ERDC/GRL TN-20-3
July 2020

Spatiotemporally Coherent Tensor Decompositions for the Analysis of Trajectory Data

By Trevor Ruiz and Charlotte Ellison

PURPOSE: Location acquisition technologies such as global positioning systems (GPS) sensors or telemetry devices generate abundant spatiotemporal measurements of movement of people, animals, and vehicles. The resultant data represent trajectories – paths in space and time traversed by moving objects – and can often be merged with additional information about the entities in motion from connected or external data sources (Zheng 2015). New data analysis frameworks may be able to uncover patterns of human behavior from the fused trajectory and contextual information. This data and new insights gained from novel analysis tools are potentially of great interest to the Army and the geospatial community.

Methods for the analysis of collections of trajectories vary by objective. For instance, stochastic process models can be used for statistical inference of dependence among moving entities and uncertainty quantification of location from noisy or missing measurements (Scharf et al. 2016; Scharf et al. 2018); distance-based clustering methods using descriptive measures of spatiotemporal similarity provide a flexible suite of tools for searching for various inter-trajectory patterns (Zheng and Zhou 2011). Yet, these frameworks for the analysis of trajectory data, despite their many merits, provide no clear avenue for incorporating covariates or external data in order to understand relationships between movement patterns and other information.

Methods based on tensor representations (multidimensional arrays) are promising in their capacity to merge additional information with trajectory data (Zheng 2015). A wide array of potentially useful techniques are available as tensor decomposition methods can incorporate constraints (Allen 2012a; Allen 2012b; Park et al. 2019; Shashua and Hazan 2005; Sun et al. 2017; Zhang and Han 2019), leverage auxiliary information (Acar et al. 2011; Narita et al. 2012; Yilmaz et al. 2011; Zhou et al. 2012), and utilize a variety of loss functions (Hong et al. 2018; Yilmaz et al. 2011). In addition, there are quantitative methods for selection of decomposition complexity (Ceulemans and Kiers 2006; Shi et al. 2019). These methods are most often applied to tasks such as signal reconstruction for corrupted audio or image data (Yilmaz et al. 2011), recommendation systems from sparse inputs (Acar et al. 2011; Zheng et al. 2014), multiway analysis of high-dimensional contingency tables (Acar and Yener 2008), and multiview generalizations of multivariate dimension reduction techniques such as canonical correlation analysis or principal components analysis (De Lathauwer et al. 2000b; Luo et al. 2015). While there are a number of examples of analyses of trajectory data utilizing tensor decomposition methods in some capacity (Deng and Ji 2011; Sun and Axhausen 2016; Wang et al. 2014), these approaches are, at present, highly task-specific. Furthermore, existing tensor decompositions tend to implicitly assume a degree of uniformity of input data that is rarely present in measurements exhibiting spatiotemporal

continuity and complex dependence. This is a key challenge in effectively developing decomposition techniques for generalizable application to information-rich trajectory datasets.

The present work provides a tensor representation framework for trajectories and develops a constrained decomposition designed to reproduce, with better fidelity, the spatial sparsity and temporal continuity characteristic of trajectory data represented in this tensor form. This work is regarded as a starting point in developing exploratory analysis techniques based on tensor decompositions that generalize to datasets comprising trajectories merged with non-spatiotemporal information. A preliminary overview of notations, key tensor operations, and the core decomposition method is given before a detailed description of the constrained decomposition method. The method's utility in pattern detection is demonstrated using taxi trajectory data, wherein some novel techniques for exploratory analysis using decomposition outputs are given. Lastly, a concluding discussion of the promise and limitations of the present work is offered.

BACKGROUND: This section provides a selective overview of key background material, based largely on three sources (Allen 2012a; De Lathauwer et al. 2000a; Kolda and Bader 2009); some of the notations used here deviate from these sources. The subsections cover tensor notations and operations, the CANDECOMP/PARAFAC (CP) decomposition, and a method of computing the CP decomposition.

Notations. Tensors (understood here as multidimensional arrays) are denoted by calligraphic letters. A real-valued N-mode tensor is denoted by, for example, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$; and its elements are denoted either by x_i with index vector $i = (i_1, i_2, \ldots, i_N)$ or more explicitly by $x_{i_1 i_2 \cdots i_N}$ depending on the circumstance (as it is occasionally useful to identify specific index positions). A superdiagonal tensor of ones is denoted by $\mathrm{diag}(1) = \mathcal{I}$. Matrices are denoted by capital letters, such as A, and vectors are denoted by lowercase letters, such as a; their elements are denoted using subscripts (a_{ij} and a_i). For matrices, rows and columns are denoted by single subscripts and the designation of row or column is contextually implicit.

The outer product of two vectors, a and b, is denoted by $a \circ b$, and their elementwise, or Hadamard product by $a * b$. The Hadamard product is also defined for matrices and tensors, and the same notation is used. The mode product between a tensor, $\mathcal{X}$, and a matrix, $A \in \mathbb{R}^{M \times I_n}$, (conformal in the column dimension with the salient mode dimension) is denoted by $\mathcal{X} \times_n A$ and defined elementwise by $(\mathcal{X} \times_n A)_{i_1 \cdots j \cdots i_N} = \sum_{i_n=1}^{I_n} x_i \, A_{ji}$. The mode product is commutative.

An effort is made to set and follow some implicit, but imperfectly observed conventions with regard to choices of Roman and Greek letters; for example, N is reserved for the number of modes of a tensor, and n as the corresponding index. Standard notation is used for other operations (such as norms) and clarified where needed.

CANDECOMP/PARAFAC (CP) Decomposition. Tensor factorizations stipulate simple latent structures that in their algebraic combination produce higher-dimensional arrays of apparently greater complexity. There are numerous existing factorization methods differing in the form of the stipulated latent structures and in the method of their combination (for an overview, see Kolda and Bader [2009]). The CP decomposition is the simplest and most interpretable method of factorization,

and is chosen here for this parsimony. The CP decomposition of a tensor, $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, into latent factor matrices $\left\{A^{(n)} = \left(a_1^{(n)} \ \cdots \ a_R^{(n)}\right) : a_r^{(n)} \in \mathbb{R}^{I_n}\right\}_{n=1}^{N}$ is:

$$\mathcal{Y} = \mathcal{X} + \mathcal{Z} \ , \quad \text{where} \quad \mathcal{X} = \sum_{r=1}^{R} a_r^{(1)} \circ a_r^{(2)} \circ \cdots \circ a_r^{(N)}$$

In general, an N-mode tensor is of rank one if it can be expressed as the compound outer product of N vectors; a CP decomposition expresses a tensor as a sum of R rank one tensors with a noise tensor $\mathcal{Z}$, and for this reason R is referred to as the rank of the decomposition.

Another useful representation of the CP decomposition is obtained by normalizing the columns of the factor matrices, $\{A^{(n)}\}_{n=1}^{N}$, to unit length. With $\lambda_r = \left(\prod_{n=1}^{N} \| \, a_r^{(n)} \, \|_2\right)^{-1}$ as the weight coefficients and $\tilde{a}_r^{(n)} = \frac{a_r^{(n)}}{\|a_r^{(n)}\|_2}$ as the normalized latent factor columns, $\mathcal{X}$ can be written as a weighted sum of unit-norm rank one tensors, $\mathcal{X} = \sum_{r=1}^{R} \lambda_r \, \tilde{a}_r^{(1)} \circ \tilde{a}_r^{(2)} \circ \cdots \circ \tilde{a}_r^{(N)}$.

The CP decomposition provides a framework for separating an observed tensor, $\mathcal{Y}$, into signal, $\mathcal{X}$, and noise, $\mathcal{Z}$, components in which the signal component emerges from lower-dimensional embedded structures. $\mathcal{X}$ can be viewed as an approximate reconstruction of $\mathcal{Y}$, or alternatively as a reconstruction of $\mathcal{Y}$ from the latent factors $\{A^{(n)}\}_{n=1}^{N}$. A natural strategy for finding a CP decomposition is to search for latent factors that minimize the reconstruction error as measured by a particular loss function. Considering Euclidean distance as the measure of error, the following optimization problem is given:

$$\begin{aligned} \text{minimize}_{\lambda, \tilde{A}^{(1)}, \ldots, \tilde{A}^{(N)}} \quad & \| \, \mathcal{Y} - \mathcal{X} \, \|_F^2 \\ \text{subject to} \quad & \| \, \tilde{a}_r^{(n)} \, \|_2^2 - 1 = 0 \ , \quad r = 1, \ldots, R \ , \ n = 1, \ldots, N \end{aligned}$$

This is the rank-R CP problem; it is non-convex. Unfolding the tensor $\mathcal{Y}$ along the nth mode by concatenating the columns in mode n yields a classical least squares problem in $\tilde{A}^{(n)}$ with a closed-form solution (although its calculation involves computationally intensive matrix operations Kaya and Uçar [2015]). Local optima are most commonly computed by iteratively updating the factor matrices one at a time via computing the closed form solution to the least squares problem resulting from the corresponding tensor unfolding. This technique is known as Alternating Least Squares (ALS).

Rank-one approximation and tensor power method. The rank-R CP problem can be solved using a variant of ALS known as the power method (Allen 2012a; Allen 2012b). The power method for a rank-R CP problem amounts to computing a series of R rank-1 approximations – that is, finding solutions to a series of rank-1 CP problems. This approach is desirable for several reasons. First, ALS updates are much less computationally intensive for rank-1 CP problems than for rank-R CP problems, which allows for fast implementations. Second, rank-1 CP problems allow relatively straightforward derivation of a number of useful properties of their optima via Karush-Kuhn-Tucker (KKT) conditions – a set of necessary conditions that must hold of any

optima. These derivations can lead to closed-form updates even under additional constraints; in this sense, the approach is flexible. Third, the method tends to produce a set of rank-1 components ordered by diminishing contribution to the quality of the reconstruction $\mathcal{X}$, which can be useful in selection of R. The following develops the power method in full detail, beginning with the ALS technique for computing a rank-1 approximation.

The rank-1 CP problem is:

$$\begin{aligned} \text{minimize}_{\lambda,\tilde{a}^{(1)},\ldots,\tilde{a}^{(N)}} \quad & \left\| \mathcal{Y} - \lambda \tilde{a}^{(1)} \circ \cdots \circ \tilde{a}^{(N)} \right\|_F^2 \\ \text{subject to} \quad & \left\| \tilde{a}^{(n)} \right\|_2^2 - 1 = 0 \ , \ n = 1, \ldots, N \end{aligned}$$

It is possible to rewrite the above as a linear program by manipulating the objective function in accord with necessary optimality properties established by the KKT conditions. The details of these manipulations are reproduced in the appendix. The result is a class of equivalent optimization problems indexed in n:

$$\begin{aligned} \text{minimize}_{\lambda,\tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^T b_n \\ \text{subject to} \quad & \left\| \tilde{a}^{(n)} \right\|_2^2 - 1 = 0 \end{aligned}$$

It is straightforward to derive that the optimum is $\tilde{a}^{(n)} = \frac{b_n}{\|b_n\|_2}$ where $\mathrm{b_n} = \left(\mathcal{Y} \underset{\mathrm{k \neq n}}{\times} \left(\tilde{\mathrm{a}}^{(\mathrm{k})}\right)^{\mathrm{T}}\right)$. The ALS technique for the rank-1 CP problem consists in iteratively updating $\tilde{a}^{(n)}$ by cycling through the modes $n = 1, \ldots, N$ and computing this solution.

Any rank-R CP decomposition has an expansion in which each rank-one component $\mathcal{X}^{(r)} = \lambda_r \tilde{a}_r^{(1)} \circ \cdots \circ \tilde{a}_r^{(N)}$ is written explicitly:

$$\mathcal{Y} = \mathcal{X}^{(1)} + \mathcal{X}^{(2)} + \cdots + \mathcal{X}^{(R)} + \mathcal{Z}$$

This expansion facilitates definition of cumulative residual terms: for $r = 1,2, \ldots, R$ and with the rth residual term is $\mathcal{Z}^{(r)} = \mathcal{Y} - \sum_{k \leq r} \mathcal{X}^{(k)}$. Then, for each consecutive r, a rank-1 CP decomposition of the previous residual tensor, $\mathcal{Z}^{(r-1)}$, is solved to find the current rank-one component tensor, $\mathcal{X}^{(r)}$, and residual tensor, $\mathcal{Z}^{(r)}$, with:

$$\mathcal{Z}^{(r-1)} = \mathcal{X}^{(r)} + \mathcal{Z}^{(r)}$$

The power method, beginning with $r = 1$, sequentially applies the ALS technique for rank-1 approximation to $\mathcal{Z}^{(r-1)}$ to obtain the rank-1 component $\mathcal{X}^{(r)}$ and increments r until reaching $r = R$. The method is described in full detail in Algorithm 1.

Algorithm 1: Tensor Power Method

Input:
data $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$;
decomposition rank R
for $r = 1$ to R **do**
initialize $\{\tilde{a}_r^{(n)}\}_{n=1}^N$;
repeat
for $n = 1$ to N **do**
$b_n \leftarrow \left(\mathcal{Z}^{(r-1)} \times_{k_{k \neq n}} (\tilde{a}^{(k)})^T\right)$;
$\tilde{a}^{(n)} \leftarrow \frac{b_n}{\|b_n\|_2}$
end for
until convergence criterion met
$\lambda_r \leftarrow \left(\mathcal{Z}^{(r-1)} \times_{n_{n=1}}^N (\tilde{a}^{(n)})^T\right)$
$\mathcal{X}^{(r)} \leftarrow \lambda_r \times_{n_{n=1}}^N \tilde{a}_r^{(n)}$
$\mathcal{Z}^{(r)} \leftarrow \mathcal{Z}^{(r-1)} - \mathcal{X}^{(r)}$
end for
Output:
reconstruction $\mathcal{X}$;
factor matrices $\{A^{(n)}\}_{n=1}^N$

METHODS: A decomposition method was designed to preserve the spatiotemporal coherence of trajectories when represented as tensors. It begins with a data representation framework for converting raw trajectory information into tensor format; subsequently, the tensor power method is extended by deriving alternative update rules that enforce constraints motivated by the distinctive properties of tensor representations of trajectory data.

Trajectories as tensors. A trajectory is a path in space and time traversed by an object in motion. A single trajectory containing J points can be expressed as a set of points $\{(l_j, t_j)\}_{j \in J}$, where l_j describes location and t_j describes time; the coordinate (l_j, t_j) indicates the object's presence at the given place and time. The following methodological exposition is given in terms of trajectories on a two-dimensional surface so that $l_j = (x_j, y_j)$, although the methods are fully general.

Let $\{(x_j, y_j, t_j)\}_{j \in J}$ be an observed trajectory regularly sampled in time. In practice, a path is observed with finite precision; in keeping, J is an enumerable index set. Let X, Y, T be sets of contiguous intervals of widths $\Delta x, \Delta y, \Delta t$ that discretize the spatiotemporal extent of the data. Each element, $y_{i_1 i_2 i_3}$, of trajectory tensor $\mathcal{Y} \in \mathbb{R}^{|X| \times |Y| \times |T|}$ is defined by:

$$y_{i_1 i_2 i_3} = \sum_{j \in J} \mathbf{1}\{(x_j, y_j, t_j) \in X_{i_1} \times Y_{i_2} \times T_{i_3}\}$$

The entries, $y_{i_1 i_2 i_3}$, are proportional to the duration that the moving object was in a particular region during a particular time window. Multiple trajectories can be represented by higher-order tensors, with an additional mode indexing trajectory identity.

By extension, any number of categorical or discrete variables can be used to generate increasingly higher-order trajectory tensors in which the additional dimensions beyond space and time jointly

index the values of other variables. This provides a highly flexible framework for merging trajectory data with categorical information or categorical representations of quantitative information.

An N-mode trajectory tensor constructed as above has a number of distinctive attributes. First, it comprises almost entirely of zeros, especially in the spatial modes, as moving objects visit only a few locations in a small time interval. Second, it exhibits smoothness in the temporal mode because moving objects make continuous transitions through spatial regions over time.

Sparsity constraint. A modified method of computing CP decompositions that enforces these properties —appropriate sparsity and smoothness in the signal component —can consequently be expected to produce better approximations. The following subsections establish constraints on the latent factors that induce the desired properties in the reconstruction and derive alternative updates in the tensor power method that adhere to these constraints.

The first property of the reconstruction to enforce is (spatial) sparsity. Let $\mathcal{X}$ be the signal component of a rank-R CP decomposition of an arbitrary tensor $\mathcal{Y}$, as before. If $\mathcal{X}$ is sparse, then the latent factors must be such that $x_i = 0$ for many i. Elementwise, $\mathcal{X}$ is specifiable as a Hadamard product of the rows of the factor matrices $\{A^{(n)}\}_{n=1}^{N}$ that correspond to the ith entry position $x_i = \lambda^T \left(\tilde{a}_{i_1}^{(1)} * \cdots * \tilde{a}_{i_N}^{(N)}\right)$, and if $\left(\tilde{a}_{i_1}^{(1)} * \cdots * \tilde{a}_{i_N}^{(N)}\right) = 0$, then $x_i = 0$. Therefore, sparsity in the factor matrices generally induces sparsity in the reconstruction.[1]

A sparse variant on the tensor power method can be obtained by adjusting constraints in the rank-1 CP problem and repeating the calculations involved in deriving update rules for the power method (Allen 2012a). In other words, a sparse CP decomposition can be computed by replacing the usual power method update with the solution to the modified subproblem:

$$\begin{aligned} \text{minimize}_{\lambda,\tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^T b_n \\ \text{subject to} \quad & \left\|\tilde{a}^{(n)}\right\|_2^2 - 1 \leq 0 \\ & \frac{1}{t}\left\|\tilde{a}^{(n)}\right\|_1 - 1 \leq 0 \end{aligned}$$

The equality constraint is relaxed to simplify the derivation of a solution — details are reproduced in full in the appendix — and t is a hyperparameter determining the degree of sparsity. The solution to this modified subproblem is given by:

$$\tilde{a}^{(n)} = \frac{S(b_n, \gamma)}{\left\|S(b_n, \gamma)\right\|_2}$$

[1] The relationship between sparsity in the factor matrix for one mode and sparsity in the reconstruction can be modulated by the sparsity patterns of factor matrices for other modes. For example, setting $\tilde{a}_{kr}^{(s)}$ to zero for most r, except, say, for $r = 1,2$, reproduces infrequent visits to location k only if there are indices i with $i_s = k$ for which $\tilde{a}_{i_n 1}^{(n)} \neq 0$ and $\tilde{a}_{i_n 2}^{(n)} \neq 0$ for all $n \neq s$ (and otherwise yields no visits to k). If there are many such indices i, the reconstruction is dense in location k. Just as other modes can modulate sparsity in position $i_s = k$, other modes can modulate density.

In the update rule, $S(x,\gamma)$ denotes the soft-threshold operator $\text{sign}(x)(|x|-\gamma)_+$ applied elementwise in the first argument; γ is a hyperparameter proportional to t.[1]

Smoothness constraint. The second constraint is (temporal) smoothness. When understood in the sense of similarity between nearby entries, smoothness in the reconstruction $\mathcal{X}$ reduces to smoothness in the latent factors. The difference between two entries x_i and $x_{i'}$ when i and i' differ in only one mode (say, the first mode) can be written directly in terms of the corresponding factors. Noting that $x_{i_1\cdots i_N} = \sum_{r=1}^{R} a^{(1)}_{i_1 r}\cdots a^{(N)}_{i_N r}$, algebraic expansion and rearrangement yields that $x_{i_1\cdots i_N} - x_{i_1',\cdots i_N} = \left(a^{(2)}_{i_2} * \cdots * a^{(N)}_{i_N}\right)^T \left(a^{(1)}_{i_1} - a^{(1)}_{i_1'}\right)$. It follows that $\left\|x_{i_1\cdots i_N} - x_{i_1',\cdots i_N}\right\|_2 \leq \|d\|_2 \left\|a^{(1)}_{i_1} - a^{(1)}_{i_1'}\right\|_2$ where $d = \left(a^{(2)}_{i_2} * \cdots * a^{(N)}_{i_N}\right)^T$. So, the distances between rows of the factor matrix $\| a^{(1)}_{i_1 r} - a^{(1)}_{i_1' r} \|_2$ bound the distances between entries in corresponding positions. Accordingly, smoothness in the columns of a factor matrix induces smoothness in the reconstruction in the corresponding mode.

In this scenario, temporal smoothness means any two entries, x_i and $x_{i'}$, that differ only in time, are of increasing similarity with increasing temporal closeness $|i-i'|$. For example, if the decay were linear, this property could be expressed as $\left\|x_{i_1\cdots i_N} - x_{i_1',\cdots i_N}\right\|_2 \propto |i_1 - i_1'|$. More generally, the property can be described by replacing $|i_1 - i_1'|$ with an arbitrary monotonic function w: $\left\|x_{i_1\cdots i_N} - x_{i_1',\cdots i_N}\right\|_2 \propto w(|i_1 - i_1'|)^{-1}$ where $w(k) > w(k+1)$ for every $k \in \mathbb{N}$. Extending this idea across all entries at a given lag and across multiple lags motivates the following definition. $\mathcal{X}$ is temporally smooth if for suitable c:

$$\sum_{k=1}^{K} w_k \sum_{|i_1 - i_1'|=k} \left\|x_{i_1\cdots i_N} - x_{i_1',\cdots i_N}\right\|_2^2 < c$$

This property is expressible in terms of the factors, and gives a smoothness constraint that can be directly appended to the CP decomposition.

First, consider entrywise differences $x_i - x_{i'}$ one lag apart in the time index i_1, so that $|i_1 - i_1'| = 1$, and let $R = 1$. From the foregoing:

$$\sum_{|i_1 - i_1'|=1} \left\|x_{i_1\cdots i_N} - x_{i_1',\cdots i_N}\right\|_2^2 \propto \left(a^{(1)}_1 - a^{(1)}_2\right)^2 + \left(a^{(1)}_2 - a^{(1)}_3\right)^2 + \cdots + \left(a^{(1)}_{I_1-1} - a^{(1)}_{I_1}\right)^2$$

This latter expression can be written compactly in terms of a lag-1 differencing matrix D:

[1] Furthermore, t must be well-specified such that $\|S(b_n,\gamma)\|_2 > 0$. As this is not always ensured in practice when tuning γ, the heuristic of reverting to the unconstrained update rule whenever $S(b_n,\gamma) = 0$ is adopted here. While the literature suggests setting $\tilde{a}^{(n)} = 0$ instead, this is ill-advised since after such an update $b_n = 0$ and therefore all factors are set to zero thereafter.

$$Da^{(1)} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & -1 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & -1 & 0 \\ 0 & 0 & 0 & \cdots & 1 & -1 \end{pmatrix} \begin{pmatrix} a_1^{(1)} \\ a_2^{(1)} \\ a_3^{(1)} \\ \vdots \\ a_{(I_1-1)}^{(1)} \\ a_{I_1}^{(1)} \end{pmatrix} = \begin{pmatrix} a_1^{(1)} - a_2^{(1)} \\ a_2^{(1)} - a_3^{(1)} \\ \vdots \\ a_{(I_1-1)}^{(1)} - a_{I_1}^{(1)} \end{pmatrix}$$

Therefore:

$$\sum_{|i_1 - i_1'|=1} \left\| x_{i_1 \cdots i_N} - x_{i_1' \cdots i_N} \right\|_2^2 \propto \| \, Da^{(1)} \, \|_2^2$$

More generally, the sum of lag-k differences can be expressed similarly in terms of a lag-k differencing matrix D_k. The main diagonal entries of D_k are identically 1, and the kth superdiagonal entries are identically -1. Now, the smoothness property can be written:

$$\sum_{k=1}^{K} w_k \, \| \, D_k a^{(1)} \, \|_2^2 < c$$

An update rule adhering to this constraint for mode n can be obtained by modifying the core rank-1 subproblem as:

$$\begin{aligned} \text{minimize}_{\lambda, \tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^T b_n \\ \text{subject to} \quad & \left\| \tilde{a}^{(n)} \right\|_2^2 - 1 \leq 0 \\ & \frac{1}{c} \sum_{k=1}^{K} w_k \, \| \, D_k a^{(1)} \, \|_2^2 - 1 \leq 0 \end{aligned}$$

The solution, derived in detail in the appendix, is with $Q = \sum_k w_k \, D_k^T D_k$:

$$\tilde{a}^{(n)} = \frac{(I + \gamma Q)^{-1} b_n}{\left\| (I + \gamma Q)^{-1} b_n \right\|_2}$$

It is assumed that the lag weights w_k are known; γ is a hyperparameter proportional to c.

Tensor power algorithm with mixed mode constraints. The sections above established mode-wise smooth and mode-wise sparse update rules that can be substituted in the tensor power method (Algorithm 2) to obtain a method for computing a CP decomposition abiding a mixture of constraints across modes. The present motivation for doing so is enforcing smoothness in the temporal mode and sparsity in the spatial modes may produce a CP decomposition that produces a better approximation of duration tensors than unconstrained decomposition. Assume the first two modes are spatial and the third is temporal. Then, the following constraints modify the CP problem to search only for optima that are spatiotemporally coherent:

$$
\begin{aligned}
\text{minimize}_{\lambda, A^{(1)}, \ldots, A^{(N)}} \quad & \| \mathcal{Y} - \mathcal{X} \|_F^2 \\
\text{subject to} \quad & \| a_r^{(n)} \|_2^2 - 1 \leq 0 \ , \quad r = 1, \ldots, R \ , \ n = 1, \ldots, N \\
& \frac{1}{t} \| a_r^{(n)} \|_1 - 1 \leq 0 \ , \quad r = 1, \ldots, R \ , \ n = 1{,}2 \\
& \frac{1}{c} \sum_k w_k \| D_k a_r^{(n)} \|_2^2 - 1 \leq 0 \ , \quad r = 1, \ldots, R \ , \ n = 3
\end{aligned}
$$

Algorithm 2 provides a method of solution based on the tensor power method. It is written in a broader generality to emphasize this extension of the tensor power method to accommodate mixed constraints is further generalizable to any number of update rules beyond those considered here. Each of the unconstrained, sparse, and smooth updates derived in the prior sections are functions of b_n and possibly hyperparameters, so the algorithm is written in terms of a set of update functions that maps one-to-one with the modes of the input data tensor. It is assumed, at present, that $\text{update}_n(\cdot)$ is one of the three rules derived above, but this need not be the case.

Algorithm 2: Tensor Power Method with Mixed Mode Constraints

Input:
data $\mathcal{Y} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$;
decomposition rank R;
sparsity hyperparameters $\gamma \in \mathbb{R}^N$;
smoothness hyperparameters $\nu \in \mathbb{R}^N$;
smoothness weights $w \in \mathbb{R}^{N \times K}$;
update rules $\{\text{update}_n(\cdot)\}_{n=1}^N$
for $r = 1$ to R **do**
  initialize $\{\tilde{a}_r^{(n)}\}_{n=1}^N$;
  repeat
    for $n = 1$ to N **do**
      $b_n \leftarrow \left(\mathcal{Z}^{(r-1)} \times_{k \neq n} (\tilde{a}^{(k)})^T\right)$;
      $\tilde{a}^{(n)} \leftarrow \text{update}_n(b_n, \gamma_n, \nu_n, w_n)$
    end for
  until convergence criterion met
  $\lambda_r \leftarrow \left(\mathcal{Z}^{(r-1)} \times_{n=1}^N (\tilde{a}^{(n)})^T\right)$
  $\mathcal{X}^{(r)} \leftarrow \lambda_r \times_{n=1}^N \tilde{a}_r^{(n)}$
  $\mathcal{Z}^{(r)} \leftarrow \mathcal{Z}^{(r-1)} - \mathcal{X}^{(r)}$
end for
Output:
reconstruction $\mathcal{X}$;
factor matrices $\{A^{(n)}\}_{n=1}^N$

APPLICATIONS: To illustrate performance and suggest possible applications of the methodology above, a small collection of trajectories were extracted from a dataset of 1.7 million taxi trips in the city of Porto, Portugal. These were decomposed using constrained and unconstrained versions of the method, both alone and in small collections. A few examples are reproduced in this section.

The taxi trips were recorded via GPS at a resolution of 15 sec intervals between location measurements. For the purposes of decomposition, these trajectories were represented in a duration tensor with a temporal discretization of 2 min and a spatial discretization comprising a 25×25 grid of the spatial extent of the extracted trajectories.

Single-trajectory decomposition, summarized in Figure 1, indicates that temporally smooth and spatially sparse reconstructions achieve lower or comparable reconstruction errors compared with unconstrained reconstructions. This shows that constraining the decomposition does not diminish the quality of the reconstruction. Qualitatively, the novel modifications also better capture spatial patterns in the tensor and thereby produce a more faithful representation of the trajectory by reducing noise in unvisited locations. The spatial patterns are clearer due to the reduction of small, non-zero, noisy entries in the constrained reconstruction produced by the novel method as compared with the unconstrained reconstruction, as shown in Figure 2.

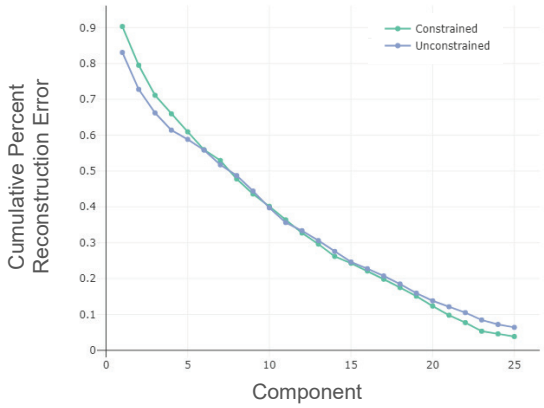

Figure 1. Comparison of reconstruction errors as a function of decomposition rank R.

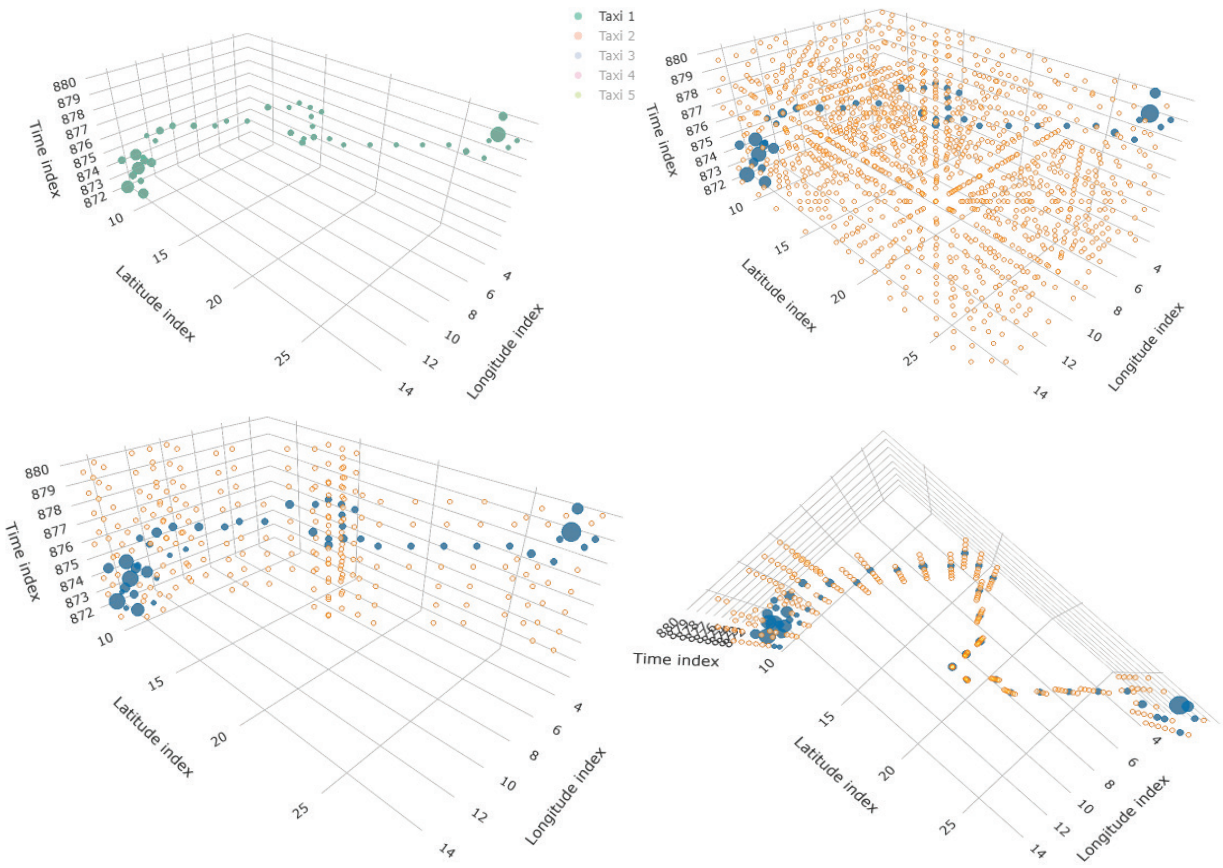

Figure 2. Top left, trajectory tensor of one taxi trip; top right, reconstruction via unconstrained CP decomposition; bottom left and right, two views of reconstruction via constrained CP decomposition. Small, but non-zero entries are designated by orange points; the constrained version captures the spatial pattern of the trajectory and slightly lowers the reconstruction error.

Figure 3 indicates that multi-trajectory decomposition of two nearby trajectories in space and time produce a set of shared rank-1 components $\mathcal{X}^{(r)}$ that capture the common portions of the trajectories. This latter feature of the decomposition outputs in the multi-trajectory case can be leveraged to allow for an efficient identification of spatiotemporal similarity among a set of paths. In this example, the norm of the reconstruction error from such shared components only is roughly 30% of the norm of the data tensor; thus, the shared components capture about 70% of the data tensor structure.

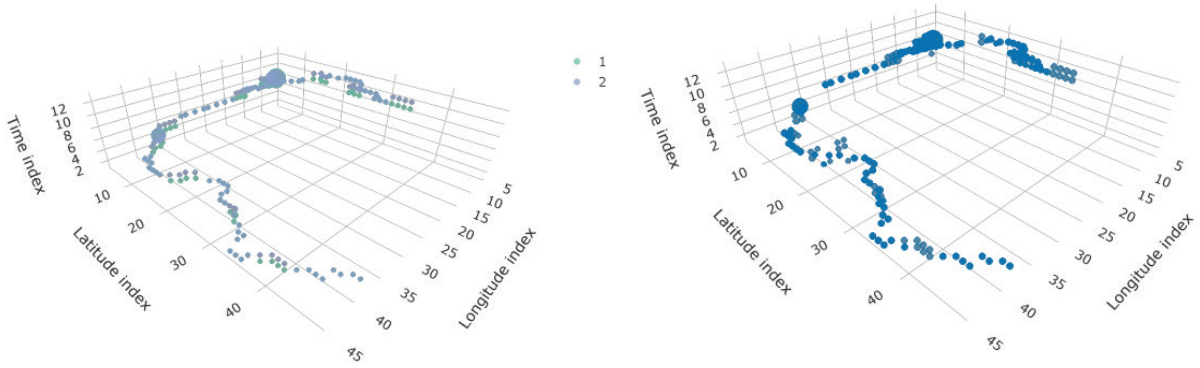

Figure 3. Left, duration tensor representation of two otherwise identical trajectories lagged by one minute in time; right, reconstruction subtensor obtained from summing $\mathcal{X}^{(r)}$ over all r for which both vehicle latent factors are nonzero.

To extend this idea further, let $\mathcal{Y} \in \mathbb{R}^{I_1 \times I_2 \times I_3 \times I_4}$ be a duration tensor where i_1, i_2, i_3 index space and time, and i_4 indexes vehicle. From a CP decomposition that is sparse in the vehicle mode, define the sets of rank-1 components associated with the i_4th vehicle as $R_{i_4} = \{r : a^{(4)}_{i_4 r} \neq 0\}$. The overlap between two sets R_{i_4}, R_{i_4}, corresponds to the similarity of the trajectories of vehicles $i_4, {i_4}'$ in the reconstruction. If the overlap is perfect, the corresponding (reconstructed) trajectories differ only according to the magnitudes of the latent factors – that is, the vehicles traverse the discretization in exactly the same order and to within a rescaling of the duration spent in each spatiotemporal region. Thus, the Jaccard index between R_{i_4}, R_{i_4}, provides a coarse measure of the spatiotemporal similarity of paths traversed by any two vehicles $i_4, {i_4}'$ in $\mathcal{Y}$ as represented by the latent factors in the outputs of a CP decomposition. To illustrate this, Figure 4 depicts the Jaccard index as a function of temporal shifts between eleven trajectories generated by shifting a single trajectory by one minute, 10 consecutive times.[1] The figure demonstrates a steep drop in Jaccard index with time lag and a plateau after about three minutes; while a similar phenomenon is expected for any collection of nearby paths, the relationship may hold on different temporal scales depending on the discretization used for the decomposition. This suggests one way the decomposition outputs could be leveraged in clustering.

[1] For the decomposition, the eleven trajectories were represented in a duration tensor on a 3-minute temporal discretization and a 25×25 spatial grid, and $R = 100$ components were used, yielding a reconstruction with about 35% error.

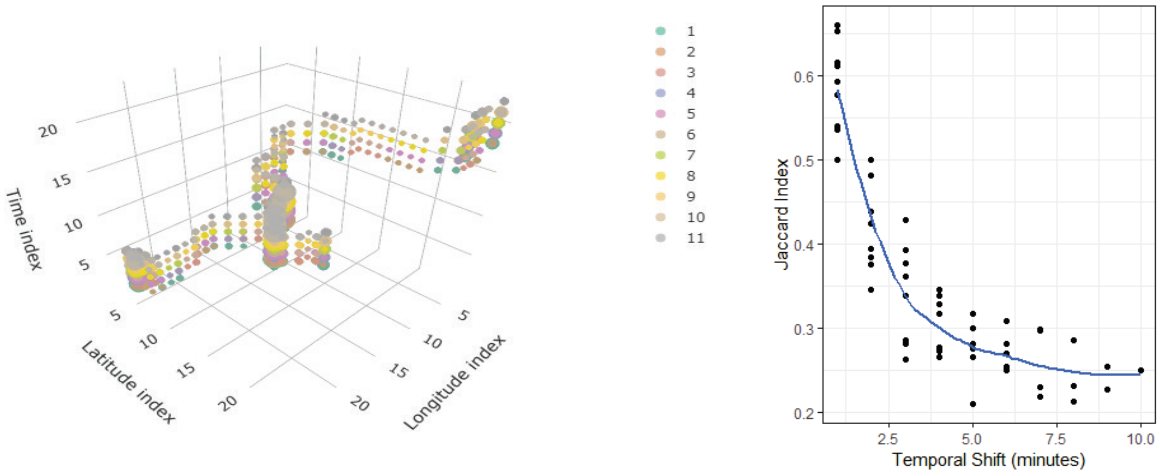

Figure 4. Left, visualization of duration tensor representing eleven identical trajectories lagged by consecutive one-minute intervals; right, Jaccard index between $R_{i_4}, R_{i_4'}$ for all $i_4, i_4' = 1, \ldots, 11$ with $i_4 \neq i_4'$ as a function of time shift with a local regression conditional mean overlaid to aid in identifying the trend.

DISCUSSION: This technical note (TN) lays groundwork to enable representation of large trajectory datasets containing additional covariate information and conduct exploratory analysis aimed at pattern detection with a minimum of assumptions about the data while preserving basic spatiotemporal regularity properties. Specifically, the methods developed in this document allow for spatial sparsity and temporal smoothness to be maintained when approximating trajectory data with latent features using a CP tensor decomposition. While the focus is on developing appropriate constraints for isolated trajectories, the methods naturally extend to approximation of trajectory data merged with non-spatiotemporal information in high-order tensors without significant modification. The tests on Porto taxi data illustrated the quantitative reconstruction was comparable or better for the constrained decomposition. The value of the constraints was made clear in the reduction of noise and assistance in uncovering spatiotemporal trends. Further work on hyperparameter selection and the development of scalable computations would greatly advance the foregoing research in that direction and facilitate approximations of large collections of trajectories. More broadly, these approximations could potentially support a variety of pattern mining tasks. The ability to fuse and decompose trajectory data while including known properties, such as spatial and temporal coherence, potentially provides insight into patterns of human behavior, a task of interest to the Army GEOINT and HUMINT domains.

ACKNOWLEDGMENTS: Mr. Trevor Ruiz thanks the NSF and ORISE for funding his research through the MSGI internship program.

REFERENCES

Acar, E., and B. Yener. 2008. "Unsupervised multiway data analysis: A literature survey." *IEEE Transactions on Knowledge and Data Engineering* 21(1):6–20.

Acar, E., D. M. Dunlavy, T. G. Kolda, and M. Mørup. 2011. "Scalable tensor factorizations for incomplete data." *Chemometrics and Intelligent Laboratory Systems* 106(1):41–56.

Acar, E., T. G. Kolda, and D. M. Dunlavy. 2011. "All-at-once optimization for coupled matrix and tensor factorizations." *ArXiv /1105.3422*.

Allen, G. I. 2012a. "Sparse higher-order principal components analysis." *Artificial Intelligence and Statistics* 27–36.

Allen, G. I. 2012b. "Regularized tensor factorizations and higher-order principal components analysis." *ArXiv /1202.2476*.

Ceulemans, E., and H. A. Kiers. 2006. "Selecting among three-mode principal component models of different types and complexities: A numerical convex hull based method." *British Journal of Mathematical and Statistical Psychology* 59(1):133–150.

De Brébisson, A., É. Simon, A. Auvolat, P. Vincent, and Y. Bengio. 2015. "Artificial neural networks applied to taxi destination prediction." *ArXiv /1508.00021*.

De Lathauwer, L., B. De Moor, and J. Vandewalle. 2000a. "On the best rank-1 and rank-$(R_1 R_2, \ldots, R_N)$ approximation of higher-order tensors." *SIAM Journal on Matrix Analysis and Applications* 21(4):1324–1342.

De Lathauwer L., B. De Moor, and J. Vandewalle. 2000b. "A multilinear singular value decomposition." *SIAM Journal on Matrix Analysis and Applications* 21(4):1253–1278.

Deng, Z., and M. Ji. 2011. "Spatiotemporal structure of taxi services in shanghai: Using exploratory spatial data analysis." In *2011 19th International Conference on Geoinformatics* pg. 1–5.

Hong, D., T. G. Kolda, and J. A. Duersch. 2018. "Generalized canonical polyadic tensor decomposition." *ArXiv /1808.07452*.

Kaya, O., and B. Uçar. 2015. "Scalable sparse tensor decompositions in distributed memory systems." In *SC'15: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pg. 1–11.

Kolda, T. G., and B. W. Bader. 2009. "Tensor decompositions and applications." *SIAM review*, 51(3):455–500.

Luo, Y., D. Tao, K. Ramamohanarao, C. Xu, and Y. Wen. 2015. "Tensor canonical correlation analysis for multi-view dimension reduction." *IEEE Transactions on Knowledge and Data Engineering* 27(11):3111–3124.

Moreira-Matias, Luis., J. Gama, M. Ferreira, J. Mendes-Moreira, and L. Damas. 2013. "Predicting taxi–passenger demand using streaming data." *IEEE Transactions on Intelligent Transportation Systems* 14(3):1393–1402.

Narita, A., K. Hayashi, R. Tomioka, and H. Kashima. 2012. "Tensor factorization using auxiliary information." *Data Mining and Knowledge Discovery* 25(2):298–324.

Park, M., J-G. Jang, and S. Lee. 2019. "Vest: Very sparse tucker factorization of large-scale tensors." *ArXiv/1904.02603*.

Scharf, H. R., M. B. Hooten, B. K. Fosdick, D. S. Johnson, J. M. London, J. W. Durban. 2016. "Dynamic social networks based on movement." *The Annals of Applied Statistics* 10(4):2182–2202.

Scharf, H. R., M. B. Hooten, D. S. Johnson, and J. W. Durban. 2018. "Process convolution approaches for modeling interacting trajectories." *Environmetrics* 29(3):e2487.

Shashua, A., and T. Hazan. 2005. "Non-negative tensor factorization with applications to statistics and computer vision." In *Proceedings of the 22nd International Conference on Machine learning*, pg. 792–799.

Shi, C., W. Lu, and R. Song. 2019. "Determining the number of latent factors in statistical multi-relational learning." *The Journal of Machine Learning Research* 20(1):809–846.

Sun, L., and K. W. Axhausen. 2016. "Understanding urban mobility patterns with a probabilistic tensor factorization framework." *Transportation Research Part B: Methodological* 91:511–524.

Sun, W. W., J. Lu, H. Liu, and G. Cheng. 2017. "Provable sparse tensor decomposition." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79(3):899–916.

Wang, Y., Y. Zheng, and Y. Xue. 2014. "Travel time estimation of a path using sparse trajectories." In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pg. 25–34.

Yılmaz, K. Y., A. T. Cemgil, and U. Simsekli. 2011. "Generalised coupled tensor factorisation." In *Advances in Neural Information Processing Systems*, pg. 2151–2159.

Zhang, A., and R. Han. 2019. "Optimal sparse singular value decomposition for high-dimensional high-order data." *Journal of the American Statistical Association*, 1–34.

Zheng, Y. 2015. "Trajectory data mining: an overview." *ACM Transactions on Intelligent Systems and Technology (TIST)* 6(3):29.

Zheng, Y., and X. Zhou. 2011. "Computing with spatial trajectories." *Springer Science & Business Media*.

Zheng, Y., T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang. 2014. "Diagnosing New York City's noises with ubiquitous data." In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pg. 715–725.

Zhou, T., H. Shan, A. Banerjee, and G. Sapiro. 2012. "Kernelized probabilistic matrix factorization: Exploiting graphs and side information." In *Proceedings of the 2012 SIAM international Conference on Data mining*, pg. 403–414.

APPENDIX: SUPPLEMENT TO SPATIOTEMPORALLY COHERENT TENSOR DECOMPOSITIONS FOR THE ANALYSIS OF TRAJECTORY DATA

Detailed derivations pertinent to the preliminaries and methods sections are reproduced here under matching section headers.

BACKGROUND

Rank-One Approximation and Tensor Power Method. The following gives a detailed derivation of the update rules for the ALS technique in the rank-1 CP problem, beginning with manipulations of the objective function to obtain a linear program. It can be derived from the KKT conditions that any optima must satisfy the following property (De Lathauwer 2000a):

$$\lambda = \mathcal{Y} \underset{1}{\times} \left(\tilde{a}^{(1)}\right)^{T} \underset{2}{\times} \cdots \underset{N}{\times} \left(\tilde{a}^{(N)}\right)^{T}$$

For short, denote a composite of consecutive mode products over the set of modes S by $\mathcal{X} \times_{n \in S} A^{(n)}$. As before, denote $\lambda \times_{n=1,\ldots,N} \tilde{a}^{(n)}$ by $\mathcal{X}$. Then since:

$$\begin{aligned}
\| \mathcal{Y} - \mathcal{X} \|_F^2 &= \| \mathcal{Y} \|_F^2 - 2\langle \mathcal{Y}, \mathcal{X} \rangle + \| \mathcal{X} \|_F^2 \\
\langle \mathcal{Y}, \mathcal{X} \rangle &= \sum_{i_1,\ldots,i_N} y_{i_1,\ldots,i_N} \lambda \tilde{a}_{i_1}^{(1)} \cdots \tilde{a}_{i_N}^{(N)} \\
&= \lambda \sum_{i_1,\ldots,i_N} y_{i_1,\ldots,i_N} \tilde{a}_{i_1}^{(1)} \cdots \tilde{a}_{i_N}^{(N)} \\
&= \lambda \left(\mathcal{Y} \underset{n=1,\ldots,N}{\times} \left(\tilde{a}^{(n)}\right)^{T} \right) \\
\| \mathcal{X} \|_F^2 &= \sum_{i_1,\ldots,i_N} \lambda^2 \left(\tilde{a}_{i_1}^{(1)}\right)^2 \cdots \left(\tilde{a}_{i_N}^{(N)}\right)^2 \\
&= \lambda^2 \left(\prod_{n=1}^{N} \| \tilde{a}^{(n)} \|_2^2 \right) \\
&= \lambda^2
\end{aligned}$$

It follows that any optima of the rank-1 CP problem must satisfy:

$$\| \mathcal{Y} - \mathcal{X} \|_F^2 = \| \mathcal{Y} \|_F^2 - \lambda^2 = \| \mathcal{Y} \|_F^2 - \left(\mathcal{Y} \underset{n=1,\ldots,N}{\times} \left(\tilde{a}^{(n)}\right)^{T} \right)^2$$

For any $n = 1, \ldots, N$ one can write:

$$\mathcal{Y} \underset{n=1,\ldots,N}{\times} \left(\tilde{a}^{(n)}\right)^{T} = \left(\tilde{a}^{(n)}\right)^{T} b_n \,, \quad \text{where} \quad b_n = \left(\mathcal{Y} \underset{k \neq n}{\times} \left(\tilde{a}^{(k)}\right)^{T} \right)$$

Now, considering the rank-1 problem only in terms of $\tilde{a}^{(n)}$ by fixing $\{\tilde{a}^{(k)}\}_{k \neq n}$, rewriting the objective function in accord with the foregoing, and observing that $\lambda > 0$ by hypothesis, one obtains the subproblem:

$$\begin{aligned} \text{minimize}_{\lambda,\tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^{T} b_n \\ \text{subject to} \quad & \|\tilde{a}^{(n)}\|_2^2 - 1 = 0 \end{aligned}$$

The KKT conditions entail that any optima must satisfy $0 = -b_n + 2\nu\tilde{a}^{(n)}$ and the constraint function. With $\nu = \| b_n \|_2/2$ these conditions hold for:

$$\tilde{a}^{(n)} = \frac{b_n}{\| b_n \|_2}$$

Since this subproblem is convex, the KKT conditions are also sufficient for optimality, so the foregoing is a global optimum for the subproblem.

METHODS

Trajectories as tensors. An alternate view of trajectory tensors as presented in the main body of the text is as an aggregated version of an indicator tensor. That view is developed here. Let $\{(x_j, y_j, t_j)\}_{j\in J}$ be an observed trajectory. Denote the distinct values of the components of the triple by X, Y, and T; and index the elements of these sets by $I_1 = \{1,2,\dots,|X|\}$, $I_2 = \{1,2,\dots,|Y|\}$, and $I_3 = \{1,2,\dots,|T|\}$. The trajectory can be encoded by a three-dimensional indicator tensor $\mathcal{Y} \in \{0,1\}^{I_1\times I_2\times I_3}$ where:

$$y_{i_1 i_2 i_3} = 1 \quad \Leftrightarrow \quad (x_j, y_j, t_j) = (X_{i_1}, Y_{i_2}, T_{i_3}) \quad \text{for some } j \in \{1,\dots,J\}$$

Trajectory data vary in spatiotemporal resolution depending on the method of observation. Yet, in many instances, aggregation of the indicator tensor to regular spatiotemporal intervals is desirable, especially when data resolution is on a finer spatiotemporal scale than the phenomenon of interest and observations are made with temporal regularity.[1] Consider discretizing the spatiotemporal extent into sets of I_1', I_2', I_3' contiguous intervals of widths $\Delta x, \Delta y, \Delta t$ generated from left endpoints x^*, y^*, t^*:

$$\begin{aligned} X &= \{[x + (i_1' - 1)\Delta x, x + (i_1')\Delta x)\}_{i_1'=1}^{I_1'} \\ Y &= \{[y + (i_2' - 1)\Delta y, y + (i_2')\Delta y)\}_{i_2'=1}^{I_2'} \\ T &= \{[t + (i_3' - 1)\Delta t, t + (i_3')\Delta t)\}_{i_3'=1}^{I_3'} \end{aligned}$$

The indicator tensor can be aggregated by summation within bins defined by the combinations of these intervals to obtain $\mathcal{Y}^*$. Indexing the elements (intervals) of X^*, Y^*, and T^* by $X^*_{i_1'}$, $Y^*_{i_2'}$, and $T^*_{i_3'}$, the aggregated version of the indicator tensor is defined elementwise by:

[1] For example, data generated by sampling an object's GPS location with precision to within a few meters on the order of seconds to capture its movement across a spatial extent on the order of kilometers on a temporal scale on the order of hours.

$$y^*_{i_1' i_2' i_3'} = \sum_{\substack{i: X_{i_1} \in X^*_{i_1'}, \\ Y_{i_1} \in Y^*_{i_2'}, \\ T_{i_1} \in T^*_{i_3'}}} y_{i_1 i_2 i_3}$$

Sparsity Constraint. The core subproblem on which update rules are based is, with $R = 1$:

$$\begin{aligned} \text{minimize}_{\lambda, \tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^T b_n \\ \text{subject to} \quad & \|\tilde{a}^{(n)}\|_2^2 - 1 = 0 \end{aligned}$$

A sparse update rule can be obtained by modifying this subproblem as:

$$\begin{aligned} \text{minimize}_{\lambda, \tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^T b_n \\ \text{subject to} \quad & \|\tilde{a}^{(n)}\|_2^2 - 1 \leq 0 \\ & \frac{1}{t}\|\tilde{a}^{(n)}\|_1 - 1 \leq 0 \end{aligned}$$

The equality constraint is relaxed to simplify the derivation of a solution satisfying the KKT conditions, and t is a hyperparameter determining the degree of sparsity. The Lagrangian is:

$$L(\tilde{a}^{(n)}, \nu) = -\left(\tilde{a}^{(n)}\right)^T b_n + \nu_1 \left(\left(\tilde{a}^{(n)}\right)^T \left(\tilde{a}^{(n)}\right) - 1\right) + \nu_2 \left(\frac{1}{t}\|\tilde{a}^{(n)}\|_1 - 1\right)$$

With $\partial\|x\|_1 = \{q: q_i = \text{sign}(x_i)\mathbf{1}_{x_i \neq 0} + (1 - c_i)\mathbf{1}_{x_i = 0}, c_i \geq 0\}$ denoting the subdifferential of $\partial\|x\|_1$, the KKT conditions are:

$$\begin{aligned} 0 &= -b_n + 2\nu_1\left(\tilde{a}^{(n)}\right) + \frac{\nu_2}{t}\partial\|\tilde{a}^{(n)}\|_1 \\ 0 &= \nu_1 \left(\left(\tilde{a}^{(n)}\right)^T \left(\tilde{a}^{(n)}\right) - 1\right) \\ 0 &= \nu_2 \left(\frac{1}{t}\|\tilde{a}^{(n)}\|_1 - 1\right) \end{aligned}$$

Then, with $\nu_1 = \|S(b_n, \nu_2/t)\|_2/2 \;\; t\nu_2 = \frac{(b_n - S(b_n, \nu_2/t))\|S(b_n, \nu_2/t)\|_2}{\partial\|S(b_n, \nu_2/t)\|_1}$, the KKT conditions are satisfied by:

$$\tilde{a}^{(n)} = \frac{S(b_n, \nu_2/t)}{\|S(b_n, \nu_2/t)\|_2}$$

Due to the multiplicative interaction between the Lagrange multiplier ν_2 and the hyperparameter t, the quantity $\frac{\nu_2}{t}$ can be treated as a hyperparameter,[1] giving the update rule:

$$\tilde{a}^{(n)} = \frac{S(b_n, \gamma)}{\|S(b_n, \gamma)\|_2}$$

$S(x, \gamma)$ denotes the soft-threshold operator $\text{sign}(x)(|x| - \gamma)_+$ applied elementwise in the first argument; γ is a hyperparameter proportional to t.[2]

Smoothness Constraint. A smooth update rule for mode n can be obtained by modifying the core rank-1 subproblem as:

$$\begin{aligned} \text{minimize}_{\lambda, \tilde{a}^{(n)}} \quad & -\left(\tilde{a}^{(n)}\right)^T b_n \\ \text{subject to} \quad & \|\tilde{a}^{(n)}\|_2^2 - 1 \leq 0 \\ & \frac{1}{c}\sum_{k=1}^{K} w_k \| D_k a^{(1)} \|_2^2 - 1 \leq 0 \end{aligned}$$

The Lagrangian is:

$$L(\tilde{a}^{(n)}, \nu) = -\left(\tilde{a}^{(n)}\right)^T b_n + \nu_1 \left(\left(\tilde{a}^{(n)}\right)^T \left(\tilde{a}^{(n)}\right) - 1\right) + \nu_2 \left(\frac{1}{c}\sum_{k=1}^{K} w_k \| D_k a^{(1)} \|_2^2 - 1\right)$$

Its gradient is:

$$\nabla L(\tilde{a}^{(n)}, \nu) = -b_n + 2\nu_1 \tilde{a}^{(n)} + \frac{\nu_2}{c}\sum_{k=1}^{K} w_k D_k^T D_k \tilde{a}^{(n)}$$

With Q denoting $\sum_k w_k D_k^T D_k$, the KKT conditions are:

$$\begin{aligned} 0 &= -b_n + \left(2\nu_1 I + \frac{\nu_2}{c} Q\right)\tilde{a}^{(n)} \\ 0 &= \nu_1 \left(\left(\tilde{a}^{(n)}\right)^T \left(\tilde{a}^{(n)}\right) - 1\right) \\ 0 &= \nu_2 \left(\frac{1}{c}\sum_{k=1}^{K} w_k \| D_k a^{(1)} \|_2^2 - 1\right) \end{aligned}$$

[1] If t were fixed, a more careful derivation would be required to find the optimal ν_2 for that t; however, given that t is a hyperparameter, it is tenable to view the solution given above for any ν_2 as optimal for some t and thus consider the choice of $\gamma = \nu_2/t$ as equivalent to the choice of t, suppressing altogether the need to find the optimal ν_2 explicitly.

[2] Furthermore, t must be well-specified such that $\|S(b_n, \gamma)\|_2 > 0$. As this is not always ensured in practice when tuning γ, the heuristic of reverting to the unconstrained update rule whenever $S(b_n, \gamma) = 0$ is adopted here. While the literature suggests setting $\tilde{a}^{(n)} = 0$ instead, this is ill-advised since after such an update $b_n = 0$ and therefore all factors are set to zero thereafter.

Rearranging the first equation yields:

$$\tilde{a}^{(n)} = \frac{\left(I + \left(\frac{\nu_2}{2c\nu_1}\right)Q\right)^{-1} b_n}{2\nu_1}$$

With $\gamma = \frac{\nu_2}{2c\nu_1}$ such that $\sum_{k=1}^{K} w_k \| D_k a^{(1)} \|_2^2 = c$, the KKT conditions are satisfied by:

$$\tilde{a}^{(n)} = \frac{(I + \gamma Q)^{-1} b_n}{\|(I + \gamma Q)^{-1} b_n\|_2}$$

Since any γ is optimal for some c, and c is a hyperparameter proportional to γ, the latter can be treated as a hyperparameter. This suppresses the need to find closed-form expressions for an optimal ν given c.