF provides an alternative approach for traffic decomposition, given the condition that the data components must be non-negative. In other words, by forcing the factorized matrices to be non-negative, NMF reveals underlying data structures with the same direction of correlation.

D. Outline

We give a mathematical formulation of our efforts in Section II, writing a low-rank approximation as a minimization

problem, and note that the formulation allows for missing data. We construct a penalty term that encourages *sparsity* in the H matrix of signature coefficients. An iterative algorithm is introduced to solve the resulting constrained sparse non-negative matrix factorization problem in Section III. Section IV is dedicated to a case study from data in New York City. In working through this case study, we identify and address some natural numerical stability questions. The analysis of the New York City dataset is continued in Section V. The supporting source code for this work is published at [36].

There are several specific contributions in this work. Primarily, we include a *normalization* which allows us to interpret our decomposition in terms of how a “unit mass” distributes its occupancy throughout the time interval. Normalization allows us to make scale-free comparisons. We secondly introduce a scale-invariant way to measure sparsity in Equations (10) and (11). This allows us to think precisely about sparsity and compare decompositions. Thirdly, we propose a scale-free framework for quantifying the tradeoff between the error of the factorization, the sparsity, and the dimension of the proposed low-rank factorization; see Figures 11 and 12 and the discussion of Subsection IV-D. This may provide a useful way to think about city-to-city comparison of traffic complexity.

II. SETUP

Our starting point in this article is a data set containing traffic counts (one might also consider traffic speeds) on individual links at individual times. The goal is to understand behavioral structures and patterns in this data. In a large city, there may be many links (in New York City, which we consider in Section IV, there are over 260,000 links), and we may be interested in time fluctuations over an entire year (in our analysis of Section IV, we consider data from all of 2011). We take advantage of the calculations of [37] and [38] to recover an estimate of these traffic counts from origin-destination pairs for taxi data. Of course we need to choose the granularity of the time index t . Each time t represents a time interval (or bin); we want these intervals to be small enough to capture meaningful fluctuations, but if they are too small, the individual bins won’t have enough data to be statistically meaningful. The computations of [37] and [38] give us hourly estimates of traffic, so we will take that as our granularity. In our calculations of Section IV, there will thus be 8760 time instants in 2011.

We organize these traffic counts as a matrix, letting $D_{t,\ell}$ denote the traffic count in link ℓ in hour t ; the rows of D correspond to time and the columns correspond to link. The matrix D can, in real problems, be quite large, and have missing entries; i.e., if D represents speeds, no speed information is available if there are no cars on the link. One would like to identify patterns in D , to study its underlying structure, and fill in the missing entries.

Our goal here is to develop a low-rank approximation of D (similar to principal component analysis). This low rank approximation is mathematically a *matrix factorization*; i.e., we want to write $D \approx WH$ for two lower-dimensional matrices W and H (of the right sizes). We want to build into

this low-rank approximation the fact that traffic counts must be non-negative, so we want W and H to have only non-negative entries.

As a final component of decomposing D into patterns, we want to algorithmically encourage *sparsity* in H . Although we may have a large number of possible elementary signatures (as described in the discussion of Subsection I-D), we want to write the behavior on each link as a linear combination of relatively few signatures. We will develop a penalization procedure to enforce this.

A. Matrix Factorization

We use matrix factorization to decompose a matrix $D \in \mathbb{R}^{T \times L}$, where $T, L \in \mathbb{N}$. This allows us to write

$$\underbrace{D}_{T \times L} \approx \underbrace{W}_{T \times N} \times \underbrace{H}_{N \times L}, \quad (2)$$

where N is fixed and $N \leq \min\{T, L\}$. We will refer to N as the *rank of the factorization*. In the case of non-negative matrix factorization, we have $D \in \mathbb{R}_+^{T \times L}$ and therefore look for factors $W \in \mathbb{R}_+^{T \times N}$ and $H \in \mathbb{R}_+^{N \times L}$, respectively. The non-negative property is important when the data matrix D is constructed from non-negative traffic quantities such as speed, density, or travel time.

For $(t, \ell) \in \{1, 2, \dots, T\} \times \{1, 2, \dots, L\}$, we use the standard notation

$$(WH)_{t,\ell} \stackrel{\text{def}}{=} \sum_{n=1}^N W_{t,n} H_{n,\ell}.$$

For any matrix $A \in \mathbb{R}^{R \times C}$, denote the r th row and c th column respectively as

$$A_{r,\cdot} \stackrel{\text{def}}{=} [A_{r,1} \quad A_{r,2} \quad \cdots \quad A_{r,C}], \quad A_{\cdot,c} \stackrel{\text{def}}{=} \begin{bmatrix} A_{1,c} \\ A_{2,c} \\ \vdots \\ A_{R,c} \end{bmatrix}.$$

We will denote the Hadamard product and quotient respectively of two matrices of the same size $A, B \in \mathbb{R}^{R \times C}$ as

$$(A \otimes B)_{i,j} = A_{i,j} B_{i,j},$$

$$(A \oslash B)_{i,j} = \frac{A_{i,j}}{B_{i,j}},$$

for $i \in \{1, 2, \dots, R\}$ and $j \in \{1, 2, \dots, C\}$. Note these definitions are simply element-wise multiplication and division, respectively.

Fixing the rank of the factorization N , the goal of non-negative matrix factorization is to minimize

$$\|D - WH\|_F^2, \quad (3)$$

over all $W \in \mathbb{R}_+^{T \times N}$ and $H \in \mathbb{R}_+^{N \times L}$, where $\|\cdot\|_F^2$ denotes the square of the Frobenius norm; i.e., the sum of squares of entries of a matrix. There is a trade-off in that a larger N allows lower error, but poorer compression of the data in D . Each unit increment in rank requires the storage of T additional values in W and an additional L values in H .

The cost function of (3) is convex in W and H each, but not jointly [32]. Secondly, nonnegative matrix factorization is known to be NP-hard [39].

B. Missing Data Values

In many practical scenarios the data matrix D might have missing values. For example, if $D_{t,\ell}$ is the speed on link ℓ at time t measured from floating cars, then $D_{t,\ell}$ is missing if there are no GPS-equipped vehicles on that link at that time. For specificity, let's say that $D_{t,\ell} = \text{NaN}$ in this case (NaN is short for not-a-number). Matrix factorization algorithms are useful for filling in these “holes” by exploiting some low-rank structure in the original data. To construct the low-rank factorization without these entries, define

$$\mathcal{N} \stackrel{\text{def}}{=} \{(t, \ell) \in \{1, 2, \dots, T\} \times \{1, 2, \dots, L\} : D_{t,\ell} \neq \text{NaN}\}.$$

For $A \in \mathbb{R}^{T \times L}$, let's then define a masked matrix $[A]_{\mathcal{N}} \in \mathbb{R}^{T \times L}$ by setting

$$([A]_{\mathcal{N}})_{t,\ell} \stackrel{\text{def}}{=} \begin{cases} A_{t,\ell} & \text{if } (t, \ell) \in \mathcal{N} \\ 0 & \text{else} \end{cases}. \quad (4)$$

We note that the map $A \mapsto [A]_{\mathcal{N}}$ is linear.

Taking into account these missing values, we can modify the penalty in (3) to be

$$\begin{aligned} \mathcal{E}_o(W, H) &\stackrel{\text{def}}{=} \sum_{(t,\ell) \in \mathcal{N}} (D - WH)_{t,\ell}^2 \\ &= \|[D - WH]_{\mathcal{N}}\|_F^2. \end{aligned} \quad (5)$$

In (5), the missing entries do not contribute to the Frobenius norm of the factorization error. Since (4) is linear, \mathcal{E}_o is quadratic in both W and H .

C. Sparsity and Normalization

We want H to be *sparse* (i.e., to have many zeroes), meaning that each column of D can be represented by a small number of signatures. Rather than forcibly restricting the rank N of the approximation in (2), we can consider a larger N and introduce a penalty which rewards sparsity. *Sparse Nonnegative Matrix Factorization* (SNMF), as seen in [40], fixes the positive parameter β and minimizes the following:

$$\mathcal{E}_{\beta,\eta}(W, H) \stackrel{\text{def}}{=} \mathcal{E}_o(W, H) + \beta \sum_{\ell=1}^L \|H_{\cdot,\ell}\|_1^2 + \eta \|W\|_F^2. \quad (6)$$

Informally, N represents the size of the “universe” of available signatures, while a large value of β in (6) should result in any link being represented by only a small number of signatures in this universe.

We also want to enforce a constraint that each column of W sums to 1 (i.e., has L_1 -norm of 1). This gives us a *relative* occupancy count; allowing us to better understand how the traffic count in the signature is temporally broken down (i.e., by time, week, and season). Questions of the total count are then transferred to H . We first find an approximate minimum

$(\widehat{W}, \widehat{H})$ of $\mathcal{E}_{\beta, \eta}$. We then construct a new pair (W', H') such that

$$W'H' = \widehat{W}\widehat{H}, \quad (7)$$

and each column of W' has L_1 -norm of 1. Namely, we define

$$\begin{aligned} W'_{:,n} &\stackrel{\text{def}}{=} \widehat{W}_{:,n} / \|\widehat{W}_{:,n}\|_1 \\ H'_{n,\cdot} &\stackrel{\text{def}}{=} \widehat{H}_{n,\cdot} \cdot \|\widehat{W}_{:,n}\|_1, \end{aligned} \quad (8)$$

where $\|\widehat{W}_{:,n}\|_1$ denotes the L_1 norm of the n -th column of \widehat{W} . Thus we have

$$\|W'_{:,n}\|_1 = 1 \quad (9)$$

for all $n \in \{1, 2, \dots, N\}$, and (7) holds. Perhaps $\mathcal{E}_{\beta, \eta}(W', H') \geq \mathcal{E}_{\beta, \eta}(\widehat{W}, \widehat{H})$, in which case (W', H') is not necessarily an improvement of (W, H) ; nevertheless we use it to generate the next pair (W, H) .

Mathematically, $(\widehat{W}, \widehat{H})$ defines a point in the space

$$\mathbf{S} \stackrel{\text{def}}{=} \mathbb{R}_+^{T \times N} \times \mathbb{R}_+^{N \times L}$$

(representing the allowable spaces for W and H). The rescaling of (8) corresponds to finding the point in

$$\{(w, h) \in \mathbf{S} : wh = \widehat{W}\widehat{H}\}$$

which satisfies the desired L_1 constraint. We will properly define a measure of sparsity of H in the next section but here we comment that our measure ensures $\text{Sparsity}(H') = \text{Sparsity}(\widehat{H})$ and $\mathcal{E}_o(W', H') = \mathcal{E}_o(\widehat{W}, \widehat{H})$, which we will consider sufficient for our purposes of finding factors for D .

Definition II.1 (Primary objective). *Fix $N > 0$. Minimize (6) over all $W \in \mathbb{R}_+^{T \times N}$ and $H \in \mathbb{R}_+^{N \times L}$. Project solution to also satisfy (9).*

While the primary objective is essential, for the ease of interpreting the results, we would also like to achieve some secondary objectives. The algorithm will take care of the primary, and we can slightly modify the solution to meet the secondary objectives.

Definition II.2 (Secondary objectives). *Furthermore, we want*

- 1) *small values in H to be set to zero, without significantly affecting the approximation.*
- 2) *to get consistent results across different runs of the algorithm.*

Since our goal is an approximation of the form (1), the constraint (9) normalizes the signatures by L^1 norms, making dependence on the scale-factors appear in the H coefficients. We use the L^1 normalization to:

- Compare signatures with each other as if they were scale-independent. This allows immediate comparison without searching for scale factors which require H .
- Treat each signature as a probabilistic distribution in time

(since the entries all add up to 1). This essentially allows us to compare a signature with itself, e.g. if we have a signature with 0.1 at noon and 0.01 at 4pm, then there is 10 times as much traffic at noon.

- Compare W matrices across different runs of the algorithm (up to permutation of columns). Again, scale factors would further complicate this.
- It also allows one to think about doubling the matrix D and investigate whether the result will simply be a doubling of H , or a rearrangement of counts within and across signatures.

An alternate normalization would be based on the L^2 norms of the columns of W . The interpretation of what W entries mean would then be different.

We also note that, in general, an L_1 penalty only approximately enforces sparsity. Rigorously, sparsity would be enforced by an L_0 penalty; L_1 norms are used as the L_1 ball is the convex hull of an L_0 ball (in certain cases, L_1 penalty does give a “soft thresholding”; see [41]).

D. Measure of Sparsity

Note (6) penalizes the L^1 -norm of columns of H through the term $\beta \sum_{\ell=1}^L \|H_{:, \ell}\|_1^2$ as a proxy for the sparsity of H . However, this term is sensitive to the magnitude of H -entries. To measure sparsity that is not sensitive in this way, we instead use the function defined in [1] based on the relationship between L^1 and L^2 norms.

For a vector $x \neq 0$ of length N ,

$$\text{Sparsity}(x) = \frac{\sqrt{N} - \|x\|_1 / \|x\|_2}{\sqrt{N} - 1} \quad (10)$$

For a matrix $H_{N \times L}$ with non-negative entries and at least one positive entry per column, we let

$$\text{Sparsity}(H) = \frac{1}{L} \sum_{1 \leq \ell \leq L} \text{Sparsity}(H_{:, \ell}). \quad (11)$$

A completely sparse H i.e., with columns looking like $(1, 0, \dots, 0)^T, (0, 1, \dots, 0)^T, \dots$, or $(0, \dots, 0, 1)^T$ would have $\text{Sparsity}(H) = 1$. Our measure of sparsity enjoys the following properties:

- 1) $0 \leq \text{Sparsity}(A) \leq 1$
- 2) $\text{Sparsity}(A) = \text{Sparsity}(c \cdot A) \forall c \neq 0$ i.e., the measure of sparsity is scale-invariant.

This provides us with a normalized measure of sparsity that can be used to compare different H -matrices. We note that this measure of sparsity is unaffected by the rescaling of (8).

III. THE ALGORITHM

At this point, Definition II.1 represents a precise mathematical formulation of our search for traffic patterns. The cost function $\mathcal{E}_{\beta, \eta}$ in (6) is quadratic in both W and H , but not necessarily convex in the pair (W, H) [32]. Furthermore, we want the entries of W and H to be non-negative, and we are subsequently imposing constraints on W . This is a *constrained sparse non-negative matrix factorization problem* with missing values (see Subsection II-B).

Algorithm 1 Constrained Sparse Non-negative Matrix Factorization

```

1:  $\beta, \eta$ , rank, threshold ▷ Global variables
2: Initialize  $W^{(m+1)}, H^{(m+1)}$  with random positive entries
3: repeat
4:    $W^{(m)} \leftarrow W^{(m+1)}$ 
5:    $H^{(m)} \leftarrow H^{(m+1)}$ 
6:    $W^{(m+1)} \leftarrow \text{Update\_W}(D, W^{(m)}, H^{(m)})$  ▷ refer to (12)
7:    $H^{(m+1)} \leftarrow \text{Update\_H}(D, W^{(m+1)}, H^{(m)})$  ▷ refer to (12)
8:   Error =  $\|D - W^{(m+1)}H^{(m+1)}\|_F / \|D\|_F$  ▷ relative error
9: until  $\|W^{(m+1)} - W^{(m)}\|_F + \|H^{(m+1)} - H^{(m)}\|_F > \text{threshold}$  ▷ scale factor of each term omitted for brevity
10:  $W^{(m+1)}, H^{(m+1)} \leftarrow \text{Normalize\_W}(W^{(m+1)}, H^{(m+1)})$  ▷ impose  $L^1$  constraint on columns of  $W$ 
11: return  $W^{(m+1)}, H^{(m+1)}, \text{Error}$ 

```

Our algorithm is summarized as follows. For positive integers r (number of rows) and c (number of columns), let $\mathbf{1}_{r \times c}$ be the $(r \times c)$ -dimensional matrix whose entries are all 1. We want to minimize $\mathcal{E}_{\beta, \eta}$ as defined in (6). For any $(W, H) \in \mathbb{R}_+^{T \times N} \times \mathbb{R}_+^{N \times L}$ we use the following sequence of recursive update rules:

$$\begin{aligned}
 W &\leftarrow W \otimes ([D]_{\mathcal{N}} H^T) \odot ([WH]_{\mathcal{N}} H^T + \eta W) \\
 H &\leftarrow H \otimes (W^T [D]_{\mathcal{N}}) \odot (W^T [WH]_{\mathcal{N}} + \beta \mathbf{1}_{N \times N} H)
 \end{aligned} \tag{12}$$

These rules are based on [32] with sparsity modifications as laid out in [40]. We have reviewed these calculations in Appendix A. Using these update rules constitutes one iteration in our algorithm.

A. Criterion for Terminating the Algorithm

We will denote the initializations of W and H as $W^{(0)}$ and $H^{(0)}$, respectively. The m^{th} iteration of the update rules will then produce $W^{(m)}$ and $H^{(m)}$ until we have established convergence.

There are three natural indicators of convergence of these sequences:

- $\|W^{(m+1)} - W^{(m)}\|_F$,
- $\|H^{(m+1)} - H^{(m)}\|_F$, or
- $|\mathcal{E}_{\beta, \eta}(W^{(m+1)}, H^{(m+1)}) - \mathcal{E}_{\beta, \eta}(W^{(m)}, H^{(m)})|$.

We use a combination of the first two i.e., we stop our algorithm when

$$\frac{\|W^{(m+1)} - W^{(m)}\|_F}{\|W^{(m)}\|_F} + \frac{\|H^{(m+1)} - H^{(m)}\|_F}{\|H^{(m)}\|_F} \leq \text{threshold}.$$

B. Initial Conditions

In order for the algorithm to function correctly, we need to ensure that the initializations for W and H are such that:

- all their entries are positive and
- the columns of W and the rows of H are linearly independent.

The first condition above stems from the fact that our algorithm is multiplicative and once an entry is zero, it will continue to be zero in the successive iterations.

A safe way to cater to both these conditions is to simply initialize their entries by sampling uniform random positive values. This works because random matrices are usually full rank. More accurately, because this selection of initial conditions gives measure zero to the collection of singular matrices. If by chance this random initialization is not full rank, we try re-picking W and H matrices.

C. Utilizing Sparsity of H

Once Algorithm 1 runs, we proceed to address Definition II.2 (Secondary Objectives). Recall that we have borrowed our measure of sparsity from [1]. We can utilize sparsity in H to express our approximation of $D_{:, \ell}$ as a linear combination of as few signatures as possible. To implement this, we utilize a threshold operator on H . For each column of H , we set the lowest entries which sum to an `axe_threshold` to zero, where $0 \leq \text{axe_threshold} \leq 1$. For example, if `axe_threshold` = 0.4, we set the all entries below the 40th percentile to zero in each column of H . This “compressed” H will result in a higher error percentage, but also a higher sparsity value and is better for compression since now one can keep track of only the non-zero entries in H (i.e. we can store H as a sparse matrix if needed).

Mathematically, if we consider the signatures to be the basis elements of a (rank=) 50-dimensional vector space, then compressing H has the effect of projecting each coefficient vector (a column of H) onto the closest subspace of the lowest dimension possible.

We can predict that compressing H will cause an increase in error and an increase in sparsity. However, these changes are not significant as we will show in Table 4 once we run the algorithm on an actual data set.

IV. ALGORITHM: NYC CASE STUDY

We start with the analysis of [37] (see also [38]). The data set contains hourly traffic data for 2011; there are thus

$$1 \text{ yr} \times \frac{365 \text{ days}}{\text{yr}} \times \frac{24 \text{ hours}}{\text{day}} = T \text{ hours}$$

where $T \stackrel{\text{def}}{=} 8760$.

There are originally $L_o \stackrel{\text{def}}{=} 260855$ one-directional links (roadways) in the dataset. A one-way road segment corresponds exactly to a single link, while a two-directional road segment corresponds to two links. The dataset corresponds to a matrix $D^\circ \in \mathbb{N}_0^{T \times L_o}$ of numbers of taxi trips indexed by time and link (the number of taxi trips being integer-valued, the elements of D° are in $\mathbb{N}_0 \stackrel{\text{def}}{=} \{0, 1, \dots\}$); $D_{t,\ell}^\circ$ is the number (if known) of taxi trips along link ℓ in hour t .

We are assuming that taxi trips are reasonable representations of overall traffic flow; we are thus assuming that D represents overall traffic density in Manhattan. We note that D has

$$8,760 \times 260,855 \approx 2.3 \times 10^9$$

entries. To be specific,

$$D_{34,128255}^\circ = 3$$

corresponds to 3 taxi trips on link 128255 during hour 34 of 2011.

Hour 34 corresponds to the 34th hour of 2011; i.e., between 09 and 10 AM on January 2, 2011. Link 128255 corresponds to the stretch of West 44th Street, from¹ (40.759728, -73.991684) to (40.758532, -73.988843); i.e., from 8th Ave to 7th Ave.

On the other hand,

$$D_{25,128255}^\circ = 0,$$

meaning that there were no taxi trips on link 128255 in hour 25 (i.e., between midnight and 01 AM on January 2, 2011). Precisely, this means that the algorithm of [37] assigned no taxis to this link at this time. This may have occurred for several reasons:

- there was practically no traffic *at all* on this link at this time, or
- the link was so congested with traffic that taxis avoided it altogether.

We will remain agnostic about these two possibilities. However, zero taxis does not mean zero traffic. We use our algorithm to estimate the traffic densities on these links based on large-scale behaviors. To do this we treat the zero values as missing entries in our data.

A. Missing Data and Reduction of Scope

In fact, there are 2181923208 zero (i.e., missing) entries; i.e., about 95% of all entries of D are missing. To proceed, let \mathcal{L} consist of those links for which there are at most 720

hours (30 days worth) of missing data; i.e.,

$$\mathcal{L} \stackrel{\text{def}}{=} \left\{ \ell \in \{1, 2, \dots, L_o\} : \sum_{t=1}^T \mathbf{1}_{\{D_{t,\ell}^\circ = \text{NaN}\}} \leq 720 \right\}.$$

where $\mathbf{1}$ denotes the indicator function. This 30 day cutoff is arbitrary, but it allows us to feed a smaller matrix to the algorithm. Increasing this cutoff will mean more missing values in D . As the proportion of missing values in D increases, the efficiency of the algorithm drops (due to the masking positions \mathcal{N} in (5)).

Let $L \stackrel{\text{def}}{=} |\mathcal{L}| = 2302$; there are 2302 links with at most 720 hours of missing data.

For example, link 151845 is 5th Avenue from 17th Street to 18th Street.² It is missing only 40 entries, so $151845 \in \mathcal{L}$. On the other hand, link 128255 has 1965 missing entries and thus $128255 \notin \mathcal{L}$.

We now let $D \in \mathbb{N}_0^{T \times L}$ be the submatrix of D_o corresponding to links in \mathcal{L} . We use this D as the input for our matrix factorization algorithm.

B. Constants and Initializations

For the taxi traffic data of year 2011, on 2302 of the busiest links, we use the following values, and we will subsequently justify these choices of values.

- rank = 50
- $\beta = 5000$
- $\eta = \max_{i,j}(D_{ij})$
- threshold = 0.005
- axe_threshold = 0.4

In order to decide on these values, we first ran the algorithm on a range of (rank, β) values. For all these runs, we use $\eta = \max_{i,j}(D_{ij})$, where η is a regularization factor which ensures that entries in W do not grow too large. Having it at the same scale as entries in D ensures that all of the terms in $\mathcal{E}_{\beta,\eta}$ are of the same order of magnitude.

Also, we generally consider a sparsity value between 0.8 and 1.0 to be sufficient as it meets our Secondary Objective for the NYC dataset. The results are summarized in the graphs in Figures 11 and 12. The choice for rank and β are independent of the scale of entries in D and robustly capture how well D can be approximated by lower-rank approximations.

We note that for high rank, high β combinations, the algorithm forces a column of zero entries in W . Once this happens, there values are stuck at zero since the update rules are multiplicative. Our code was written so as to capture these occurrences and flag them as being a sub-optimal usage of rank. Hence, these are not shown in Figures 1 and 2.

This suggests that there is a tradeoff: higher rank and higher beta give us better (error, sparsity) results, but are also more likely to break the algorithm for certain initializations of W and H .

These considerations leave us with two contending (rank, β) values:

- rank = 50, $\beta = 5000$, and

²start and end coordinates (40.737917, -73.992225) and (40.738504, -73.991798) respectively.

¹Earth coordinates will be given in (Latitude, Longitude)

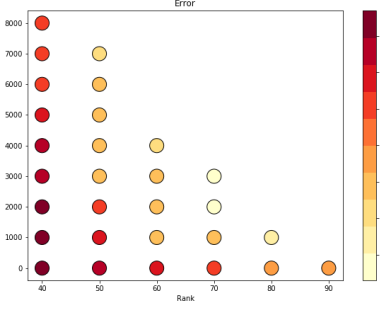


Fig. 1. Relative error percentages for various (rank, β) pairs

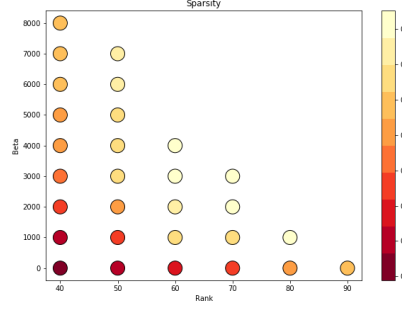


Fig. 2. Sparsity of H for various (rank, β) pairs

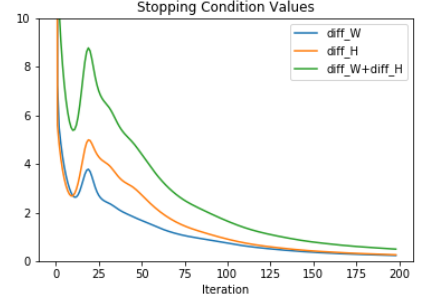


Fig. 3. Progression of Stopping Function over One Run of the Algorithm

	Rank=50, $\beta = 5000$	Rank=70, $\beta = 2000$
Error before compression	25.8	23.7
Error after compression	39.3	38.2
Sparsity before compression	0.789	0.814
Sparsity after compression	0.900	0.913
Mean dispersion	0.624	0.658
Std. dev. of dispersion	0.174	0.146

Fig. 4. Comparison of two (rank, β) choices

4

- rank= 70, $\beta = 2000$.

To decide between these two choices, we used a measure of dispersion for signatures. We first define, for each signature, a mean-weekly-trend. This is obtained by averaging traffic for that signature over all Mondays to obtain typical Monday behaviour, all Tuesdays to obtain typical Tuesday behaviour, and so on. Then the dispersion of each signature is calculated as the mean absolute deviation (MAD) of that signature from its mean-weekly-trend. The mean and standard deviation of this dispersion across all signatures are listed and compared in Table 4.

Dispersion in signatures (as defined above) can be seen as a tendency to deviate from a 7-day periodic behaviour. A higher mean dispersion indicates that signatures are erratic and are picking up all of the noise in the D matrix. A higher standard deviation for dispersion indicates that all the dispersion is localized into just a few of the signatures. Thus, we prefer lower mean and higher deviation values for this measure. Given the closeness of all the other values, we settle on rank= 50 and $\beta = 5000$ as our parameters. For further context, we can use the entrywise absolute deviation in order to put our relative error into taxi units. For rank=50 and $\beta = 5000$, the median absolute deviation is 37 taxis with the first and third quartiles of 12 and 95 taxis, respectively. So, you can see the values are skewed higher, and most of the deviations are closer to 0. Hence, for a given hour and link, one can think of our approximation as being off by about 37 taxis.

With these choices fixed, we then initialize W and H to have entries which are random values sampled uniformly from the interval $(0, 1)$. This bounded positive interval is chosen simply for ease of programming. The algorithm typically terminates after about 200 ± 20 iterations. The total runtime

is under 6 minutes on a 2.20GHz processor.

Figure 13 shows a plot of

$$\frac{\|W^{(m+1)} - W^{(m)}\|_F}{\|W^{(m)}\|_F} + \frac{\|H^{(m+1)} - H^{(m)}\|_F}{\|H_n\|_F}$$

plotted across iterations. The graph tells us that most of the convergence takes place in the first 100 iterations, and it also serves as a justification for our choice of threshold= 0.005.

C. Interpretation of low rank decomposition for traffic

The algorithm outputs two matrices W and H of sizes 8760×50 and 50×2302 . The columns of W are L^1 -normalized and the columns of H are sparse. The normalization of the columns of W could even allow us to compare traffic patterns across different cities and different years.

Recall that columns of W represent traffic signatures over time. Each signature is a time-series for the entire year and hence need not be periodic. For example, the signatures can capture traffic anomalies during holidays, hurricanes, and blackouts.

Further, the columns of H represent coefficients, which represent the way in which the links are decomposed into distinct signatures. For example, if the 4th column of H is $(0, 7, 2, 0, \dots, 0)^T$, then the traffic in link 4 of \mathcal{L} can be written as 7 times the second signature plus 2 times the third signature. This decomposition allows us to identify spatial patterns in traffic across the city.

The plots of Figures 11 and 12 may be more widely useful in comparing the complexity of traffic patterns in different cities or under different circumstances. Namely, if the traffic in a city can be well-explained by a few signatures, the error of the approximation should level off at a low value for large values of N and sparsity penalty β .

D. Tradeoff between Sparsity and Independence of Signatures

In order to ensure that the signatures obtained from the algorithm are linearly independent, we calculate the condition number of W . By performing several runs of our algorithm for rank 50 with different (W, H) -initializations, we determine

$$\text{Condition Number for } W = 24 \pm 2$$

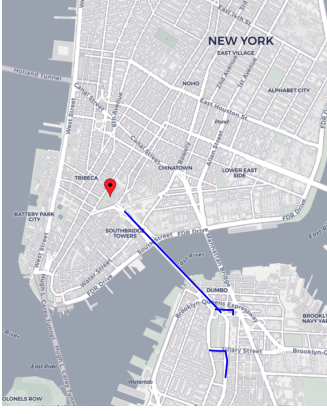


Fig. 5. A map of signature 21 with location of Wisconsin Labor Rally

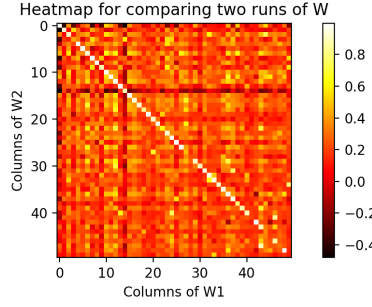


Fig. 6. Heatmap of correlation coefficients of columns of W from two runs of algorithm

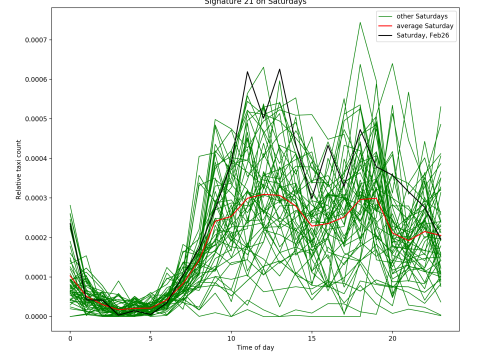


Fig. 7. Signature 21 during the Wisconsin Labor Rally

This number is low enough to give us confidence in the W returned by the algorithm. In practice, a high condition number (in the thousands) can indicate that the rank needs to be reduced.

E. Robustness of Algorithm

The factorization produced by Algorithm 1 is not unique. However, in all observed cases, two successive instances of W differ mostly by a permutation of the signatures. Figure 26 shows the heatmap of the correlation coefficients of W calculated from two runs of the algorithm after applying a suitable permutation. We observe high correlation coefficients on the diagonals.

V. OBSERVATIONS ENABLED BY DATA REDUCTION

The goal of this mathematical analysis is to enable discovery of broad patterns and identifying anomalous behavior of traffic. First, we note that the signatures (and hence columns of D) are roughly periodic, with a periodicity of 7 days. This was confirmed by computing the autocorrelation of each signature as well as columns of D . Deviation from this periodic pattern could arise due to many events, e.g., holidays, elections, extreme weather, or from noise present in D .

Due to the periodicity, we can look at one day of the week across the entire year, e.g. all Mondays and compute the average traffic for that day. Next, we determine which dates have relative taxi counts that differ significantly from the average. In the subsections below, we list some of these anomalous traffic patterns. It should be noted that we are presenting here just a selection from the vast number of observations possible using the W -matrix we have obtained.

A. Hurricane Irene

Figure 38 shows Signature 0 capturing a near-shutdown of taxi traffic on August 27, 2011. This may have been caused by Hurricane Irene hitting NYC. There was an early warning and all subways and buses were shut down at noon on Saturday, August 27. A zoned taxi system was implemented at 9am and taxis were thereafter running flat fares instead of meters [42].

All other signatures also show similar behavior on and around August 28.

B. Wisconsin Labor Rally

Figure 27 shows the behaviour of Signature 21 on February 26, 2011. The traffic deviates from the average Saturday trend. This may have been caused a Labor Rally that took place near the New York City Town Hall on this day in support of Wisconsin public employees [43]. Figure 25 shows that Signature 21 is used by links near the Town Hall, which can be seen as further evidence connecting the rally to this traffic deviation.

C. Christmas Day

Anomalous behavior was also observed on Christmas Day. This can be seen in Figure 39 for Signature 0 and Figure 310 for Signature 4.

D. Endemic Signatures

We note that of the 50 signatures, some tend to be geographically restricted (called *endemic*), while others are spread out over larger areas (called *dispersive*).

The endemic signatures might sometimes explain traffic densities only on a single but long stretch of road. For example, Figure 12 shows that Signature 10 is largely used by the the north-bound 3rd Avenue and streets like Bowery, Lafayette St. and the southernmost part of Broadway that feed into 3rd Avenue. Similarly, Figure 14 shows Signature 40 being used exclusively by a small section of the south-bound Broadway traffic near Central Park.

In some other cases, signatures can be seen as having a lateral sphere of influence in that they affect not only one street but also other feeding into or out of the street transversally — for example, Signature 24 as seen in Figure 13.

VI. FUTURE REMARKS

In this paper, we proposed a scale-free framework for studying the trade-off between the error of approximation, the

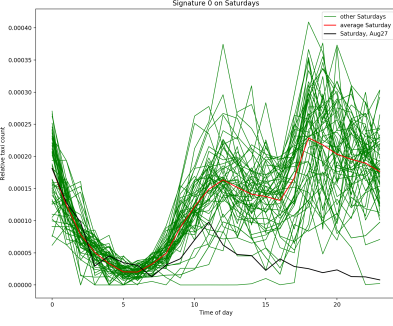


Fig. 8. Signature 0 during Hurricane Irene

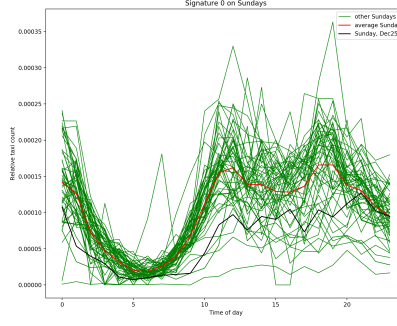


Fig. 9. Signature 0 during Christmas day

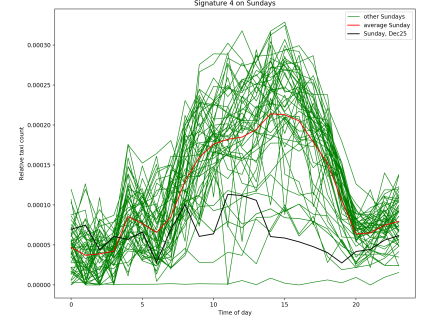


Fig. 10. Signature 4 during Christmas day



Fig. 11. Signature 0 is a dispersive signature, meaning it is not restricted to a particular street or neighbourhood.

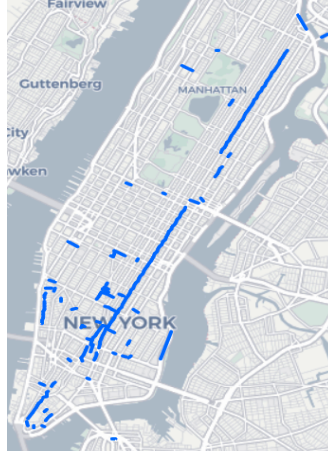
Fig. 12. Signature 10 representing 3rd Avenue and tributary streets.Fig. 13. Signature 24 representing 7th Avenue and distributary streets.

Fig. 14. [Signature 40 representing a section of Broadway near Central Park.

Figure. Some examples of endemic signatures, and one example of a dispersive signature.

sparsity (i.e. the number of signatures), and the dimension of the proposed low-rank factorization. One can further expand this analysis to the less dense traffic in New York City 2011. This structure can provide direction for future research on comparing the complexity of traffic patterns in different cities or under different circumstances. Various city attributes can be connected to different traffic conditions and signatures, which opens the possibility of future analysis on problems such as urban planning and decision making. Another avenue for further research is to combine our quantitative tool with the element of probability. In other words, one would use this format to predict the future state of traffic signatures on a given link (road).

VII. CONCLUSION

This article contributes to the study of large-scale urban traffic matrices via low-rank factorization. We execute a non-negative, *sparse* matrix factorization algorithm to study the complexity of city traffic, and explain its underlying behavior with a small number of signatures. A multiplicative numerical procedure is examined in order to solve the factorization problem and a quantitative measure of sparsity is presented. The algorithm also enforces a normalization constraint on the

solution which serves as a key step in making the results interpretable as temporal patterns. This factorization algorithm is then applied to a 2011 New York City traffic data set, and anomalous traffic patterns are interpreted from the results. We were able to explain the complexity of dense New York traffic in 2011 with one to eight signatures per link (road segment). In this process, we were able to identify several city-wide attributes and lay the ground work for future analyses.

REFERENCES

- [1] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *Journal of machine learning research*, vol. 5, no. Nov, pp. 1457–1469, 2004.
- [2] Y. Han and F. Moutarde, "Analysis of large-scale traffic dynamics in an urban transportation network using non-negative tensor factorization," *International Journal of Intelligent Transportation Systems Research*, vol. 14, no. 1, pp. 36–49, 2016.
- [3] X. J. Ban, P. Hao, and Z. Sun, "Real time queue length estimation for signalized intersections using travel times from mobile sensors," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 6, pp. 1133–1156, 2011.
- [4] J. Zheng and H. X. Liu, "Estimating traffic volumes for signalized intersections using connected vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 347–362, 2017.
- [5] R. Herman and I. Prigogine, "A two-fluid approach to town traffic," *Science*, vol. 204, no. 4389, pp. 148–151, 1979.

- [6] H. S. Mahmassani, J. C. Williams, and R. Herman, "Investigation of network-level traffic flow relationships: some simulation results," *Transportation Research Record: Journal of the Transportation Research Board*, no. 971, pp. 121–130, 1984.
- [7] N. Geroliminis and C. F. Daganzo, "Existence of urban-scale macroscopic fundamental diagrams: Some experimental findings," *Transportation Research Part B: Methodological*, vol. 42, no. 9, pp. 759–770, 2008.
- [8] W. Krichene, M. S. Castillo, and A. Bayen, "On social optimal routing under selfish learning," *IEEE Transactions on Control of Network Systems*, 2016.
- [9] J. A. Deri and J. M. F. Moura, "Taxi data in new york city: A network perspective," in *2015 49th Asilomar Conference on Signals, Systems and Computers*, pp. 1829–1833, Nov 2015.
- [10] Y. Zhu, K. Ozbay, K. Xie, and H. Yang, "Using big data to study resilience of taxi and subway trips for hurricanes sandy and irene," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2599, pp. 70–80, 2016.
- [11] X. Zhan, S. V. Ukkusuri, and F. Zhu, "Inferring urban land use using large-scale social media check-in data," *Networks and Spatial Economics*, vol. 14, no. 3, pp. 647–667, 2014. <http://dx.doi.org/10.1007/s11067-014-9264-4>.
- [12] X. Guan, C. Chen, and D. Work, "Tracking the evolution of infrastructure systems and mass responses using publicly available data," *PloS one*, vol. 11, no. 12, p. e0167267, 2016.
- [13] J. Alonso-Mora, S. Samaranayake, A. Wallar, E. Frazzoli, and D. Rus, "On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment," *Proceedings of the National Academy of Sciences*, p. 201611675, 2017.
- [14] N. Ferreira, J. Poco, H. T. Vo, J. Freire, and C. T. Silva, "Visual exploration of big spatio-temporal urban data: A study of new york city taxi trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013.
- [15] N. Gillis, "Introduction to Nonnegative Matrix Factorization," *ArXiv e-prints*, Mar. 2017.
- [16] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, pp. 865–873, April 2015.
- [17] M. Chagoyen, P. Carmona-Saez, H. Shatkay, J. M. Carazo, and A. Pascual-Montano, "Discovering semantic features in the literature: a foundation for building functional associations," *BMC bioinformatics*, vol. 7, no. 1, p. 41, 2006.
- [18] V. P. Pauca, F. Shahnaz, M. W. Berry, and R. J. Plemmons, "Text mining using non-negative matrix factorizations," in *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp. 452–456, SIAM, 2004.
- [19] J.-P. Brunet, P. Tamayo, T. R. Golub, and J. P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *Proceedings of the national academy of sciences*, vol. 101, no. 12, pp. 4164–4169, 2004.
- [20] P. Carmona-Saez, R. D. Pascual-Marqui, F. Tirado, J. M. Carazo, and A. Pascual-Montano, "Biclustering of gene expression data by non-smooth non-negative matrix factorization," *BMC bioinformatics*, vol. 7, no. 1, p. 78, 2006.
- [21] Y. Gao and G. Church, "Improving molecular cancer class discovery through sparse non-negative matrix factorization," *Bioinformatics*, vol. 21, no. 21, pp. 3970–3975, 2005.
- [22] P. M. Kim and B. Tidor, "Subsystem identification through dimensionality reduction of large-scale gene expression data," *Genome research*, vol. 13, no. 7, pp. 1706–1718, 2003.
- [23] E. A. Maher, C. Brennan, P. Y. Wen, L. Durso, K. L. Ligon, A. Richardson, D. Khatri, B. Feng, R. Sinha, D. N. Louis, *et al.*, "Marked genomic differences characterize primary and secondary glioblastoma subtypes and identify two distinct molecular and clinical secondary glioblastoma entities," *Cancer research*, vol. 66, no. 23, pp. 11502–11513, 2006.
- [24] D. R. Carrasco, G. Tonon, Y. Huang, Y. Zhang, R. Sinha, B. Feng, J. P. Stewart, F. Zhan, D. Khatri, M. Protopopova, *et al.*, "High-resolution genomic profiles define distinct clinico-pathogenetic subgroups of multiple myeloma patients," *Cancer cell*, vol. 9, no. 4, pp. 313–325, 2006.
- [25] P. Pehkonen, G. Wong, and P. Törönen, "Theme discovery from gene lists for identification and viewing of multiple functional groups," *BMC bioinformatics*, vol. 6, no. 1, p. 162, 2005.
- [26] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng, "Learning spatially localized, parts-based representation," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I–I, IEEE, 2001.
- [27] H. Tan, Y. Wu, G. Feng, W. Wang, and B. Ran, "A new traffic prediction method based on dynamic tensor completion," *Procedia-Social and Behavioral Sciences*, vol. 96, pp. 2431–2442, 2013.
- [28] D. Dueck, Q. D. Morris, and B. J. Frey, "Multi-way clustering of microarray data using probabilistic sparse matrix factorization," *Bioinformatics*, vol. 21, no. suppl_1, pp. i144–i151, 2005.
- [29] A. Pascual-Montano, J. M. Carazo, K. Kochi, D. Lehmann, and R. D. Pascual-Marqui, "Nonsmooth nonnegative matrix factorization (nsnmf)," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 3, pp. 403–415, 2006.
- [30] V. P. Pauca, J. Piper, and R. J. Plemmons, "Nonnegative matrix factorization for spectral data analysis," *Linear algebra and its applications*, vol. 416, no. 1, pp. 29–47, 2006.
- [31] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
- [32] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in neural information processing systems*, pp. 556–562, 2001.
- [33] Z. Li, Y. Zhu, H. Zhu, and M. Li, "Compressive sensing approach to urban traffic sensing," in *2011 31st International Conference on Distributed Computing Systems*, pp. 889–898, June 2011.
- [34] E. O. A. Hyvriinen, J. Karhunen, *Independent Component Analysis*. Wiley, 2001.
- [35] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *eprint arXiv:cs/0408058*, Aug. 2004.
- [36] V. Karve, D. Yager, M. Abolhelm, D. Work, and R. Sowers, "NYC Traffic Patterns cSNMF Source Code," <https://gitlab.engr.illinois.edu/TrafficPatterns/CSNMF.git>.
- [37] B. Donovan and D. B. Work, "Using coarse gps data to quantify city-scale transportation system resilience to extreme events," *arXiv preprint arXiv:1507.06011*, 2015.
- [38] M. T. Asif, N. Mitrovic, J. Dauwels, and P. Jaillet, "Matrix and tensor based methods for missing data estimation in large traffic networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1816–1825, 2016.
- [39] S. A. Vavasis, "On the complexity of nonnegative matrix factorization," *SIAM Journal on Optimization*, vol. 20, no. 3, pp. 1364–1377, 2009.
- [40] H. Kim and H. Park, "Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method," *SIAM journal on matrix analysis and applications*, vol. 30, no. 2, pp. 713–730, 2008.
- [41] H. Lee, M. K. Chung, H. Kang, B.-N. Kim, and D. S. Lee, "Discriminative persistent homology of brain networks," in *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, pp. 841–844, IEEE, 2011.
- [42] "NY subway system shuts down due to Hurricane Irene (updated)," <http://www.wnyc.org>.
- [43] "Wisconsin worries: Labor rallies in NY," <https://nypost.com/2011/02/26/wisconsin-worries-labor-rallies-in-ny/>.

APPENDIX

Let's write down the calculations leading to the Algorithm I of Section III.

Writing $\mathcal{E}_{\beta,\eta}$ of (6) out gives

$$\mathcal{E}_{\beta,\eta}(W, H) = \sum_{(t,\ell) \in \mathcal{N}} |D_{t,\ell} - (WH)_{t,\ell}|^2 + \beta \sum_{n=1}^N \left(\sum_{\ell=1}^L H_{n,\ell} \right)^2 + \eta \sum_{n=1}^N \sum_{t=1}^T W_{\ell,t}^2.$$

We should be able to solve this by alternating between minimization problems in W and H . Namely, if we start with a fixed $(W, H) \in \mathbb{R}_+^{T \times N} \times \mathbb{R}_+^{N \times L}$, we can construct a descent step for the function $\mathcal{E}_{\beta,\eta}(W, \cdot)$ and then, letting H' be the result, we can construct a descent step for $\mathcal{E}_{\beta,\eta}(\cdot, H')$. This hopefully gives us an improvement in the pair (W, H) , and we can then proceed iteratively.

The gradients of $\mathcal{E}_{\beta,\eta}$ in the directions of W and H are

given by

$$\begin{aligned} \frac{\partial \mathcal{E}_{\beta,\eta}}{\partial W_{\hat{t},\hat{n}}}(W, H) \\ &= -2 \sum_{\ell: (\hat{t}, \ell) \in \mathcal{N}} \left(D_{\hat{t},\ell} - \sum_{n=1}^N W_{\hat{t},n} H_{n,\ell} \right) H_{\hat{n},\ell} + \eta W_{\hat{t},\hat{n}} \\ &= -2 ([D - WH]_{\mathcal{N}} H^T + \eta W)_{\hat{t},\hat{n}} \end{aligned}$$

and

$$\begin{aligned} \frac{\partial \mathcal{E}_{\beta,\eta}}{\partial H_{\hat{n},\hat{\ell}}}(W, H) \\ &= -2 \sum_{t: (t, \hat{\ell}) \in \mathcal{N}} \left(D_{t,\hat{\ell}} - \sum_{n=1}^N W_{t,n} H_{n,\hat{\ell}} \right) W_{t,\hat{n}} \\ &\quad + 2\beta \left(\sum_{n=1}^N H_{n,\hat{\ell}} \right) \\ &= -2 (W^T [D - WH]_{\mathcal{N}})_{\hat{n},\hat{\ell}} + 2\beta (\mathbf{1}_{N \times N} H)_{\hat{n},\hat{\ell}}. \end{aligned}$$

Let's review the ideas of [40]. We want to iteratively find the critical points of $\mathcal{E}_{\beta,\eta}$; i.e., the solutions of

$$\begin{aligned} [WH]_{\mathcal{N}} H^T - [D]_{\mathcal{N}} H^T + \eta W &= 0 \\ W^T [WH]_{\mathcal{N}} - W^T [D]_{\mathcal{N}} + \beta \mathbf{1}_{N \times N} H &= 0 \end{aligned}$$

Let's now construct a multiplicative descent rule (which need not be *gradient descent*; see [32]). Fix $(W, H) \in \mathbb{R}_+^{T \times N} \times \mathbb{R}_+^{N \times L}$. Assume that

$$\frac{\partial \mathcal{E}_{\beta,\eta}}{\partial H_{n,\ell}} > 0; \quad (13)$$

we can then decrease the value of $\mathcal{E}_{\beta,\eta}$ by decreasing $H_{n,\ell}$. Let's rewrite (13) as

$$-2 (W^T [D - WH]_{\mathcal{N}})_{n,\ell} + 2\beta (\mathbf{1}_{N \times N} H)_{n,\ell} > 0$$

or rather

$$(W^T [WH]_{\mathcal{N}})_{n,\ell} + \beta (\mathbf{1}_{N \times N} H)_{n,\ell} > (W^T [D]_{\mathcal{N}})_{n,\ell};$$

since W , H , and D all have nonnegative entries, both sides of this equation are nonnegative, so this in turn is $\chi_{n,\ell}^h(W, H) < 1$ where

$$\chi_{n,\ell}^h(W, H) \stackrel{\text{def}}{=} \frac{(W^T [D]_{\mathcal{N}})_{n,\ell}}{(W^T [WH]_{\mathcal{N}})_{n,\ell} + \beta (\mathbf{1}_{N \times N} H)_{n,\ell}}.$$

Thus, another way to decrease $H_{n,\ell}$ while still retaining nonnegativity is to multiply it by $\chi_{n,\ell}^h(W, H)$. Reviewing these steps, we also see that if $\frac{\partial \mathcal{E}_{\beta,\eta}}{\partial H_{n,\ell}} < 0$, we want to increase $H_{n,\ell}$, and can again multiply by $\chi_{n,\ell}^h(W, H)$. Finally, if $\frac{\partial \mathcal{E}_{\beta,\eta}}{\partial H_{n,\ell}} = 0$ (i.e., we have found a critical point) $\chi_{n,\ell}^h(W, H) = 1$, so multiplying $H_{n,\ell}$ by $\chi_{n,\ell}^h(W, H)$ leaves $H_{n,\ell}$ unchanged.

The update rule for $W_{t,n}$ is similar. To start, let's assume that

$$\frac{\partial \mathcal{E}_{\beta,\eta}}{\partial W_{t,n}} > 0; \quad (14)$$

then we can decrease $\mathcal{E}_{\beta,\eta}$ by decreasing $W_{t,n}$. We can rewrite

(14) as

$$-2 ([D - WH]_{\mathcal{N}} H^T + \eta W)_{t,n} > 0.$$

We can again rewrite this as the comparison of two nonnegative quantities;

$$([WH]_{\mathcal{N}} H^T + [ToDo : \eta] W)_{t,n} > ([D]_{\mathcal{N}} H^T)_{t,n};$$

This in turn is equivalent to $\chi_{t,n}^w(W, H) < 1$ where

$$\chi_{t,n}^w(W, H) \stackrel{\text{def}}{=} \frac{([D]_{\mathcal{N}} H^T)_{t,n}}{([WH]_{\mathcal{N}} H^T + [ToDo : \eta] W)_{t,n}}$$

In other words, we can decrease $W_{t,n}$ by multiplying by $\chi_{t,n}^w(W, H)$. One can similarly see that if $\frac{\partial \mathcal{E}_{\beta,\eta}}{\partial W_{t,n}} < 0$, gradient descent again increases or decreases W with the same sign as multiplying by $\chi_{t,n}^w(W, H)$.

Our proposed update rule for W and H is now

$$\begin{aligned} W'_{t,n} &= W_{t,n} \chi_{t,n}^w(W, H) \\ H'_{n,\ell} &= H_{n,\ell} \chi_{n,\ell}^h(W, H). \end{aligned}$$

which is equivalent to (III). These equations follow from [40].



Vaibhav Karve is a PhD student in the Department of Mathematics at University of Illinois at Urbana-Champaign. Vaibhav received a Bachelor of Science and a Master of Science in Mathematics (2015) from the Indian Institute of Science Education and Research – Kolkata. His research interests are in traffic estimation, computational algebra and computational topology.



Derrek Yager is a Ph.D. student in the Department of Mathematics at University of Illinois at Urbana-Champaign. Derrek received a Bachelor of Arts degree (2010) in mathematics and Spanish from Wabash College and a Master of Arts degree (2012) in mathematics from Miami University (OH). His research interests are in traffic estimation, big data, and Combinatorics.



Marzieh Abolhelm is a PhD student in the Department of Industrial and Enterprise Systems Engineering at University of Illinois at Urbana-Champaign. Marzieh received a Bachelor of Science degree (2012) in Financial Management from University of Tehran and a Master of Science degree (2013) in Finance from University of Illinois at Urbana-Champaign. Her research interests are in financial risk analysis, data analytic, and traffic estimation.



Daniel B. Work is an associate professor in the Department of Civil and Environmental Engineering, Electrical Engineering and Computer Science, and the Institute for Software Integrated Systems at Vanderbilt University. Prof. Work earned his bachelor of science degree (2006) from the Ohio State University, and a master of science (2007) and Ph.D. (2010) from the University of California, Berkeley, each in civil engineering. Prof. Work has research interests in transportation cyber physical systems and data analytics. Prof. Work is a recipient of the

US National Academy of Engineering Armstrong Endowment for Young Engineers Gilbreth Lectureship in 2018 and a CAREER award from the National Science Foundation in 2014.



Richard Sowers is a professor of Industrial and Enterprise Systems Engineering and Mathematics at the University of Illinois at Urbana-Champaign. Prof. Sowers earned a bachelor of science degree (1986) from Drexel University, a master of science degree (1988) from University of Maryland at College Park, each in electrical engineering. Prof. Sower received a Ph.D. (1991) in Applied Mathematics from University of Maryland at College Park.