Multiresolution Tensor Learning for Efficient and Interpretable Spatial Analysis

Jung Yeon Park [1] **Kenneth Theo Carr** [1] **Stephan Zheng** [2] **Yisong Yue** [3] **Rose Yu** [1 4]

Abstract

Efficient and interpretable spatial analysis is crucial in many fields such as geology, sports, and climate science. Tensor latent factor models can describe higher-order correlations for spatial data. However, they are computationally expensive to train and are sensitive to initialization, leading to spatially incoherent, uninterpretable results. We develop a novel Multiresolution Tensor Learning (`MRTL`) algorithm for efficiently learning interpretable spatial patterns. `MRTL` initializes the latent factors from an approximate full-rank tensor model for improved interpretability and progressively learns from a coarse resolution to the fine resolution to reduce computation. We also prove the theoretical convergence and computational complexity of `MRTL`. When applied to two real-world datasets, `MRTL` demonstrates $4 \sim 5$x speedup compared to a fixed resolution approach while yielding accurate and interpretable latent factors.

1. Introduction

Analyzing large-scale spatial data plays a critical role in sports, geology, and climate science. In spatial statistics, kriging or Gaussian processes are popular tools for spatial analysis (Cressie, 1992). Others have proposed various Bayesian methods such as Cox processes (Miller et al., 2014; Dieng et al., 2017) to model spatial data. However, while mathematically appealing, these methods often have difficulties scaling to high-resolution data.

We are interested in learning high-dimensional tensor latent factor models, which have shown to be a scalable alternative for spatial analysis (Yu et al., 2018; Litvinenko et al., 2019). High resolution spatial data often contain higher-order correlations between features and locations, and tensors can naturally encode such multi-way correlations. For example, in competitive basketball play, we can predict how each player's decision to shoot is jointly influenced by their shooting style, his or her court position, and the position of the defenders by simultaneously encoding these features as a tensor. Using such representations, learning tensor latent factors can directly extract higher-order correlations.

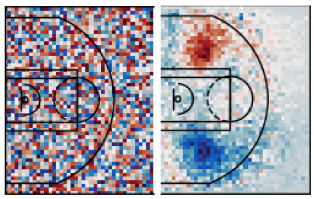

Figure 1. Latent factors: random (left) vs. good (right) initialization. Latent factors vary in interpretability depending on initialization.

A challenge in such models is high computational cost. High-resolution spatial data is often discretized, leading to large high-dimensional tensors whose training scales exponentially with the number of parameters. Low-rank tensor learning (Yu et al., 2018; Kossaifi et al., 2019) reduces the dimensionality by assuming low-rank structures in the data and uses tensor decomposition to discover latent semantics; for an overview of tensor learning, see review papers (Kolda & Bader, 2009; Sidiropoulos et al., 2017). However, many tensor learning methods have been shown to be sensitive to noise (Cheng et al., 2016) and initialization (Anandkumar et al., 2014). Other numerical techniques, including random sketching (Wang et al., 2015; Haupt et al., 2017) and parallelization, (Austin et al., 2016; Li et al., 2017a) can speed up training, but they often fail to utilize the unique properties of spatial data such as spatial auto-correlations.

Using latent factor models also gives rise to another issue: interpretability. It is well known that a latent factor model is generally not identifiable (Allman et al., 2009), leading to uninterpretable factors that do not offer insights to domain experts. In general, the definition of interpretability is highly

[1]Khoury College of Computer Sciences, Northeastern University, Boston, MA, USA. [2]Salesforce AI Research, Palo Alto, CA, USA. Work done while at California Institute of Technology. [3]Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA, USA. [4]Computer Science and Engineering, University of California, San Diego, San Diego, CA, USA.. Correspondence to: Jung Yeon Park, Rose Yu <park.jungy@northeastern.edu, roseyu@eng.ucsd.edu >.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

application dependent (Doshi-Velez & Kim, 2017). For spatial analysis, one of the unique properties of spatial patterns is *spatial auto-correlation*: close objects have similar values (Moran, 1950), which we use as a criterion for interpretability. As latent factor models are sensitive to initialization, previous research (Miller et al., 2014; Yue et al., 2014) has shown that randomly initialized latent factor models can lead to spatial patterns that violate spatial auto-correlation and hence are not interpretable (see Fig. 1).

In this paper, we propose a Multiresolution Tensor Learning algorithm, MRTL, to efficiently learn accurate and interpretable patterns in spatial data. MRTL is based on two key insights. First, to obtain good initialization, we train a full-rank tensor model approximately at a low resolution and use tensor decomposition to produce latent factors. Second, we exploit spatial auto-correlation to learn models at multiple resolutions: we train starting from a coarse resolution and iteratively finegrain to the next resolution. We provide theoretical analysis and prove the convergence properties and computational complexity of MRTL. We demonstrate on two real-world datasets that this approach is significantly faster than fixed resolution methods. We develop several finegraining criteria to determine when to finegrain. We also consider different interpolation schemes and discuss how to finegrain in different applications. The code for our implementation is available [1].

In summary, we:

- propose a Multiresolution Tensor Learning (MRTL) optimization algorithm for large-scale spatial analysis.
- prove the rate of convergence for MRTL which depends on the spectral norm of the interpolation operator. We also show the exponential computational speedup for MRTL compared with fixed resolution.
- develop different criteria to determine when to transition to a finer resolution and discuss different finegraining methods.
- evaluate on two real-world datasets and show MRTL learns faster than fixed-resolution learning and can produce interpretable latent factors.

2. Related Work.

Spatial Analysis Discovering spatial patterns has significant implications in scientific fields such as human behavior modeling, neural science, and climate science. Early work in spatial statistics has contributed greatly to spatial analysis through the work in Moran's I (Moran, 1950) and Getis-Ord general G (Getis & Ord, 1992) for measuring spatial auto-correlation. Geographically weighted regression (Brunsdon et al., 1998) accounts for the spatial heterogeneity with a local version of spatial regression but fails to capture higher order correlation. Kriging or Gaussian processes are popular tools for spatial analysis but they often require carefully designed variograms (also known as kernels) (Cressie, 1992). Other Bayesian hierarchical models favor spatial point processes to model spatial data (Diggle et al., 2013; Miller et al., 2014; Dieng et al., 2017). These frameworks are conceptually elegant but often computationally intractable.

Tensor Learning Latent factor models utilize correlations in the data to reduce the dimensionality of the problem, and have been used extensively in multi-task learning (Romera-Paredes et al., 2013) and recommendation systems (Lee & Seung, 2001). Tensor learning (Zhou et al., 2013; Bahadori et al., 2014; Haupt et al., 2017) uses tensor latent factor models to learn higher-order correlations in the data in a supervised fashion. In particular, tensor latent factor models aim to learn the higher-order correlations in spatial data by assuming low-dimensional representations among features and locations. However, high-order tensor models are non-convex by nature, suffer from the curse of dimensionality, and are notoriously hard to train (Kolda & Bader, 2009; Sidiropoulos et al., 2017). There are many efforts to scale up tensor computation, e.g., parallelization (Austin et al., 2016) and sketching (Wang et al., 2015; Haupt et al., 2017; Li et al., 2017b). In this work, we propose an optimization algorithm to learn tensor models at multiple resolutions that is not only fast but can also generate interpretable factors. We focus on tensor latent factor models for their wide applicability to spatial analysis and interpretability. While deep neural networks models can be more accurate, they are computationally more expensive and are difficult to interpret.

Multiresolution Methods Multiresolution methods have been applied successfully in machine learning, both in latent factor modeling (Kondor et al., 2014; Ozdemir et al., 2017) and deep learning (Reed et al., 2017; Serban et al., 2017). For example, multiresolution matrix factorization (Kondor et al., 2014; Ding et al., 2017) and its higher order extensions (Schifanella et al., 2014; Ozdemir et al., 2017; Han & Dunson, 2018) apply multi-level orthogonal operators to uncover the multiscale structure in a single matrix. In contrast, our method aims to speed up learning by exploiting the relationship among multiple tensors of different resolutions. Our approach resembles the multigrid method in numerical analysis for solving partial differential equations (Trottenberg et al., 2000; Hiptmair, 1998), where the idea is to accelerate iterative algorithms by solving a coarse problem first and then gradually finegraining the solution.

3. Tensor Models for Spatial Data

We consider tensor learning in the supervised setting. We describe both models for the full-rank case and the low-rank

[1] https://github.com/Rose-STL-Lab/mrtl

case. An order-3 tensor is used for ease of illustration but our model covers higher order cases.

3.1. Full Rank Tensor Models

Given input data consisting of both non-spatial and spatial features, we can discretize the spatial features at $r = 1, \ldots, R$ resolutions, with corresponding dimensions as $D_1, \ldots, D_R$. Tensor learning parameterizes the model with a weight tensor $\mathcal{W}^{(r)} \in \mathbb{R}^{I \times F \times D_r}$ over all features, where I is number of outputs and F is number of non-spatial features. The input data is of the form $\mathcal{X}^{(r)} \in \mathbb{R}^{I \times F \times D_r}$. Note that both the input features and the learning model are resolution dependent. $\mathcal{Y}_i \in \mathbb{R}, i = 1, \ldots, I$ is the label for output i.

At resolution r, the full rank tensor learning model can be written as

$$\mathcal{Y}_i = a\left(\sum_{f=1}^{F}\sum_{d=1}^{D_r} \mathcal{W}^{(r)}_{i,f,d}\mathcal{X}^{(r)}_{i,f,d} + b_i\right), \tag{1}$$

where a is the activation function and b_i is the bias for output i. The weight tensor $\mathcal{W}$ is contracted with $\mathcal{X}$ along the non-spatial mode f and the spatial mode d. In general, Eqn. (1) can be extended to multiple spatial features and spatial modes, each of which can have its own set of resolution-dependent dimensions. We use a sigmoid activation function for the classification task and the identity activation function for regression.

3.2. Low Rank Tensor Model

Low rank tensor models assume a low-dimensional latent structure in $\mathcal{W}$ which can characterize distinct patterns in the data and also alleviate model overfitting. To transform the learned tensor model to a low-rank one, we use CANDECOMP/PARAFAC (CP) decomposition (Hitchcock, 1927) on $\mathcal{W}$, which assumes that $\mathcal{W}$ can be represented as the sum of rank-1 tensors. Our method can easily be extended for other decompositions as well.

Let K be the CP rank of the tensor. In practice, K cannot be found analytically and is often chosen to sufficiently approximate the dataset. The weight tensor $\mathcal{W}^{(r)}$ is factorized into multiple factor matrices as

$$\mathcal{W}^{(r)}_{i,f,d} = \sum_{k=1}^{K} A_{i,k}B_{f,k}C^{(r)}_{d,k}$$

The tensor latent factor model is

$$\mathcal{Y}_i = a\left(\sum_{f=1}^{F}\sum_{d=1}^{D_r}\sum_{k=1}^{K} A_{i,k}B_{f,k}C^{(r)}_{d,k}\mathcal{X}^{(r)}_{i,f,d} + b_i\right), \tag{2}$$

where the columns of A, B, C^r are latent factors for each mode of $\mathcal{W}$ and $C^{(r)}$ is resolution dependent.

CP decomposition reduces dimensionality by assuming that A, B, C^r are uncorrelated, i.e. the features are uncorrelated. This is a reasonable assumption depending on how the features are chosen and leads to enhanced spatial interpretability as the learned spatial latent factors can show common patterns regardless of other features.

3.3. Spatial Regularization

Interpretability is in general hard to define or quantify (Doshi-Velez & Kim, 2017; Ribeiro et al., 2016; Lipton, 2018; Molnar, 2019). In the context of spatial analysis, we deem a latent factor as interpretable if it produces a spatially coherent pattern exhibiting spatial auto-correlation. To this end, we utilize a spatial regularization kernel (Lotte & Guan, 2010; Miller et al., 2014; Yue et al., 2014) and extend this to the tensor case.

Let $d = 1, \ldots, D_r$ index all locations of the spatial dimension for resolution r. The spatial regularization term is:

$$R_s = \sum_{d=1}^{D_r}\sum_{d'=1}^{D_r} K_{d,d'}\|\mathcal{W}_{:,:,d} - \mathcal{W}_{:,:,d'}\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ denotes the Frobenius norm and $K_{d,d'}$ is the kernel that controls the degree of similarity between locations. We use a simple RBF kernel with hyperparameter σ.

$$K_{d,d'} = e^{(-\|l_d - l_{d'}\|^2/\sigma)}, \tag{4}$$

where l_d denotes the location of index d. The distances are normalized across resolutions such that the maximum distance between two locations is 1. The kernels can be precomputed for each resolution. If there are multiple spatial modes, we apply spatial regularization across all different modes. We additionally use L_2 regularization to encourage smaller weights. The optimization objective function is

$$f(\mathcal{W}) = L(\mathcal{W}; \mathcal{X}, \mathcal{Y}) + \lambda_R R(\mathcal{W}), \tag{5}$$

where L is a task-dependent supervised learning loss, $R(\mathcal{W})$ is the sum of spatial and L_2 regularization, and λ_R is the regularization coefficient.

4. Multiresolution Tensor Learning

We now describe our algorithm `MRTL`, which addresses both the computation and interpretability issues. Two key concepts of `MRTL` are learning good initializations and utilizing multiple resolutions.

4.1. Initialization

In general, due to their nonconvex nature, tensor latent factor models are sensitive to initialization and can lead to uninterpretable latent factors (Miller et al., 2014; Yue

et al., 2014). We use full-rank initialization in order to learn latent factors that correspond to known spatial patterns.

We first train an approximate full-rank version of the tensor model at a low resolution in Eqn. (1). The weight tensor is then decomposed into latent factors and these values are used to initialize the low-rank model. The low-rank model in Eqn. (2) is then trained to the final desired accuracy. As we use approximately optimal solutions of the full-rank model as initializations for the low-rank model, our algorithm produces interpretable latent factors in a variety of different scenarios and datasets.

Full-rank initialization requires more computation than other simpler initialization methods. However, as the full-rank model is trained only for a small number of epochs, the increase in computation time is not substantial. We also train the full-rank model only at lower resolutions, for further reduction.

Previous research (Yue et al., 2014) showed that spatial regularization alone is not enough to learn spatially coherent factors, whereas full-rank initialization, though computationally costly, is able to fix this issue. We confirm the same holds true in our experiments (see Section 6.4). Thus, full-rank initialization is critical for spatial interpretability.

4.2. Multiresolution

Learning a high-dimensional tensor model is generally computationally expensive and memory inefficient. We utilize multiple resolutions for this issue. We outline the procedure of `MRTL` in Alg. 1, where we omit the bias term in the description for clarity.

We represent the resolution r with superscripts and the iterate at step t with subscripts, i.e. $\mathcal{W}_t^{(r)}$ is $\mathcal{W}$ at resolution r at step t. $\mathcal{W}_0$ is the initial weight tensor at the lowest resolution. $\mathcal{F}^{(r)} = (A, B, C^{(r)})$ denotes all factor matrices at resolution r and we use n to index the factor $\mathcal{F}^{(r),n}$.

For efficiency, we train both the full rank and low rank models at multiple resolutions, starting from a coarse spatial resolution and progressively increase the resolution. At each resolution r, we learn $\mathcal{W}^{(r)}$ using the stochastic optimization algorithm of choice `Opt` (we used Adam (Kingma & Ba, 2014) in our experiments). When the stopping criterion is met, we transform $\mathcal{W}^{(r)}$ to $\mathcal{W}^{(r+1)}$ in a process we call finegraining (`Finegrain`). Due to spatial auto-correlation, the trained parameters at a lower resolution will serve as a good initialization for higher resolutions. For both models, we only finegrain the factors that corresponds to resolution dependent mode, which is the spatial mode in the context of spatial analysis. Finegraining can be done for other non-spatial modes for more computational speedup as long as there exists a multiresolution structure (e.g. video or time series data).

Algorithm 1 Multiresolution Tensor Learning: `MRTL`

Input: initialization $\mathcal{W}_0$, data $\mathcal{X}, \mathcal{Y}$.
Output: latent factors $\mathcal{F}^{(r)}$
 # full rank tensor model
for each resolution $r \in \{1, \dots, r_0\}$ **do**
  Initialize $t \leftarrow 0$
  Get a mini-batch $\mathcal{B}$ from training set
  while stopping criterion not true **do**
    $t \leftarrow t + 1$
    $\mathcal{W}_{t+1}^{(r)} \leftarrow \texttt{Opt}\left(\mathcal{W}_t^{(r)} \mid \mathcal{B}\right)$
  end while
  $\mathcal{W}^{(r+1)} = \texttt{Finegrain}\left(\mathcal{W}^{(r)}\right)$
end for
 # tensor decomposition
$\mathcal{F}^{(r_0)} \leftarrow \texttt{CP\_ALS}\left(\mathcal{W}^{(r_0)}\right)$
 # low rank tensor model
for each resolution $r \in \{r_0, \dots, R\}$ **do**
  Initialize $t \leftarrow 0$
  Get a mini-batch $\mathcal{B}$ from training set
  while stopping criterion not true **do**
    $t \leftarrow t + 1$
    $\mathcal{F}_{t+1}^{(r)} \leftarrow \texttt{Opt}\left(\mathcal{F}_t^{(r)} \mid \mathcal{B}\right)$
  end while
  for each spatial factor $n \in \{1, \cdots, N\}$ **do**
    $\mathcal{F}^{(r+1),n} = \texttt{Finegrain}\left(\mathcal{F}^{(r),n}\right)$
  end for
end for

Once the full rank resolution has been trained up to resolution r_0 (which can be chosen to fit GPU memory or time constraints), we decompose $\mathcal{W}^{(r)}$ using `CP_ALS`, the standard alternating least squares (ALS) algorithm (Kolda & Bader, 2009) for CP decomposition. Then the low-rank model is trained at resolutions $r_0, \dots, R$ to final desired accuracy, finegraining to move to the next resolution.

When to finegrain There is a tradeoff between training times at different resolutions. While training for longer at lower resolutions significantly decreases computation, we do not want to overfit to the coarse, lower resolution data. On the other hand, training at higher resolutions can yield more accurate solutions using more detailed information. We investigate four different criteria to balance this tradeoff: 1) validation loss, 2) gradient norm, 3) gradient variance, and 4) gradient entropy.

Increase in validation loss (Prechelt, 1998; Yao et al., 2007) is a commonly used heuristic for early stopping. Another approach is to analyze the gradient distributions during training. For a convex function, stochastic gradient descent will converge into a noise ball near the optimal solution as the gradients approach zero. However, lower resolutions may be too coarse to learn more finegrained curvatures and the gradients will increasingly disagree near the optimal solution. We quantify the disagreement in the gradients with

metrics such as norm, variance, and entropy. We use intuition from convergence analysis for gradient norm and variance (Bottou et al., 2018), and information theory for gradient entropy (Srinivas et al., 2012).

Let $\mathcal{W}_t$ and ξ_t represent the weight tensor and the random variable for sampling of minibatches at step t, respectively. Let $f(\mathcal{W}_t; \xi_t) := f_t$ be the validation loss and $g(\mathcal{W}_t; \xi_t) := g_t$ be the stochastic gradients at step t. The finegraining criteria are:

- Validation Loss: $\mathbb{E}[f_{t+1}] - \mathbb{E}[f_t] > 0$
- Gradient Norm: $\mathbb{E}[\|g_{t+1}\|^2] - \mathbb{E}[\|g_t\|^2] > 0$
- Gradient Variance: $V(\mathbb{E}[g_{t+1}]) - V(\mathbb{E}[g_t]) > 0$
- Gradient Entropy: $S(\mathbb{E}[g_{t+1}]) - S(\mathbb{E}[g_t]) > 0$,

where $S(p) = \sum_i -p_i \ln(p_i)$. One can also use thresholds, e.g. $|f_{t+1} - f_t| < \tau$, but as these are dependent on the dataset, we use $\tau = 0$ in our experiments. One can also incorporate patience, i.e. setting the maximum number of epochs where the stopping conditions was reached.

How to finegrain We discuss different interpolation schemes for different types of features. Categorical/multinomial variables, such as a player's position on the court, are one-hot encoded or multi-hot encoded onto a discretized grid. Note that as we use higher resolutions, the sum of the input values are still equal across resolutions, $\sum_d \mathcal{X}^{(r)}_{:,:,d} = \sum_d \mathcal{X}^{(r+1)}_{:,:,d}$. As the sum of the features remains the same across resolutions and our tensor models are multilinear, nearest neighbor interpolation should be used in order to produce the same outputs.

$$\sum_{d=1}^{D_r} \mathcal{W}^{(r)}_{:,:,d} \mathcal{X}^{(r)}_{:,:,d} = \sum_{d=1}^{D_{r+1}} \mathcal{W}^{(r+1)}_{:,:,d} \mathcal{X}^{(r+1)}_{:,:,d}$$

as $\mathcal{X}^{(r)}_{i,f,d} = 0$ for cells that do not contain the value. This scheme yields the same outputs and thus the same loss values across resolutions.

Continuous variables that represent averages over locations, such as sea surface salinity, often have similar values at each finegrained cell at higher resolutions (as the values at coarse resolutions are subsampled or averaged from values at the higher resolution). Then $\sum_d^{D_{r+1}} \mathcal{X}^{(r+1)}_{:,:,d} \approx 2^2 \sum_d^{D_r} \mathcal{X}^{(r)}_{:,:,d}$, where the approximation comes from the type of downsampling used.

$$\sum_{d=1}^{D_r} \mathcal{W}^{(r)}_{:,:,d} \mathcal{X}^{(r)}_{:,:,d} \approx 2^2 \sum_{d=1}^{D_{r+1}} \mathcal{W}^{(r+1)}_{:,:,d} \mathcal{X}^{(r+1)}_{:,:,d}$$

using a linear interpolation scheme. The weights are divided by the scale factor of $\frac{D_{r+1}}{D_r}$ to keep the outputs approximately equal. We use bilinear interpolation, though any other linear interpolation can be used.

5. Theoretical Analysis.

5.1. Convergence

We prove the convergence rate for `MRTL` with a single spatial mode and one-dimensional output, where the weight tensor reduces to a weight vector $\mathbf{w}$. We defer all proofs to Appendix A. For the loss function f and a stochastic sampling variable ξ, the optimization problem is:

$$\mathbf{w}_\star = \text{argmin}\, \mathbb{E}[f(\mathbf{w}; \xi)] \tag{6}$$

We consider a fixed-resolution model that follows Alg. 1 with $r = \{R\}$, i.e. only the final resolution is used. For a fixed-resolution miniSGD algorithm, under common assumptions in convergence analysis:

- f is μ- strongly convex, L-smooth
- (unbiased) gradient $\mathbb{E}[g(\mathbf{w}_t; \xi_t)] = \nabla f(\mathbf{w}_t)$ given $\xi_{<t}$
- (variance) for all the $\mathbf{w}$, $\mathbb{E}[\|g(\mathbf{w}; \xi)\|_2^2] \leq \sigma_g^2 + c_g \|\nabla f(\mathbf{w})\|_2^2$

Theorem 5.1. *(Bottou et al., 2018) If the step size $\eta_t \equiv \eta \leq \frac{1}{Lc_g}$, then a fixed resolution solution satisfies*

$$\mathbb{E}[\|\mathbf{w}_{t+1} - \mathbf{w}_\star\|_2^2] \leq \gamma^t (\mathbb{E}[\|\mathbf{w}_0 - \mathbf{w}_\star\|_2^2) - \beta] + \beta,$$

where $\gamma = 1 - 2\eta\mu$, $\beta = \frac{\eta \sigma_g^2}{2\mu}$, and $w_\star$ is the optimal solution.

which gives $O(1/t) + O(\eta)$ convergence.

At resolution r, we define the number of total iterations as t_r, and the weights as $\mathbf{w}^{(r)}$. We let D_r denote the number of dimensions at r and we assume a dyadic scaling between resolutions such that $D_{r+1} = 2D_r$. We define finegraining using an interpolation operator P such that $\mathbf{w}_0^{(r+1)} = P\mathbf{w}_{t_r}^{(r)}$ as in (Bramble, 2019). For the simple case of a 1D spatial grid where $\mathbf{w}_t^{(r)}$ has spatial dimension D_r, P would be of a Toeplitz matrix of dimension $2D_r \times D_r$. For example, for linear interpolation of $D_r = 2$,

$$P\mathbf{w}^{(r)} = \frac{1}{2} \begin{bmatrix} 1 & 0 \\ 2 & 0 \\ 1 & 1 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} \mathbf{w}_1^{(r)} \\ \mathbf{w}_2^{(r)} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_1^{(r+1)}/2 \\ \mathbf{w}_1^{(r+1)} \\ \mathbf{w}_1^{(r+1)}/2 + \mathbf{w}_2^{(r+1)}/2 \\ \mathbf{w}_2^{(r+1)} \end{bmatrix}.$$

Any interpolation scheme can be expressed in this form.

The convergence of multiresolution learning algorithm depends on the following property of spatial data:

Definition 5.2 (Spatial Smoothness)**.** *The difference between the optimal solutions of consecutive resolutions is upper bounded by ϵ*

$$\|\mathbf{w}_\star^{(r+1)} - P\mathbf{w}_\star^{(r)}\| \leq \epsilon,$$

with P being the interpolation operator.

The following theorem proves the convergence rate of MRTL, with a constant that depends on the operator norm of the interpolation operator P.

Theorem 5.3. *If the step size $\eta_t \equiv \eta \leq \frac{1}{Lc_g}$, then the solution of MRTL satisfies*

$$\mathbb{E}[\|\mathbf{w}_t^{(r)} - \mathbf{w}_\star\|_2^2] \leq \gamma^t \|P\|_{op}^{2r} \mathbb{E}[\|\mathbf{w}_0 - \mathbf{w}_\star\|_2^2 + O(\eta\|P\|_{op}),$$

where $\gamma = 1 - 2\eta\mu$, $\beta = \frac{\eta\sigma_g^2}{2\mu}$, and $\|P\|_{op}$ is the operator norm of the interpolation operator P.

5.2. Computational Complexity

To analyze computational complexity, we resort to fixed point convergence (Hale et al., 2008) and the multigrid method (Stüben, 2001). Intuitively, as most of the training iterations are spent on coarser resolutions with fewer number of parameters, multiresolution learning is more efficient than fixed-resolution training.

Assuming that ∇f is Lipschitz continuous, we can view gradient-based optimization as a fixed-point iteration operator F with a contraction constant of $\gamma \in (0, 1)$ (note that *stochastic* gradient descent converges to a noise ball instead of a fixed point):

$$\mathbf{w} \leftarrow F(\mathbf{w}), \quad F := I - \eta\nabla f,$$
$$\|F(\mathbf{w}) - F(\mathbf{w}')\| \leq \gamma\|\mathbf{w} - \mathbf{w}'\|$$

Let $\mathbf{w}_\star^{(r)}$ be the optimal estimator at resolution r and $\mathbf{w}^{(r)}$ be a solution satisfying $\|\mathbf{w}_\star^{(r)} - \mathbf{w}^{(r)}\| \leq \epsilon/2$. The algorithm terminates when the estimation error reaches $\frac{C_0 R}{(1-\gamma)^2}$. The following lemma describes the computational cost of the *fixed-resolution* algorithm.

Lemma 5.4. *Given a fixed point iteration operator F with contraction constant of $\gamma \in (0, 1)$, the computational complexity of fixed-resolution training for tensor model of order p and rank K is*

$$\mathcal{C} = \mathcal{O}\left(\frac{1}{|\log\gamma|} \cdot \log\left(\frac{1}{(1-\gamma)\epsilon}\right) \cdot \frac{Kp}{(1-\gamma)^2\epsilon}\right), \quad (7)$$

where ϵ is the terminal estimation error.

The next Theorem 5.5 characterizes the computational speed-up gained by MRTL compared to fixed-resolution learning, with respect to the contraction factor γ and the terminal estimation error ϵ.

Theorem 5.5. *If the fixed point iteration operator (gradient descent) has a contraction factor of γ, multiresolution learning with the termination criteria of $\frac{C_0 r}{(1-\gamma)^2}$ at resolution r is faster than fixed-resolution learning by a factor of $\log\frac{1}{(1-\gamma)\epsilon}$, with the terminal estimation error ϵ.*

Note that the speed-up using multiresolution learning uses a global convergence criterion ϵ for each r.

6. Experiments

We apply MRTL to two real-world datasets: basketball tracking and climate data. More details about the datasets and pre-processing steps are provided in Appendix B.

6.1. Datasets

Tensor classification: Basketball tracking We use a large NBA player tracking dataset from (Yue et al., 2014; Zheng et al., 2016) consisting of the coordinates of all players at 25 frames per second, for a total of approximately 6 million frames. The goal is to predict whether a given ball handler will shoot within the next second, given his position on the court and the relative positions of the defenders around him. In applying our method, we hope to obtain common shooting locations on the court and how a defender's relative position suppresses shot probability.

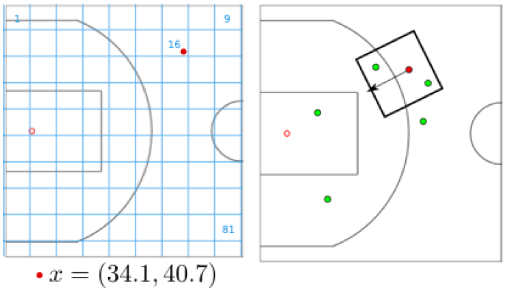

Figure 2. Left: Discretizing a continuous-valued position of a player (red) via a spatial grid. Right: sample frame with a ballhandler (red) and defenders (green). Only defenders close to the ballhandler are used.

The basketball data contains two spatial modes: the ball handler's position and the relative defender positions around the ball handler. We instantiate a tensor classification model in Eqn (1) as follows:

$$\mathcal{Y}_i = \sum_{d^1=1}^{D_r^1} \sum_{d^2=1}^{D_r^2} \sigma(\mathcal{W}_{i,d^1,d^2}^{(r)} \mathcal{X}_{i,d^1,d^2}^{(r)} + b_i) \,,$$

where $i \in \{1, \ldots, I\}$ is the ballhandler ID, d^1 indexes the ballhandler's position on the discretized court of dimension $\{D_r^1\}$, and d^2 indexes the relative defender positions around the ballhandler in a discretized grid of dimension $\{D_r^2\}$. We assume that only defenders close to the ballhandler affect shooting probability and set $D_r^2 < D_r^1$ to reduce dimensionality. As shown in Fig. 2, we orient the defender positions so that the direction from the ballhandler to the basket points up. $\mathcal{Y}_i \in \{0, 1\}$ is the binary output equal to 1 if player i shoots within the next second and σ is the sigmoid function.

We use nearest neighbor interpolation for finegraining and a

weighted cross entropy loss (due to imbalanced classes):

$$\mathcal{L}_n = -\beta \left[\mathcal{Y}_n \cdot \log \hat{\mathcal{Y}}_n + (1 - \mathcal{Y}_n) \cdot \log (1 - \hat{\mathcal{Y}}_n) \right], \quad (8)$$

where n denotes the sample index and β is the weight of the positive samples and set equal to the ratio of the negative and positive counts of labels.

Tensor regression: Climate Recent research (Li et al., 2016a;b; Zeng et al., 2019) shows that oceanic variables such as sea surface salinity (SSS) and sea surface temperature (SST) are significant predictors of the variability in rainfall in land-locked locations, such as the U.S. Midwest. We aim to predict the variability in average monthly precipitation in the U.S. Midwest using SSS and SST to identify meaningful latent factors underlying the large-scale processes linking the ocean and precipitation on land (Fig. 3). We use precipitation data from the PRISM group (PRISM Climate Group, 2013) and SSS/SST data from the EN4 reanalysis (Good et al., 2013).

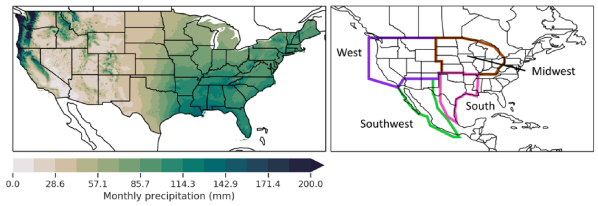

Figure 3. Left: precipitation over continental U.S. Right: regions considered in particular.

Let $\mathcal{X}$ be the historical oceanic data with spatial features SSS and SST across D_r locations, using the previous 6 months of data. As SSS and SST share the spatial mode (the same spatial locations), we set the $F_2 = 2$ to denote the index of these features. We also consider the lag as a non-spatial feature so that $F_1 = 6$. We instantiate the tensor regression model in Eqn (1) as follows:

$$\mathcal{Y} = \sum_{f_1=1}^{F_1} \sum_{f_2=1}^{F_2} \sum_{d=1}^{D_r} \mathcal{W}^{(r)}_{f_1,f_2,d} \mathcal{X}^{(r)}_{f_1,f_2,d} + b$$

The features and outputs (SSS, SST, and precipitation) are subject to long-term trends and a seasonal cycle. We use difference detrending for each timestep due to non-stationarity of the inputs, and remove seasonality in the data by standardizing each month of the year. The features are normalized using min-max normalization. We also normalize and deseasonalize the outputs, so that the model predicts standardized anomalies. We use mean square error (MSE) for the loss function and bilinear interpolation for finegraining.

Implementation Details For both datasets, we discretize the spatial features and use a 60-20-20 train-validation-test set split. We use Adam (Kingma & Ba, 2014) for optimization as it was empirically faster than SGD in our experiments. We use both L_2 and spatial regularization as described in Section 3. We selected optimal hyperparameters for all models via random search. We use a stepwise learning rate decay with stepsize of 1 with $\gamma = 0.95$. We perform ten trials for all experiments. All other details are provided in Appendix B.

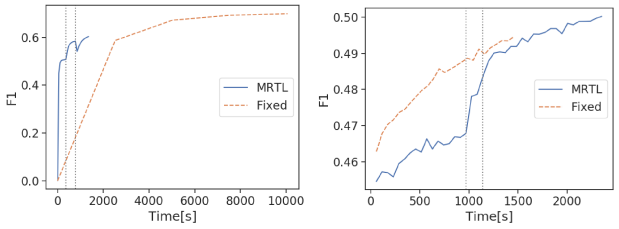

Figure 4. Basketball: F1 scores of MRTL vs. the fixed-resolution model for the full rank (left) and low rank model (right). The vertical lines indicate finegraining to the next resolution.

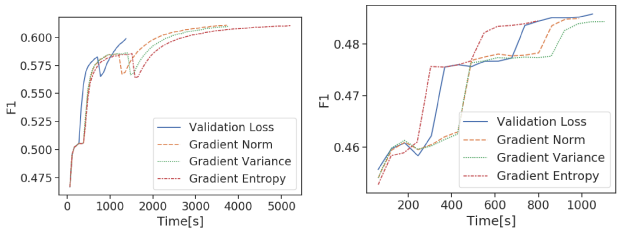

Figure 5. Basketball: F1 scores different finegraining criteria for the full rank (left) and low rank (right) model

6.2. Accuracy and Convergence

We compare MRTL against a fixed-resolution model on accuracy and computation time. We exclude the computation time for CP_ALS as it was quick to compute for all experiments (< 5 seconds for the basketball dataset). The results of all trials are listed in Table 1. Some results are provided in Appendix B.

Fig. 4 shows the F1 scores of MRTL vs a fixed resolution model for the basketball dataset (validation loss was used as the finegraining criterion for both models). For the full rank case, MRTL converges 9 times faster than the fixed resolution case (the scaling of the axes obscures convergence; nevertheless, both algorithms have converged). The fixed-resolution model is able to reach a higher F1 score for the full rank case, as it uses a higher resolution than MRTL and is able to use more finegrained information, translating to a higher quality solution. This advantage does not transfer to the low rank model.

For the low rank model, the training times are comparable and both reach a similar F1 score. There is decrease in the

Table 1. Runtime and prediction performance comparison of a fixed-resolution model vs MRTL for datasets

Dataset	Model	Full Rank			Low Rank		
		Time [s]	Loss	F1	Time [s]	Loss	F1
Basketball	Fixed	11462 ±565	0.608 ±0.00941	0.685 ±0.00544	2205 ±841	0.849 ±0.0230	0.494 ±0.00417
	MRTL	1230 ±74.1	0.699 ±0.00237	0.607 ±0.00182	2009 ±715	0.868 ±0.0399	0.475 ±0.0121
Climate	Fixed	12.5±0.0112	0.0882 ±0.0844	-	269 ±319	0.0803 ±0.0861	-
	MRTL	1.11 ±0.180	0.0825 ±0.0856	-	67.1 ±31.8	0.0409 ±0.00399	-

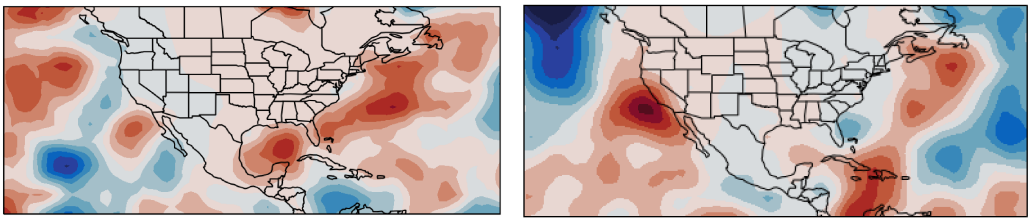

Figure 6. Climate: Some latent factors of sea surface locations after training. The red areas in the northwest Atlantic region (east of North America and Gulf of Mexico) represent areas where moisture export contributes to precipitation in the U.S. Midwest.

F1 score going from full rank to low rank for both MRTL and the fixed resolution model due to approximation error from CP decomposition. Note that this is dependent on the choice of K, specific to each dataset. Furthermore, we see a smaller increase in performance for the low rank model vs. the full rank case, indicating that the information gain from finegraining does not scale linearly with the resolution. We see a similar trend for the climate data, where MRTL converges faster than the fixed-resolution model. Overall, MRTL is approximately $4 \sim 5$ times faster and we get a similar speedup in the climate data.

6.3. Finegraining Criteria

We compare the performance of different finegraining criteria in Fig. 5. Validation loss converges much faster than other criteria for the full rank model while the other finegraining criteria converge slightly faster for the low rank model. In the classification case, we observe that the full rank model spends many epochs training when we use gradient-based criteria, suggesting that they can be too strict for the full rank case. For the regression case, we see all criteria perform similarly for the full rank model, and validation loss converges faster for the low rank model. As there are differences between finegraining criteria for different datasets, one should try all of them for fastest convergence.

6.4. Interpretability

We now demonstrate that MRTL can learn semantic representations along spatial dimensions. For all latent factor figures, the factors have been normalized to $(-1, 1)$ so that reds are positive and blues are negative.

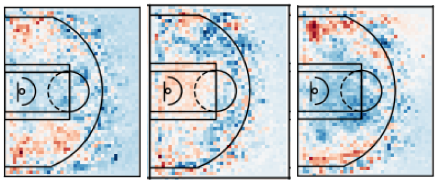

Figure 7. Basketball: Latent factor heatmaps of ballhandler position after training for $k = 1, 3, 20$. They represent common shooting locations such as the right/left sides of the court, the paint, or near the three point line.

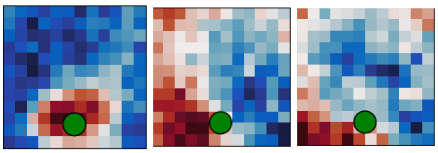

Figure 8. Basketball: Latent factor heatmaps of relative defender positions after training for $k = 1, 3, 20$. The green dot represents the ballhandler at $(6, 2)$. The latent factors show spatial patterns near the ballhandler, suggesting important positions to suppress shot probability.

Figs. 7, 8 visualize some latent factors for ballhandler position and relative defender positions, respectively (see Appendix for all latent factors). For the ballhandler position in Fig. 7, coherent spatial patterns (can be both red or blue regions as they are simply inverses of each other) can correspond to common shooting locations. These latent factors can represent known locations such as the paint or near the three-point line on both sides of the court.

For relative defender positions in Fig. 8, we see many concentrated spatial regions near the ballhandler, indicating that

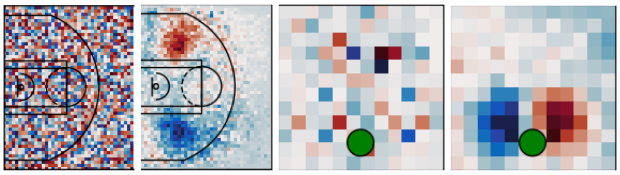

Figure 9. Latent factor comparisons ($k = 3, 10$) of randomly initialized low-rank model (1st and 3rd) and MRTL (2nd and 4th) for ballhandler position (left two plots) and the defender positions (right two plots). Random initialization leads to uninterpretable latent factors.

such close positions suppress shot probability (as expected). Some latent factors exhibit directionality as well, suggesting that guarding one side of the ballhandler may suppress shot probability more than the other side.

Fig. 6 depicts two latent factors of sea surface locations. We would expect latent factors to correspond to regions of the ocean which independently influence precipitation. The left latent factor highlights the Gulf of Mexico and northwest Atlantic ocean as influential for rainfall in the Midwest due to moisture export from these regions. This is consistent with findings from (Li et al., 2018; 2016a).

Random initialization We also perform experiments using a randomly initialized low-rank model (without the full-rank model) in order to verify the importance of full rank initialization. Fig. 9 compares random initialization vs. MRTL for the ballhandler position (left two plots) and the defender positions (right two plots). We observe that even with spatial regularization, randomly initialized latent factor models can produce noisy, uninterpretable factors and thus full-rank initialization is essential for interpretability.

7. Conclusion and Future Work

We presented a novel algorithm for tensor models for spatial analysis. Our algorithm MRTL utilizes multiple resolutions to significantly decrease training time and incorporates a full-rank initialization strategy that promotes spatially coherent and interpretable latent factors. MRTL is generalized to both the classification and regression cases. We proved the theoretical convergence of our algorithm for stochastic gradient descent and compared the computational complexity of MRTL to a single, fixed-resolution model. The experimental results on two real-world datasets support its improvements in computational efficiency and interpretability.

Future work includes 1) developing other stopping criteria in order to enhance the computational speedup, 2) applying our algorithm to more higher-dimensional spatiotemporal data, and 3) studying the effect of varying batch sizes between resolutions as in (Wu et al., 2019).

Acknowledgements

This work was supported in part by NSF grants #1850349, #1564330, and #1663704, Google Faculty Research Award and Adobe Data Science Research Award. We thank Stats Perform SportVU [2] for the basketball tracking data. We gratefully acknowledge use of the following datasets: PRISM by the PRISM Climate Group, Oregon State University [3] and EN4 by the Met Office Hadley Centre [4], and thank Caroline Ummenhofer of Woods Hole Oceanographic Institution for helping us obtain the data.

References

Allman, E. S., Matias, C., Rhodes, J. A., et al. Identifiability of parameters in latent structure models with many observed variables. *The Annals of Statistics*, 37(6A): 3099–3132, 2009.

Anandkumar, A., Ge, R., and Janzamin, M. Guaranteed non-orthogonal tensor decomposition via alternating rank-1 updates. *arXiv preprint arXiv:1402.5180*, 2014.

Austin, W., Ballard, G., and Kolda, T. G. Parallel tensor compression for large-scale scientific data. In *Parallel and Distributed Processing Symposium, 2016 IEEE International*, pp. 912–922. IEEE, 2016.

Bahadori, M. T., Yu, Q. R., and Liu, Y. Fast multivariate spatio-temporal analysis via low rank tensor learning. In *Advances in neural information processing systems*, pp. 3491–3499, 2014.

Bottou, L., Curtis, F. E., and Nocedal, J. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

Bramble, J. H. *Multigrid methods*. Routledge, 2019.

Brunsdon, C., Fotheringham, S., and Charlton, M. Geographically weighted regression. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 47(3): 431–443, 1998.

Cheng, H., Yu, Y., Zhang, X., Xing, E., and Schuurmans, D. Scalable and sound low-rank tensor learning. In *Artificial Intelligence and Statistics*, pp. 1114–1123, 2016.

Cressie, N. Statistics for spatial data. *Terra Nova*, 4(5): 613–617, 1992.

Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., and Blei, D. Variational inference via χ upper bound minimization.

[2] https://www.statsperform.com/team-performance/basketball/optical-tracking/

[3] http://prism.oregonstate.edu

[4] https://www.metoffice.gov.uk/hadobs/en4/

In *Advances in Neural Information Processing Systems*, pp. 2732–2741, 2017.

Diggle, P. J., Moraga, P., Rowlingson, B., and Taylor, B. M. Spatial and spatio-temporal log-gaussian cox processes: extending the geostatistical paradigm. *Statistical Science*, pp. 542–563, 2013.

Ding, Y., Kondor, R., and Eskreis-Winkler, J. Multiresolution kernel approximation for gaussian process regression. In *Advances in Neural Information Processing Systems*, pp. 3743–3751, 2017.

Doshi-Velez, F. and Kim, B. Towards a rigorous science of interpretable machine learning. *arXiv preprint arXiv:1702.08608*, 2017.

Getis, A. and Ord, J. K. The analysis of spatial association by use of distance statistics. *Geographical analysis*, 24 (3):189–206, 1992.

Good, S., Martin, M., and Rayner, N. En4: quality controlled ocean temperature and salinity profiles and monthly objective analyses with uncertainty estimates. *Journal of Geophysical Research: Oceans*, 118:6704–6716, 2013. Version EN4.2.1, https://www.metoffice.gov.uk/hadobs/en4/download-en4-2-1.html, accessed 06/23/19.

Hale, E. T., Yin, W., and Zhang, Y. Fixed-point continuation for ℓ_1-minimization: Methodology and convergence. *SIAM Journal on Optimization*, 19(3):1107–1130, 2008.

Han, S. and Dunson, D. B. Multiresolution tensor decomposition for multiple spatial passing networks. *arXiv preprint arXiv:1803.01203*, 2018.

Haupt, J., Li, X., and Woodruff, D. P. Near optimal sketching of low-rank tensor regression. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 3469–3479. Curran Associates Inc., 2017.

Hiptmair, R. Multigrid method for maxwell's equations. *SIAM Journal on Numerical Analysis*, 36(1):204–225, 1998.

Hitchcock, F. L. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Kolda, T. G. and Bader, B. W. Tensor Decompositions and Applications. *SIAM Review*, 51(3):455–500, 2009. ISSN 0036-1445. doi: 10.1137/07070111X.

Kondor, R., Teneva, N., and Garg, V. Multiresolution matrix factorization. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1620–1628, 2014.

Kossaifi, J., Panagakis, Y., Anandkumar, A., and Pantic, M. Tensorly: Tensor learning in python. *The Journal of Machine Learning Research*, 20(1):925–930, 2019.

Lee, D. D. and Seung, H. S. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pp. 556–562, 2001.

Li, J., Choi, J., Perros, I., Sun, J., and Vuduc, R. Model-driven sparse cp decomposition for higher-order tensors. In *2017 IEEE international parallel and distributed processing symposium (IPDPS)*, pp. 1048–1057. IEEE, 2017a.

Li, L., Schmitt, R., Ummenhofer, C., and Karnauskas, K. Implications of north atlantic sea surface salinity for summer precipitation over the us midwest: Mechanisms and predictive value. *Journal of Climate*, 29:3143–3159, 2016a.

Li, L., Schmitt, R., Ummenhofer, C., and Karnauskas, K. North atlantic salinity as a predictor of sahel rainfall. *Science Advances*, 29:3143–3159, 2016b.

Li, L., Schmitt, R. W., and Ummenhofer, C. C. The role of the subtropical north atlantic water cycle in recent us extreme precipitation events. *Climate dynamics*, 50(3-4): 1291–1305, 2018.

Li, X., Haupt, J., and Woodruff, D. Near optimal sketching of low-rank tensor regression. In *Advances in Neural Information Processing Systems 30*, pp. 3466–3476, 2017b.

Lipton, Z. C. The mythos of model interpretability. *Queue*, 16(3):31–57, 2018.

Litvinenko, A., Keyes, D., Khoromskaia, V., Khoromskij, B. N., and Matthies, H. G. Tucker tensor analysis of matérn functions in spatial statistics. *Computational Methods in Applied Mathematics*, 19(1):101–122, 2019.

Lotte, F. and Guan, C. Regularizing common spatial patterns to improve bci designs: unified theory and new algorithms. *IEEE Transactions on biomedical Engineering*, 58(2):355–362, 2010.

Miller, A., Bornn, L., Adams, R., and Goldsberry, K. Factorized point process intensities: A spatial analysis of professional basketball. In *International Conference on Machine Learning (ICML)*, 2014.

Molnar, C. *Interpretable Machine Learning*. 2019. https://christophm.github.io/interpretable-ml-book/.

Moran, P. A. Notes on continuous stochastic phenomena. *Biometrika*, pp. 17–23, 1950.

Nash, S. G. A multigrid approach to discretized optimization problems. *Optimization Methods and Software*, 14(1-2): 99–116, 2000.

Ozdemir, A., Iwen, M. A., and Aviyente, S. Multiscale analysis for higher-order tensors. *arXiv preprint arXiv:1704.08578*, 2017.

Prechelt, L. Automatic early stopping using cross validation: quantifying the criteria. *Neural Networks*, 11(4):761–767, 1998.

PRISM Climate Group. Gridded climate data for the contiguous usa., 2013. http://prism.oregonstate.edu, accessed 07/08/19.

Reed, S., Oord, A., Kalchbrenner, N., Colmenarejo, S. G., Wang, Z., Chen, Y., Belov, D., and Freitas, N. Parallel multiscale autoregressive density estimation. In *International Conference on Machine Learning*, pp. 2912–2921, 2017.

Ribeiro, M. T., Singh, S., and Guestrin, C. Why should i trust you?: Explaining the predictions of any classifier. pp. 1135–1144, 2016.

Romera-Paredes, B., Aung, H., Bianchi-Berthouze, N., and Pontil, M. Multilinear multitask learning. In *International Conference on Machine Learning*, pp. 1444–1452, 2013.

Schifanella, C., Candan, K. S., and Sapino, M. L. Multiresolution tensor decompositions with mode hierarchies. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 8(2):10, 2014.

Serban, I. V., Klinger, T., Tesauro, G., Talamadupula, K., Zhou, B., Bengio, Y., and Courville, A. C. Multiresolution recurrent neural networks: An application to dialogue response generation. In *AAAI*, pp. 3288–3294, 2017.

Sidiropoulos, N. D., De Lathauwer, L., Fu, X., Huang, K., Papalexakis, E. E., and Faloutsos, C. Tensor decomposition for signal processing and machine learning. *IEEE Transactions on Signal Processing*, 65(13):3551–3582, 2017.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. W. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

Stüben, K. A review of algebraic multigrid. In *Partial Differential Equations*, pp. 281–309. Elsevier, 2001.

Trottenberg, U., Oosterlee, C. W., and Schuller, A. *Multigrid*. Elsevier, 2000.

Wang, Y., Tung, H.-Y., Smola, A. J., and Anandkumar, A. Fast and guaranteed tensor decomposition via sketching. In *Advances in Neural Information Processing Systems*, pp. 991–999, 2015.

Wu, C.-Y., Girshick, R., He, K., Feichtenhofer, C., and Krähenbühl, P. A multigrid method for efficiently training video models, 2019.

Yao, Y., Rosasco, L., and Caponnetto, A. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315, 2007.

Yu, R., Li, G., and Liu, Y. Tensor regression meets gaussian processes. In *21st International Conference on Artificial Intelligence and Statistics (AISTATS 2018)*, 2018.

Yue, Y., Lucey, P., Carr, P., Bialkowski, A., and Matthews, I. Learning Fine-Grained Spatial Models for Dynamic Sports Play Prediction. In *IEEE International Conference on Data Mining (ICDM)*, 2014.

Zeng, L., Schmitt, R., Li, L., Wang, Q., and Wang, D. Forecast of summer precipitation in the yangtze river valley based on south china sea springtime sea surface salinity. *Climate Dynamics*, 53:5495–5509, 2019.

Zheng, S., Yue, Y., and Hobbs, J. Generating long-term trajectories using deep hierarchical networks. In *Advances in Neural Information Processing Systems*, pp. 1543–1551, 2016.

Zhou, H., Li, L., and Zhu, H. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.