Large-Scale Spatiotemporal Density Smoothing with the Graph-fused Elastic Net: Application to Ride-sourcing Driver Productivity Analysis*

Mauricio Tec
Department of Statistics and Data Science
University of Texas at Austin

Natalia Zuniga-Garcia
Department of Civil, Architectural and Environmental Engineering
University of Texas at Austin

Randy B. Machemehl
Department of Civil, Architectural and Environmental Engineering
University of Texas at Austin

James G. Scott
Department of Information, Risk, and Operations Management
Department of Statistics and Data Science
University of Texas at Austin

November 20, 2019

Abstract

Ride-sourcing or transportation network companies (TNCs) provide on-demand transportation service for compensation, connecting drivers of personal vehicles with passengers through the use of smartphone applications. This article considers the problem of estimating the probability distribution of the productivity of a driver as a function of space and time. We study data consisting of more than 1 million ride-sourcing trips in Austin, Texas, which are scattered throughout a large graph of 223k vertices, where each vertex represents a traffic analysis zone (TAZ) at a specific hour of the week. We extend existing methods for spatial density smoothing on very large general graphs to the spatiotemporal setting. Our proposed model allows for distinct spatial and temporal dynamics, including different degrees of smoothness, and it appropriately handles vertices with missing data, which in our case arise from a fine discretization over the time dimension. Core to our method is an extension of the Graph-Fused Lasso that we refer to as the Graph-fused Elastic Net (GFEN).

Keywords: spatiotemporal modeling, Graph-fused Lasso, Markov Random Fields, ADMM algorithm, density smoothing, non-parametric density estimation, big data, transportation network companies, ride-sourcing

*Preprint

1 Spatiotemporal variation in TNC driver earnings

1.1 Introduction

Ride-sourcing or transportation network companies (TNCs), such as Uber and Lyft, provide on-demand transportation service for compensation (Shaheen et al., 2016). TNCs operate as a two-sided market that connects drivers with passengers through the use of mobile applications. In recent years, TNCs have experienced rapid growth; for example, Uber saw 5.22 billion trips worldwide in 2018, up from 140 million trips in 2014. This growth has posed several challenges to transportation planners, policymakers, and researchers—for example, lack of infrastructure (e.g., at airports), geographical variation in operating rules and regulations, and potential changes in travelers' behavior (Smith, 2019). TNCs have also been the subject of controversy because of their aggressive business tactics and sometimes complex pricing systems (Li et al., 2019), whose effects on both rider and driver behavior are not well understood.

In this paper, we address a fundamental statistical question relevant to all these stakeholders: how best to quantify spatial and temporal variation in TNC driver earnings. Due to a variety of statistical challenges, which we articulate below, this variation is not well understood—nor are there good methods for estimating this variation reliably, at the scale and speed needed for analyzing millions or billions of trips at high spatial and temporal resolution. Our goal in this paper is to address this gap. Specifically, we present a nonparametric method for estimating the probability density $f_{s,t}$, at location s and time t, for the productivity of a TNC driver (defined roughly as profit per hour and explained in detail below). Our method extends the spatial density smoothing framework of Tansey et al. (2017) to the spatiotemporal setting. We discuss the main challenges involved in this extension, and we provide a tool for spatiotemporal density smoothing based on the Graph-fused Elastic Net (GFEN), which can effectively address these challenges. We accompany this paper with code written in the Julia programming language[1].

We then apply our method using data on more than 1.4 million ride-sourcing trips taken on RideAustin, a local non-profit TNC in Austin, Texas, during a period in 2016-17 when leading national TNCs were temporarily out of the city.[2] Our analysis results in a number of interesting findings—many made possible by the fact that our method yields a full probability distribution of driver earnings as a function of both space and time, giving us access to such distributional features as quantiles and tail areas. To give two examples:

- The probability that a TNC driver can expect to earn a living wage in Austin (which we get from Nadeau, 2017) exhibits high variability with respect to space and time. For a parent of two children who works a typical Saturday late-night near downtown Austin, the probability of earning a living wage for the Austin area can exceed 90%. But at midday on a Monday far from the city center, this probability can fall below 40%.
- The bottom 10% of earners among drivers accepting rides at the airport have productivity below $10/hour in a typical Monday midday. This result is considerably lower than the living wage in Austin for a single adult with no children.

1.2 Ride-Sourcing Productivity Analysis: Background & Challenges

Spatiotemporal variation in driver earnings is at the heart of many challenges faced both by the designers of TNC pricing models and by the drivers themselves. For both the TNC and

[1] https://github.com/mauriciogtec/GraphFusedElasticNet.jl

[2] Uber and Lyft left the city from May 2016 to May 2017 after the Austin City Council passed an ordinance requiring ride-hailing companies to perform fingerprint background checks on drivers, a stipulation that already applies to Austin taxi companies (Samuels, 2017).

the drivers, a desirable property of a ride-sourcing platform is what Zuniga-Garcia et al. (2019) call *destination invariance*, which they define as "the principle that two drivers dispatched on different trips from the same location at the same time do not envy each other's expected future income." In reality, however, some trip opportunities yield higher continuation payoffs than others, which implies that at least some trips are mis-priced. From the driver's perspective, this mis-pricing can result in needlessly high volatility in driver earnings, and therefore substantial variation in the likelihood that a driver will earn a living wage.

Moreover, for both the driver and the TNC, such variation can also result in substantial market efficiencies, potentially impacting service reliability at the level of the whole network (Ma et al., 2018). For example, drivers may "chase the surge," avoid short trips, or refuse trips from particular pick-up locations or within particular time frames, thus leaving riders in some areas without access to the service.[3] Moreover, strategic and/or experienced drivers can learn how to improve their earnings by predicting profitable times and locations, which exacerbates disparities in driver earnings and satisfaction, as found by Cook et al. (2018). TNCs respond to this reality in a variety of ways. For example, Uber and Lyft tried to provide drivers with more flexibility by adding destination filters, where drivers can select the desired drop-off location that would allow them to relocate themselves as they preferred (Cradeur, 2019; Lyft, 2019). However, this feature caused a negative impact on the platform by increasing riders' waiting time and other drivers' pick-up time, as strategic drivers used the filter to select trips with better-earning potential, leading Uber to cap drivers' usage of this feature at twice per day (Perea, 2017).

Recent research efforts have addressed ride-sourcing's spatial mis-pricing problem by proposing different pricing strategies and driver-passenger matching functions. Some examples that have been studied include incorporating spatial surge pricing models (He et al., 2018; Bimpikis et al., 2016), search and matching models (Bian, 2018; Buchholz, 2015; Zha et al., 2018; Castro et al., 2018), non-linear pricing models (Yang et al., 2010), and spatiotemporal pricing mechanisms (Ma et al., 2018).

In this paper, we do not explicitly consider the question of how to design a better TNC pricing model. Rather, we take the perspective that before one can design such pricing models that mitigate spatiotemporal variation in driver earnings, one must first quantify the extent of that variation—and we, therefore, seek to provide a scalable, reliable method for doing so. But this poses a complex set of statistical challenges. First, the availability of public ride-sourcing data is limited, leading some authors to rely on simulations (Ma et al., 2018; Bimpikis et al., 2016) or to limit their research to taxi-only data (He et al., 2018; Buchholz, 2015). Second, when available, spatiotemporal information is subject to noise and high sparsity, where many combinations of space and time have no or very little data. This has previously led researchers to aggregate the data into large spatial and/or temporal (e.g. peak vs. off-peak hours) blocks, as in He et al. (2018), Buchholz (2015), and Bian (2018). This aggregation helps with data sparsity but compromises the ability to find valuable high-resolution insights. Low spatiotemporal resolution owing to data sparsity can be especially problematic when analyzing detailed pricing scheme consequences. We address this problem by relying on modern spatial smoothing/interpolation techniques that penalize total variation—although, as we describe below, we must modify these techniques to handle the data-sparsity problem in a way that still yields sensible interpolations.

Our approach encodes spatiotemporal structure using a graph, where each vertex corresponds to a traffic analysis zone (TAZ)[4] at a specific hour of the week, and where edges are

[3]These examples explain why platforms do not show the trip destination before the driver accepts the ride, as in Romanyuk (2017) and Campbell (2017).

[4]TAZs are geographic areas dividing a planning region into relatively similar areas of land use and activity.

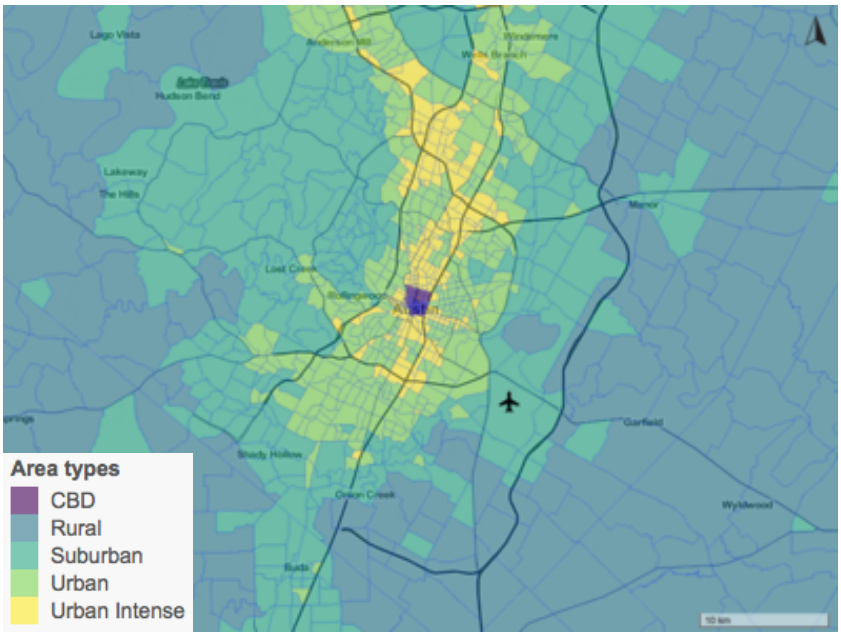


Figure 1: Austin traffic analysis zones (TAZs) classified by area type. Blue lines indicate TAZ boundaries. The central business district (CBD), which contain the downtown, is shown in purple.

used to denote geographical and temporal adjacency. Austin TAZs are shown in Figure 1. This specification results in a graph of 223k vertices, one for each location at a specific time. Our goal is to estimate a full probability distribution for the productivity of a driver at each one of these vertices (we provide further details about the construction of the graph in Section 3.3). Owing to the size of the graph, we shall build upon the scalable spatial density smoothing framework of Tansey et al. (2017); Zuniga-Garcia et al. (2019) have previously applied this technique to the RideAustin dataset and showed it to be effective in modeling purely spatial effects. However, in order to extend this framework to the spatiotemporal setting, we must address two challenges.

The first challenge comes from the question of how to smooth the raw data while still capturing important spatiotemporal effects. Space and time dimensions have different units and physical interpretations, suggesting that spatial versus temporal edges must be treated differently. Moreover, effects in space and time are likely to be a mix of smooth and non-smooth transitions. For example, the productivity of a driver may change drastically from one side to another of a highway or a river, but it would most likely be similar or even constant across highly interconnected regions with no obvious barriers. In the time dimension, by contrast, effects are more likely to be smooth, and yet there may still be sudden transitions caused by specific events, such as the increased temporal density of airport arrivals. The challenge here is to allow for separate but parsimonious spatial and temporal dynamics that incorporate a mix of both smooth and non-smooth features.

The second challenge arises from the fact that many vertices in our graph will have no data. For the RideAustin dataset, where we discretize by the hour of the week, 45.8% of the vertices of the graph have no observations. Every TAZ in our data set had at least one observation at some point of the week, but many do not have observations for every hour of the week. The challenge here is to develop a method that can borrow information efficiently across spatial and temporal adjacencies, estimating a density at every location for every hour, even if no data was observed.

1.3 Proposed Methodology

Tansey et al. (2017) propose a non-parametric density estimation technique coupled with the graph-fused lasso (GFL) to provide highly scalable and parallelizable density estimation for data distributed across a graph encoding purely spatial adjacencies. Their approach consists of three steps:

1. Split the overall problem into sub-problems recursively partitioning the variable's support into a series of half-spaces, each described by a conditional probability.

2. Smooth each half-space probability across space in such a way that encourages similarity between adjacent nodes of a graph.

3. Merge the smoothed half-space probabilities to yield full density estimates at each node.

Our proposed methodology will follow this broad outline, with some important differences in how we handle the smoothing in Step 2. The "ordinary" graph-fused lasso is not appropriate for our context because it does not distinguish between spatial and temporal edges on the graph. It is also not ideal for modeling a combination of smooth and non-smooth effects since the GFL produces estimates that are piece-wise constant across the graph. Finally, and most importantly, as we shall explain in Section 2, the GFL does not necessarily give sensible results in missing-data scenarios, since its objective function will no longer be strictly convex in this case. Ignoring this fact, or resolving it in a naive way, can lead to solutions with undesirable or counter-intuitive interpolation behavior. To address these limitations, we propose to combine the traditional ℓ_1 total variation penalty used by the GFL with an additional ℓ_2 total variation term, to impose different penalties across spatial and temporal edges. This combination will have the effect of enabling both smooth and non-smooth transitions. The ℓ_2 penalty alone is essentially equivalent to fitting a Gaussian Markov Random Field (Cressie, 1993); combining it with the ℓ_1 penalty yields something analogous to the elastic net regularization method for regression (Zou and Hastie, 2005). We call the resulting method the Graph-fused Elastic Net (GFEN).

Figure 2 gives a quick preview of the performance of this method on the RideAustin data set. It shows estimated densities using the GFEN method for three locations in Austin:

- The Austin–Bergstrom International Airport (ABIA), located approximately 5 miles south-east of downtown Austin.

- Downtown area, which we identify with the TAZ containing the intersection of Guadalupe & 6th Streets, which has very high trip demand.

- Red River at 12th Street, a small TAZ immediately next to downtown but with a low trip count.

We show reconstructed densities at two selected times: Sunday 3 AM and Monday 1 PM, which are characterized respectively by high and low demand overall across the city. Figure 2 shows that the method is capable of estimating complicated spatial and temporal interactions—for example, that these two times exhibit a significant difference in upper-tail thickness for downtown, but much less so at the airport. The results also show that even with no data observed at Red River & 12th—which is geographically next to downtown—it is still possible to interpolate a sensible probably distribution. We emphasize that this would not be possible using the Graph Fused Lasso, whose objective function is no longer strictly convex in this case.

We will provide an algorithm for implementing the GFEN using a modified version of the highly-parallelizable ADMM algorithm presented by Tansey and Scott (2015). The resulting

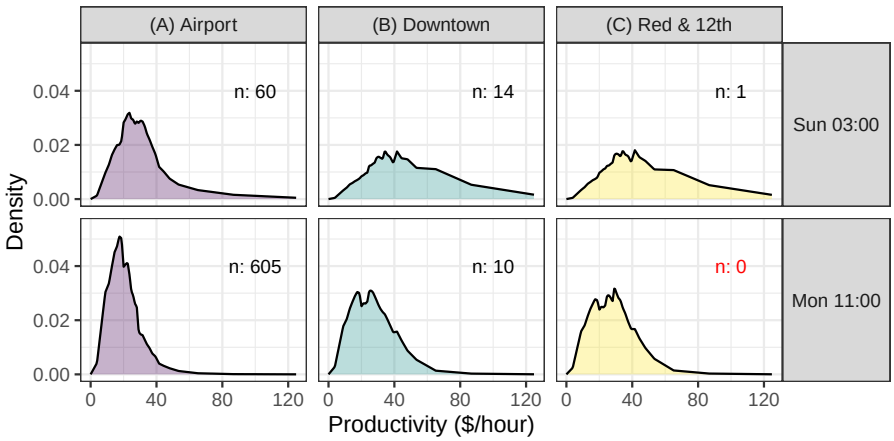


Figure 2: Shows examples of our density estimation methodology with the GFEN for some selected locations and hours. The number of data points at each specific location and time is indicated with n next to each density. Red River & 12th is a small TAZ with few observations next to downtown. The model borrows information from neighboring regions to estimate a model in this vertex.

algorithm will still consist of linear time updates and will scale to massive graphs just as its purely GFL-based counterpart. We present a discussion on the implementation details that is further detailed in the Appendix.

Our analysis will extend that of Zuniga-Garcia et al. (2019), who used the Graph-fused Lasso (GFL) to effectively estimate spatial effects on driver productivity. There are three main differences between their approach and one taken here. First, Zuniga-Garcia et al. (2019) only estimate spatial effects; and they only treat time by splitting the dataset into peak hours, midday, overnight, and weekend periods. Instead, we consider spatiotemporal effects with fine time granularity. Second, the authors only estimate mean effects, whereas here we seek full proability distributions. In particular, we exploit spatiotemporal quantiles and measures of spread learned from our estimated distributions to enrich our analysis. Finally, they only model the type of piece-wise constant effects that are well captured by the GFL (Tibshirani et al., 2005), whereas here we will consider both smooth and non-smooth effects.

1.4 Outline of the Paper

In Section 2 we explain the methodology taken in this paper to solve the key challenges mentioned in Subsection 1.2. In Section 3 we implement our proposed methodology to a case study, the RideAustin dataset, and use it to extract meaningful insights from the data. Finally, Section 4 concludes with some final remarks.

2 Methodology

2.1 Spatiotemporal Graphs and Densities

Undirected graphs are useful for encoding discrete space and time structures. Let $G = (V, E)$ be an undirected graph where V and E are respectively the set of vertices and edges. We say that G is a spatiotemporal graph when G has the following structure:

- There is exactly one node for every location and in each moment in time. The vertex

set can be written as $V = S \times T$ where S and T are the sets of locations and times, respectively. The temporal slice at location s is the set $\{s\} \times T$ and the spatial slice at time t is the set $S \times \{t\}$.

- Edges are either spatial or temporal. Thus the set of edges can be written as a disjoint union $E = E_S \cup E_T$, where E_S connects adjacent nodes in the same spatial slice and E_T connects nodes in the same temporal slice.

A spatiotemporal graph is as the Cartesian product graph $G_S \square G_T$, where G_S is a graph encoding the spatial structure and G_T is a graph encoding the temporal structure (Imrich and Klavzar, 2000).

At each vertex (s,t) we have a set of observed data

$$\mathbf{y}^{(s,t)} = \left\{ y_1^{(s,t)}, \ldots, y_{N^{(s,t)}}^{(s,t)} \right\}, \quad y_i^{(s,t)} \overset{iid}{\sim} f(s,t)$$

where $f(s,t)$ is the density function that we seek to estimate. The goal of spatiotemporal graph-based density smoothing is to estimate density $f(s,t)$ in a way that borrows information from neighboring regions as encoded by G. Graph smoothing is particularly useful when $N^{(s,t)}$ is small or zero in some vertices, and thus independently estimating a model at each vertex of the graphs leads to poor estimates.

2.2 Density Estimation with a Binary Partition

We now briefly review the technique of estimating a density based on a recursive dyadic partition, as proposed by Tansey et al. (2017). The assumptions for this approach are:

1. The output space of the data is a known set B so that $\mathbf{y}^{(s,t)} \subset B$ for all $(s,t) \in V$;
2. Given a fixed max depth K, we can recursively define a dyadic partitioning scheme as follows. First, we assume that B can be written as a union of disjoint non-empty sets $B = B_0 \cup B_1$. Now, for every $k \in \{1, \ldots, K-1\}$ and for every B_γ where $\gamma \in \{0,1\}^k$ we have that $B_\gamma = B_{\gamma 0} \cup B_{\gamma 1}$ is a union of disjoint non-empty sets. We refer to $B_{\gamma 0}$ and $B_{\gamma 1}$ as the left children and right children of B_γ.

The partitioning scheme defines a depth-K binary tree structure on B given by

$$B^{(K)} := \bigcup_{k=1}^{K} \left\{ B_\gamma \mid \gamma \in \{0,1\}^k \right\}.$$

The nodes B_γ where $\gamma \in \{0,1\}^K$ is of length K are called the terminal nodes or leaves of the tree. Note that for every $k \leq K$ we have a decomposition of B into 2^k disjoint sets.

$$B = \bigcup_{\gamma \in \{0,1\}^k} B_\gamma.$$

To illustrate the idea, suppose $B = [0,1)$. We could then define $B_0 = [0, 1/2)$ and $B_1 = [1/2, 1)$. Similarly, we could write $B_{00} = [0, 1/4)$, $B_{01} = [1/4, 1/2)$, $B_{10} = [1/2, 3/4)$ and $B_{11} = [3/4, 1)$ and so on.

Assume now we are given a partition tree $B^{(K)}$ and let $Y \sim f$ be some random variable with output space B. Then the goal is to estimate the quantities

$$\omega_\gamma = P(Y \in B_{\gamma 0} \mid Y \in B_\gamma)$$

which are the probabilities of a data point belonging to the left child $B_{\gamma 0}$ provided it belongs to the parent B_γ. We can use the variables ω_γ to give a non-parametric estimate of the probability distribution of Y on the leaves of tree. The resolution level is determined by the depth of the tree K. More precisely, for any $\gamma = (\gamma_1, \ldots, \gamma_K) \in \{0,1\}^K$ we have

$$\begin{aligned} P(Y \in B_\gamma) &= \prod_{j=0}^{K-1} \text{Bernoulli}\left(\gamma_{j+1} \mid \omega_{(\gamma_1,\ldots,\gamma_j)}\right). \\ &= \prod_{j=0}^{K-1} \omega_{(\gamma_1,\ldots,\gamma_j)}^{\gamma_{j+1}} \left(1 - \omega_{(\gamma_1,\ldots,\gamma_j)}\right)^{1-\gamma_{j+1}}. \end{aligned} \tag{1}$$

We now put the above formulation back into our spatiotemporal model. Define $m_\gamma^{(s,t)}$ as the count of total observations of $\mathbf{y}^{(s,t)}$ that fall within B_γ. Then for each non-terminal node B_γ we have

$$m_{\gamma 0}^{(s,t)} \sim \text{Binomial}(\omega_\gamma^{(s,t)}, m_\gamma^{(s,t)}) \tag{2}$$

Expression (2) enables us to estimate the ω_γ from the data. We simply have to count the number of occurrences in each bin and estimate a set of binomial probabilities for the tree structure. Expression (1) can then be used to recover the full density estimates.

It will be convenient to reparameterize (2) in terms of log-odds with a variable β_γ. Thus, for each vertex (s,t) and for each non-terminal node B_γ the corresponding negative log-likelihood function used as loss function is

$$l_\gamma(\mathbf{y}^{(s,t)}, \beta_\gamma^{(s,t)}) := -m_{\gamma 0}^{(s,t)} \log \sigma(\beta_\gamma^{(s,t)}) - m_{\gamma 1} \log(1 - \sigma(\beta_\gamma^{(s,t)})). \tag{3}$$

where $\sigma(\beta_\gamma) := (1 + \exp(-\beta_\gamma))^{-1}$.

2.3 The Graph-fused Lasso and Behavior with Missing Data

In the precedent section, we presented a binomial probability model for the splitting probability at each node of a tree structure, which corresponds to split step in the split/smooth/merge approach for density smoothing. For the smoothing step, Tansey et al. (2017) propose to use the Graph-fused lasso (GFL). We now briefly recall the method. We will also discuss the complications that will arise with the missing data scenario that is present in the research problem that we address in this paper. We shall compare the GFL with its counterpart method, the Gaussian Markov Random Field (GMRF). Our discussion in this section will motivate the introduction of the Graph-fused Elastic Net (GFEN) in the next section. Since the smoothing step will be identical for every node B_γ of the tree, we will drop γ from the notation in the interest of presentation.

The GFL objective The idea of graph-based denoising is to penalize big differences along edges. Let $\lambda > 0$ be a penalization parameter. Then the classical GFL objective is

$$\underset{\beta}{\text{minimize}} \quad \sum_{v \in V} l(\mathbf{y}^{(v)}, \beta^{(v)}) + \lambda \sum_{vw \in E} \left|\beta^{(v)} - \beta^{(w)}\right|. \tag{4}$$

where l is any loss function used to estimate a model. In our case, it will be the loss defined in expression (3). For spatiotemporal graphs, we will extend this definition to include different penalization parameters for spatial and temporal edges. In which case, the GFL objective becomes

$$\underset{\beta}{\text{minimize}} \quad \sum_{v\in V} l(\mathbf{y}^{(s,t)}, \beta^{(v)}) + \lambda_S \sum_{vw\in E_S} \left|\beta^{(v)} - \beta^{(w)}\right| + \lambda_T \sum_{vw\in E_T} \left|\beta^{(v)} - \beta^{(w)}\right|. \tag{5}$$

The GMRF objective. The Gaussian Markov Random Field approach is very similar, but instead, it would use a quadratic penalization for the edge differences. Thus the GMRF objective is

$$\underset{\beta}{\text{minimize}} \quad \sum_{v\in V} l(\mathbf{y}^{(v)}, \beta^{(v)}) + \sum_{vw\in E} \lambda \left(\beta^{(v)} - \beta^{(w)}\right)^2. \tag{6}$$

and similarly for spatiotemporal graphs

$$\underset{\beta}{\text{minimize}} \quad \sum_{v\in V} l(\mathbf{y}^{(s,t)}, \beta^{(v)}) + \lambda_S \sum_{vw\in E_S} \left(\beta^{(v)} - \beta^{(w)}\right)^2 + \lambda_T \sum_{vw\in E_T} \left(\beta^{(v)} - \beta^{(w)}\right)^2. \tag{7}$$

Both models have been used and studied extensively for smoothing and denoising problems. We shall not attempt to provide a systematic comparison. However, we will use a simple example to illustrate the difference between both models, with particular attention to the missing data case.

Problem 1 (Denoising a small chain graph with missing data)**.** *Consider a chain graph with three vertices $V = \{1, 2, 3\}$ and two edges $E = \{e_{12}, e_{23}\}$. At vertices 1 and 3 we observe the data points y_1 and y_3 respectively, assuming wlog that $y_1 < y_3$. But at vertex 2 we observe no data. The loss function we will consider is $l(y_i, \beta_i) := (y_i - \beta_i)^2$. Given a fixed $\lambda > 0$, the total-variation denoising problem takes the form*

$$\underset{\beta}{\text{minimize}} \quad (y_1 - \beta_1)^2 + (y_2 - \beta_2)^2 + \lambda \left(|\beta_2 - \beta_1|^p + |\beta_3 - \beta_2|^p\right)$$

where $p = 1$ corresponds to the GFl and $p = 2$ corresponds to the GMRF.

Fact 1 (GFL solution)**.** *If $\lambda < \frac{1}{2}(y_3 - y_1)$ then there is no unique solution to the GFL objective (4) and the solution set can be described as*

$$\begin{aligned}
\hat{\beta}_1 &= y_1 + \lambda \\
\hat{\beta}_2 &\in (\hat{\beta}_1, \hat{\beta}_3) \\
\hat{\beta}_3 &= y_3 - \lambda.
\end{aligned}$$

If $\lambda \geq \frac{1}{2}(y_3 - y_1)$ then there is a unique solution $\hat{\beta}_1 = \hat{\beta}_2 = \hat{\beta}_3 = \frac{1}{2}(y_1 + y_3)$ at the midpoint of the observed data points.

Remark 1. Fact 1 shows that the GFL does not perform any interpolation and therefore there is no unique solution for the missing data point. A common regularization technique in many problems is to add a shrinking prior of the form $\|\beta\|$ for some norm $\|\cdot\|$. We can observe that in this case it would not be desirable since it would have a strong informative effect and result in $\hat{\beta}_2$ being set to the end of the solution interval that is closest to zero.

Remark 2. If we consider the binomial model (2) that is the basis for the density estimation technique of this paper given by $m_{\gamma 0}^{(i)} \sim \text{Binomial}(\sigma(\beta\gamma^{(i)}), m_\gamma^{(i)})$. Then the solution will be the same as in Fact 1 with $y_i = \log(\hat{p}_i/(1 - \hat{p}_i))$ and $\hat{p}_i = m_{\gamma 0}^{(i)})/m_\gamma^{(i)}$.

Remark 3. If we consider the two previous remarks together, we see that adding a prior of the form $\|\beta\|$ would shrink the splitting probabilities towards 1/2. This will result in an undesirable informative effect of the the binary tree decomposition. For example, a binary tree with uniformly spread splits in B with even split probabilities will result in a uniform distribution in B. On the other hand, one can take any distribution f and a partitioning scheme where the splitting values come from a set of quantiles $\{q_{\alpha_i}(f)\}$ where the quantile probabilities are uniformly spread in $(0, 1)$. Then having even split probabilities will yield return the distribution f itself up the the resolution of the tree. Thus, these type of regularization priors are not advised for the density estimation technique considered in this paper.

Fact 1 also highlights one of the good properties of the GFL. For small values of λ, the estimation of β_1 depends only on y_1 and not on y_3. On the other hand, for larger values of λ, the estimates are collapsed, creating flat regions. Intuitively, this is why the GFL estimates consist of flat regions with discontinuous jumps. We now compare with the solution of the GMRF.

Fact 2 (GMRF solution)**.** *For every $\lambda > 0$ the solution set of the GMRF objective (6) is*

$$\begin{aligned}\hat{\beta}_1 &= y_1 + \frac{\lambda}{1+\lambda} \cdot \frac{y_3 - y_1}{2} \\ \hat{\beta}_2 &= \frac{\hat{\beta}_1 + \hat{\beta}_3}{2} \\ \hat{\beta}_3 &= y_3 - \frac{\lambda}{1+\lambda} \cdot \frac{y_3 - y_1}{2}.\end{aligned}$$

Thus the vertex with missing point is assigned to the middle point, regardless of the value of λ. Moreover, all the points converge to the middle point as $\lambda \to \infty$.

Fact 2 shows two fundamental differences in comparison with the GFL. First, we observe that the GMRF performs interpolation: regardless of the value of λ, the missing data vertex is assigned to the middle point of its neighbors. Second, the difference between y_1 and y_3 determines the magnitude of the smoothing effect. As a consequence, outliers will have a stronger impact on the GMRF solution than on the GFL. Also, here β_1 and β_3 are only asymptotically converging to the middle point, whereas the GFL would collapse them for high enough value of λ.

2.4 The Graph-fused Elastic Net (GFEN)

The GFEN is Graph-fused Elastic Net (GFEN) is essentially the combination of the GFL and GMRF penalties. Since notation can become cumbersome, we introduce the following more general notation and definition.

Definition 1 (ℓ_p-total variation)**.** *Given a graph $G = (V, E)$ and a set of scalar parameters $\beta = \{\beta^{(v)}\}_{v \in V}$ at each vertex of the graph. The ℓ_p-total variation of β along a subset of edges $E' \subset E$ is defined as*

$$\mathrm{TV}_p(\beta, E') = \sum_{vw \in E'} |\beta^{(v)} - \beta^{(w)}|^p \tag{8}$$

where $p > 0$.

Definition 2 (GFEN)**.** *Given a graph $G = (V, E)$, and a set of negative log-likelihoods $l(\mathbf{y}^{(v)}, \beta^v)$ at each node $v \in V$. The GFEN problem is defined as*

$$\underset{\beta}{\text{minimize}} \sum_{v \in V} l(\mathbf{y}^{(v)}, \beta^{(v)}) + \sum_{p \in \{1,2\}} \lambda_p \mathrm{TV}_p(\beta, E) \tag{9}$$

where $\lambda_1, \lambda_2 > 0$ are penalty hyperparameters for each norm. If G is a spatiotemporal graph with vertex set $S \times T$ and spatial and temporal edges E_S and E_T respectively. Then the spatiotemporal problem is

$$\underset{\beta}{\text{minimize}} \sum_{(s,t) \in S \times T} l(\mathbf{y}^{(s,t)}, \beta^{(s,t)}) + \sum_{d \in \{S,T\}} \sum_{p \in \{1,2\}} \lambda_{d,p} \text{TV}_p(\beta, E_d) \tag{10}$$

where $\lambda_{S,1}, \lambda_{S,2}, \lambda_{T,1}, \lambda_{T,2} > 0$ are the penalty hyperparameters.

To provide further intuition on the effect of combining the norms, we provide the solution to Problem 1 given by the GFEN.

Fact 3 (GFEN solution)**.** *The solution of the GFEN objective (9) for Problem 1 is given by*

$$\begin{aligned}
\hat{\beta}_1 &= y_1 + \lambda_1 + \frac{\lambda_2}{1+\lambda_2} \cdot \frac{y_3 - y_1}{2} \\
\hat{\beta}_2 &= \frac{\hat{\beta}_1 + \hat{\beta}_3}{2} \\
\hat{\beta}_3 &= y_3 - \lambda_1 - \frac{\lambda_2}{1+\lambda_2} \cdot \frac{y_3 - y_1}{2}.
\end{aligned}$$

for all $\lambda_1, \lambda_2 \geq 0$ such that

$$\lambda_1 + \frac{\lambda_2}{1+\lambda_2} \cdot \frac{y_3 - y_1}{2} < \frac{y_1 + y_3}{2}.$$

If the above condition does not hold, then there is a unique solution $\hat{\beta}_1 = \hat{\beta}_2 = \hat{\beta}_3 = \frac{1}{2}(y_1 + y_3)$ at the midpoint of the observed data points. Thus, the GFEN interpolates $\hat{\beta}_2$ to the middle point regardless of the smoothing hyperparameters. And it collapses the estimates to a flat region for high penalties.

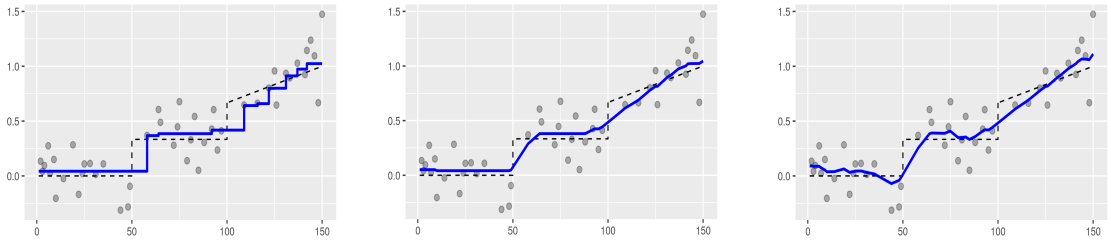
The relative weights assigned to the ℓ_1 and ℓ_2-total variation penalties in expressions (9) and (10) have a very intuitive interpretation. The ratio to each norm controls the degree of sharpness and smoothness in the solutions. The GFEN solutions will have both smooth transitions and piece-wise constant transitions with sharp edges. It will also always interpolate in missing data vertices. In the context of density estimation with a binary tree, interpolation will remove the undesired dependence on the splitting values that may arise from shrinking the log-odds coefficients towards zero. This intermediate behavior is illustrated in Figure 3, which shows a scenario similar to Problem 1 with missing data vertices on a chain graph.

There is an alternative intuitive interpretation for adding a small extra ℓ_2 total variation regularization. It can be regarded as adding a prior that shrinks the GFL estimates towards the average estimates of its neighboring vertices, in contrast with the classical shrinking prior that contracts estimates towards the origin.

2.5 Implementation Details and Model Selection

The GFEN problem (9) can be solved at scale using a variant of the fast algorithm for the GFL presented by Tansey and Scott (2015). A full derivation is presented in Appendix A. Here we summarize the main ideas behind this approach:

1. We start from a decomposition of the edges E into a set of non-overlapping trails $E = \bigcup\{\tau \mid \tau \in \mathcal{T}\}$. The core of the strategy is to reduce the optimization objective to solving individual smoothing or "proximal" problems along each trail for each ℓ_p penalty



(a) GFL (b) GFEN (c) GMRF

Figure 3: Comparison of methods on the estimation of a signal on a chain graph of size $N = 150$. The true signal (dashed) is a mix of piece-wise constant and smooth transitions. Data is observed with noise, with 66% of the vertices having missing data. The GFL (left) does a good job at estimating the flat regions of the signal. Since a quadratic shrinking penalty was added to the GFL to ensure the problem is strictly convex due to the missing data, regions with missing observations are shrunk towards the origin to the closest value with data. The GMRF estimates a smoother signal but with more sensitivity to large deviations. The behavior of the GFEN is intermediate between the other two, being piece-wise constant and smooth in the corresponding regions of the true signal and interpolating in regions with missing data. The hyperparameters for all models were independently chosen, using cross-validation to maximize the out-of-sample quadratic error.

separately. This approach is sometimes referred to as "proximal stacking" (see Barbero and Sra (2018)). The advantage of doing this is that the optimization problem in each trail is 1-dimensional and can be solved very efficiently.

2. A principled mathematical way is necessary to combine the solutions in each trail and guarantee that the original objective is minimized. To do this, we first introduce a slack variable $z_{\tau,p}$ for each trail τ and for each $p \in \{0, 1\}$. We will require the following linear constraints which define a consensus problem $z_{\tau,p} = \beta[\tau]$[5] where β is the solution of the GFEN. We use the ADMM algorithm to solve the consensus problem, which is attractive because of its flexibility and convergence properties (Boyd et al., 2011). The ADMM is an iterative technique for solving convex optimization problems of the form $\min_{\beta,z} f(\beta)+g(z)$ subject to a linear constraint of the form $A\beta = Bz$. Here, f will take the role of the likelihood and g of the total variation penalty. The variable z will be a concatenation of each variable $z_{\tau,p}$ and the linear constraints will come from the restrictions $z_{\beta,\tau} = \beta[\tau]$. Each iteration of the ADMM will consist of three simple unconstrained optimization subproblems that only involve one variable at a time.

3. The optimization subproblem corresponding to the update for β can be replaced by an iteration of Newton's method. For the updates corresponding to z, which itself will consists of parallel updates for each $z_{\tau,p}$, we leverage exact linear-time solvers for 1-dimensional smoothing problems. For $p = 1$ we use the method of Barbero and Sra (2018) for ℓ_1-total variation denoising and for $p = 2$ we use the Kalman smoother (see Welch et al. (1995)).

We now discuss the tuning of the penalty hyperparameters. The fact that the GFEN is able to handle smooth and non-smooth transitions, as well as dealing with differentiated spatial and temporal dynamics, comes at the expense of increasing the number of hyperparameters to tune.

[5]This notation is used as follows. A trail τ can be interpreted as a sequence of visited vertices $\tau = (v_1, \ldots, v_k)$. We then define $\beta[\tau] := (\beta^{(v_1)}, \ldots, \beta^{(v_k)})$.

First, solution path approaches that are typically used to tune the GFL penalty hyperparameters will no longer be effective, since they are based on gradually increasing the value of a single hyperparameter (see Tibshirani et al. (2005)). Second, tuning methods based on information criteria that depend on degrees of freedom (e.g., AIC, BIC) are harder to compute since the smoothness induced by the additional ℓ_2 total variation penalty estimates the effective degrees of freedom very hard. We propose instead to optimize the hyperparameters using a cross-validation framework with the out-of-sample negative log-likelihood. To select new candidates of hyperparameters to test we use strategy of Bayesian optimization (Shahriari et al., 2016), which is based on modeling the cross-validated loss as a function of the hyperparameters using a Gaussian Process, which allows to give a prediction for the loss of unseen points, which is used to select promising candidates. Further details are given in Appendix B.

We found it useful to select the tree splitting scheme based on the quantiles of the global distribution. It provided more numerical stability since many of the subproblems had more balanced data. Also, it naturally provides more resolution in values that have higher density. See Appendix C for further discussion.

2.6 Statistical Background & Related Work

Our methods build on two independent lines of work: first, spatial density smoothing over a graph; second, extensions of spatial models to spatiotemporal models. We do not attempt to provide a fully detailed literature review of these mature fields. But we will outline the relevant work most closely related to our methodological approach.

First, for background on spatial density smoothing over graphs, we refer the reader to the paper by Tansey et al. (2017), as well as the references therein. Our method is an extension of their methodology (see the previous section) that uses the same technique of representing a probability distribution via a recursive dyadic partition. This technique has been widely used—for example, in multiscale models for Poisson and multinomial estimation (Fryzlewicz and Nason, 2002; Jansen, 2006; Willett and Nowak, 2007), and in nonparametric Bayesian inference via Polya-tree priors (Mauldin et al., 1992; Lavine et al., 1992, 1994). Another paper that we refer the reader to for a review is due to Zhao and Hanson (2011), who take a related approach by coupling a Polya-tree prior with a conditional autoregressive (CAR) prior in a fully Bayesian model. The scalable algorithm for spatial denoising over a general graph that we take as a departure point for this paper is provided by Tansey and Scott (2015), which in turn builds on computationally efficient estimation for the class of convex optimization problem that arises from the GFL model, including the papers by Tibshirani and Taylor (2011); Ramdas and Tibshirani (2016); Wang et al. (2016).

The GFL is related to the technique known as total variation denoising in the signal processing literature (Getreuer, 2012). The total variation penalty based on the ℓ_1-norm is used to denoise images by encouraging locally constant effects with sharp edge boundaries. There are highly scalable algorithms to implement such models for image data, for example, based on the parametric max-flow algorithm (Hochbaum, 2001; Chambolle and Darbon, 2009). Modeling spatial effects using total variation denoising with the ℓ_2-norm is a special case of a Gaussian Markov Random Fields (GMRF) (Rue and Held, 2005). GMRFs are also common in the image denoising literature. A recent linear-time algorithm is provided by Yasuda et al. (2018). GMRFs are well-known to promote smooth edges as opposed to the sharp contrasts encouraged by the ℓ_1-norm. Note that these image denoising techniques do not address the problem of full density estimation since they are primarily concerned with denoising a signal. Moreover, they are typically designed for rectangular grids, which are common with images and other signal data.

In the statistics literature, the GFL is also related to some work in Bayesian inference for spatial data models. Specifically, it is similar to conditional auto-regressive (CAR) models (Besag, 1974), which also affect spatial smoothing by discouraging large pairwise differences across edges in a graph. Bayesian models consider the approach of estimating the density from the point of view of the posterior predictive density. Non-parametric Bayesian approaches for density estimation with spatial smoothing were investigated by Gelfand et al. (2005), Reich and Fuentes (2007), Rodríguez et al. (2010), among many others. Also highly related is the approach by Li et al. (2015), who propose a non-parametric Bayesian model for areal data that can detect boundaries between spatial neighbors where a sharp change occurs. We refer the interested readers to these papers and their references for more detail.

We now consider the background work regarding the extension of spatial models to spatiotemporal settings. In the statistics literature, they have been two main approaches for this task. The first approach consists of treating time as an additional undifferentiated dimension from space. This approach would typically involve estimating a covariance matrix that includes both the space and time dimensions, e.g. (Cressie and Huang, 1999; Allcroft and Glasbey, 2003). More generally, this approach can be used for all kernel methods (Bashtannyk and Hyndman, 2001). The drawback of such approaches is that kernel methods depend on a meaningful distance measure, which is problematic for spatiotemporal modeling since space and time have different units and physical interpretations.

The second approach within the statistics literature is to use dynamic probabilistic models (Stroud et al., 2001). In this approach, the parameters of a spatial model are assumed to change smoothly over time following an auto-regressive model. For example, in (Cressie et al., 2010) and (Katzfuss and Cressie, 2011), the authors model spatial effects using low-rank basis approximations to the spatial correlation function and auto-regressive processes for the temporal dependence. Closely related approaches combining low-rank basis approximation and dynamical models in the fully Bayesian setting can be found in (Katzfuss and Cressie, 2012) and (Finley et al., 2012). Auto-regressive processes based on the Gaussian distribution are essentially GMRFs on the temporal dimension and are related to the Kalman Filter. An example application of GMRFs for both spatial and temporal effects in disease modeling is given by Rushworth et al. (2017). As pointed out by Xu et al. (2015), GMRFs have the advantage of being able to deal with missing observations at some point in time, which is common in geostatistical practice and is one of the challenges for the RideAustin dataset.

In the video denoising literature, the smooth transitions induced by the GMRF model inadequately model motion, which has lead to the search for alternative transition distributions for Markov Random Fields—for example, Chen and Tang (2007) consider estimating a non-parametric transition distribution for the temporal dimension. Total variation denoising based on the ℓ_1-norm has also been used for video denoising tasks. For a recent survey, we refer the reader to the introduction of (Arias and Morel, 2018) and (Parisotto and Schönlieb, 2019). However, similar to the image denoising case, most of these methods are designed to work for a rectangular grid only and not for general graphs. They are also generally focused on denoising a signal, and not on density estimation.

3 Case study: Driver Productivity Analysis

3.1 Ride-sourcing data

The non-profit TNC *RideAustin*, based in Austin, Texas, published data about their ride-sourcing service in early 2017 (Data World, 2017). The dataset records rides the happened between June 2nd, 2016, and April 13th, 2017. Each trip corresponds to a row in the database

and includes information about the origin and destination coordinates, starting and ending time, driver number, cost and request time. During this period *RideAustin* had no major competition since rival companies such as Uber and Lyft were temporarily restricted from operating in Austin.

Since the demand during the first month was limited, we restricted our analysis on data from September 1st, 2016 to April 13th, 2017. We selected rides having the origin and destination coordinates within the traffic analysis zones (TAZs) of Austin.

Since a trip can have different vehicle categories and rates (standard, premium, luxury, and sport utility vehicle [SUV]), to make every trip comparable, we standardized all of them to the regular car category using RideAustin's public pricing formula. We did the same thing to remove the surge price from some trips. Our motivation for removing the surge price is that whereas the pricing scheme is known and dependent on standard features such as mileage and time rates, the surge price onset and offset is less predictable.

3.2 Measuring productivity

To measure driver productivity, we follow the approach suggested by Zuniga-Garcia et al. (2019). Our *productivity* measurement, π, is taken *prospectively* from the driver's future rides. Consider a driver serving a rider during the first trip ($Trip$ 1), originated from the pick-up location $r \in R$ to the destination drop-off location $s \in S$, during the ride duration t_{rs}. After finishing this trip, the driver must wait for the system to assign the second trip ($Trip$ 2) at the pick-up location $r^* \in R$ with a destination location $s^* \in S$. We then denote:

- Driver-idle time (w_{sr^*}): the time in hours that the driver of $Trip$ 1 will wait until a subsequent trip is assigned.
- Reach time (p_{r^*}): the time between the trip assignment and the rider pick-up for $Trip$ 2.
- Duration of $Trip$ 2 ($t_{r^*s^*}$): the time in hours that the same driver will take to complete the subsequent trip; during this time the driver is generating revenue determined by the tariff system.
- Fare of $Trip$ 2 ($F_{r^*s^*}$): the final fare of the subsequent trip; it is a function of a distance and a time tariff rate.

Our variable of interest is then defined as

$$\pi_s := \frac{F_{r^*s^*}}{w_{sr^*} + p_{r^*} + t_{r^*s^*}} \tag{11}$$

Expression (11) yields an interesting definition of productivity because it combines the time that the driver will stay unproductive with the quality of the subsequent trips. Moreover, its values are given naturally in dollars/hour. The idea is that when a trip ends, the driver starts searching for new riders. This prospective measurement gives the expected earnings given that a driver is at the specific location and time at which the last trip ended. If a trip ends in a location of low demand, the idle time will be large, but also subsequent trips could be longer. We remark that this definition deliberately ignores the fare trip that led to that position. Figure 4a presents the distributions of the productivity values for all trips considered.

We only considered trips in which the waiting time for a subsequent trip was less than one hour. This assumption was necessary in order to exclude the cases where the driver took a break or stopped working for the day. Since during the time of data collection RideAustin did not have a major competing company, we do not have problems with long inter-trip times due to app switching. Figures 4b and 4c shows the distribution of the idle time in between trips after preprocessing and its distribution across TAZs.

3.3 Construction of the Spatio-Temporal Graph

The TAZs of Austin, shown in Figure 1, provide the advantage of using a size that vary accordingly to the traffic intensity, which is also correlated to the number of trips present in the dataset. We thus have a high resolution near downtown and a low resolution in more rural areas. A total of $1,333$ TAZs were considered, requiring that at least one trip originated or ended in that location. Time was discretized hourly, with a periodicity of one week, for a total of $168 = 24 \times 7$ time periods.

Altogether, we considered $223,944 = 168 \times 1,333$ units of observation, which we used as the vertices of an undirected spatiotemporal graph. Figures 4d and 4e shows the total counts of trips aggregating marginally for each time unit and each space unit. When considered marginally, all space units and all time units have some data. However, once we split by both space and time, only 102,600 of the 223,944 nodes (45.8%) have some data.

We construct a set of edges E as the union of a disjoint set of spatial and temporal edges $E = E_S \cup E_T$. The set E_S of edges in the spatial direction was constructed geographically, drawing an edge for all geographically adjacent TAZs for spatial a slice at every time. We excluded a few TAZs that were disconnected to the largest connected component of the graph. Edges in the temporal direction E_T were built for every time slice, i.e., joining the vertices for the same TAZ in subsequent hours. An extra edge $(s, 168)$ between $(s, 1)$ was added for every TAZ s to account for the weekly periodicity.

3.4 Model Selection Details

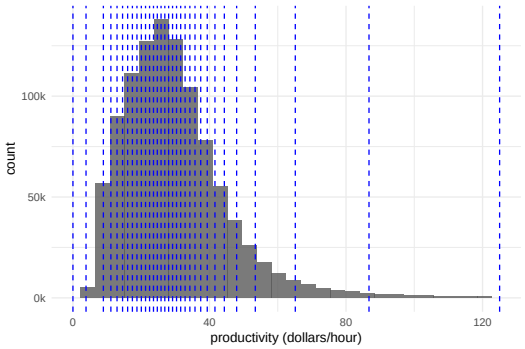
We used a binary tree of 5 levels, yielding 32 bins. To choose this bins smartly, we used quantiles of the global productivity distribution to define splitting points in the range (0, 125) which contains 99.98% of the data[6]. The resulting cutting points are shown in Figure 4a. While this choice has the undesirable consequence of making our model choice data-dependent, it brings two great benefits:

- It naturally uses resolution in regions that had more observations, alleviating the effects of discretization and decreasing the need for deeper binary trees.
- It generates better balanced splits unless the local distribution of a vertex dramatically differs from the global distribution. The quantity of observations left, and right of each split is more similar, which significantly improved the inference process since logistic regression does not perform well in highly unbalanced data scenarios.

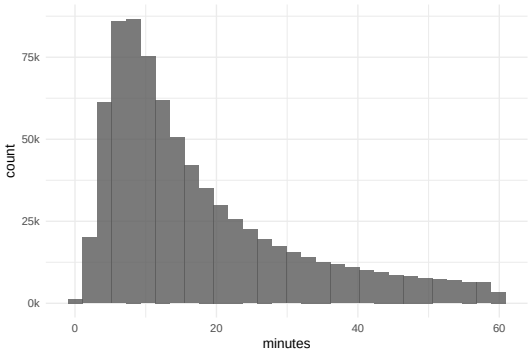
We used the Bayesian Optimization procedure that we described in Section 2.5 and Appendix B. For the Gaussian process, we use the radial kernel $K_{ij} = \exp(-a\|\lambda_i - \lambda_j\|_2^2)$ with $a = 0.5$ and with a small value of $\sigma = 10^{-8}$ for the observation uncertainty.

To exploit parallelism, we proceed in generations; in each step, we use the current estimate of the predictive distribution to generate a sample of hyperparameters with small expected loss. We then estimate the cross-validated loss for each one of them in parallel. After that, we update the Gaussian Process and sample a new generation of candidates. In our case, ten generations of size 16 worked well.

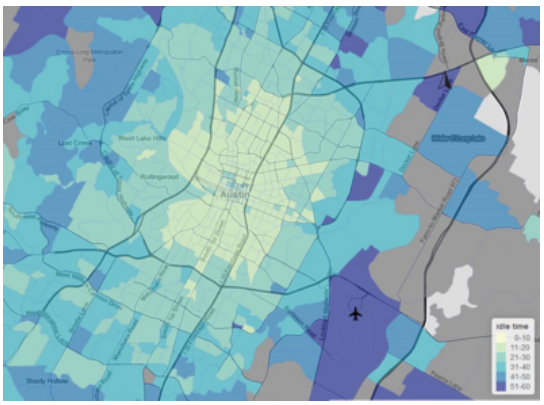
[6] After examining the data, there are reasons to believe that the top 0.02% observations come from failures in the data recording system.



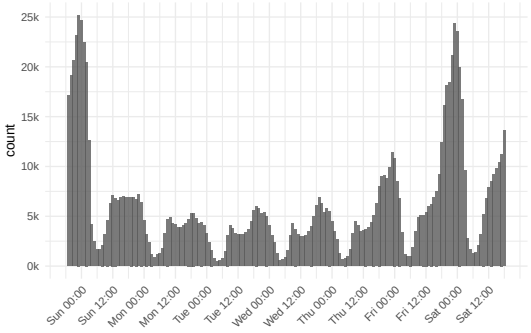
(a) Histogram of productivity considering all trips. Vertical dashed lines are the splits used by the binary tree constructed using the quantiles of the global distribution shown in gray.



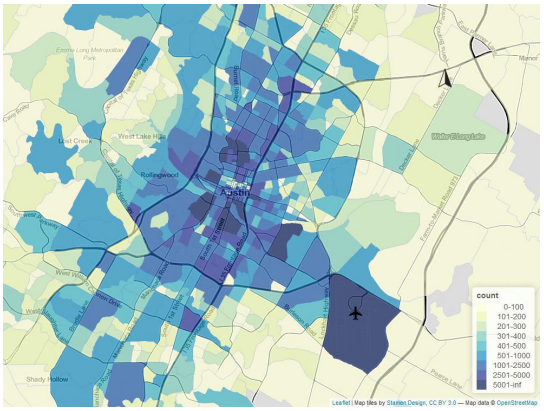
(b) Distribution of idle time after trip end



(c) Median idle minutes in between trips by TAZ



(d) Counts aggregated by time units



(e) Counts aggregated by space units

Figure 4: Description of the estimated productivity metrics

3.5 Results

3.5.1 Overview of the inference results

Figure 5 shows the estimated densities. We include five locations that have distinct characteristics that highlight different aspects of the inference. Table 1 shows a summary description of the selected sites. We choose central areas with high and low demand (university, downtown, Red River & 12th), two suburbs with different trip count (Domain and Pflugerville), and the airport area. We show the estimates in intervals of 12 hours from 3 AM to 3 PM. Several interesting

Table 1: Locations used in Figure 5 for comparison of density estimates

Location	Description
Airport	Austin-Bergstrom International Airport. It is a single TAZ with more trips.
The Domain	Office, retail, and residential center outside central Austin. It has a medium-low density of trips.
Pflugerville	Large suburban area located far from the central area. It has a low count of trips.
University	The University of Texas at Austin main campus. It is located adjacent to the central business district. It has a high number of trips.
Downtown	Central business district. It comprises several small TAZs with a large number of trip counts. An arbitrary TAZ was selected in the intersection of 6th & Guadalupe St.
Red River & 12th	Red River is a popular street with restaurants and bars near the central area. However, the exact TAZ that contains the intersection with 12th has a low trip count.

qualitative remarks are readily available:

- *Not all days are equal*, when considering whether we should include a time observation for each time of the week or just each day, we suspected that each day had slightly different dynamics. We see that in a typical Wednesday at 8 AM, most locations have a density close to the global mean, whereas, in a typical weekend morning, the locations in the central area (University, downtown, Red River & 12th) have higher productivity. Mondays and Fridays also have higher productivity than the rest of the business days.
- *Smoothing is turned on* with a few exceptions, the estimates of the locations we chose in central Austin (university, downtown, and Red River & 12th). This finding is particularly remarkable since Red River had almost no observations. It is also evident that regions without observations are not pull towards the global mean, since the region behaves very differently from it.
- *We recover periodic patterns.* In our analysis, we chose not to create edges between different days of the week at the same time of the day (e.g., there is no direct edge between a Monday 8 PM and Tuesday 8 PM). Nevertheless, we do observe periodic patterns for some locations, notably, the airport. Every morning around 8 PM, its distribution is close to the global mean. However, every afternoon it is shifted downwards.

To further evaluate the time smoothness of the estimates, we show in Figure 6 the results at the airport area in intervals of two hours. This completes the picture of the periodic pattern found in Figure 5, since it shows a smooth transition between mornings, when the distribution of productivity is closer to the global mean, to afternoons, when it is smaller. It is interesting to mention that while other locations showed a very different distribution during weekends and business days, the airport is more similar every day.

3.5.2 Investigating driver productivity

We present a list of interesting scientific inquiries that can be answered using the full distributions of productivity at each location and time.

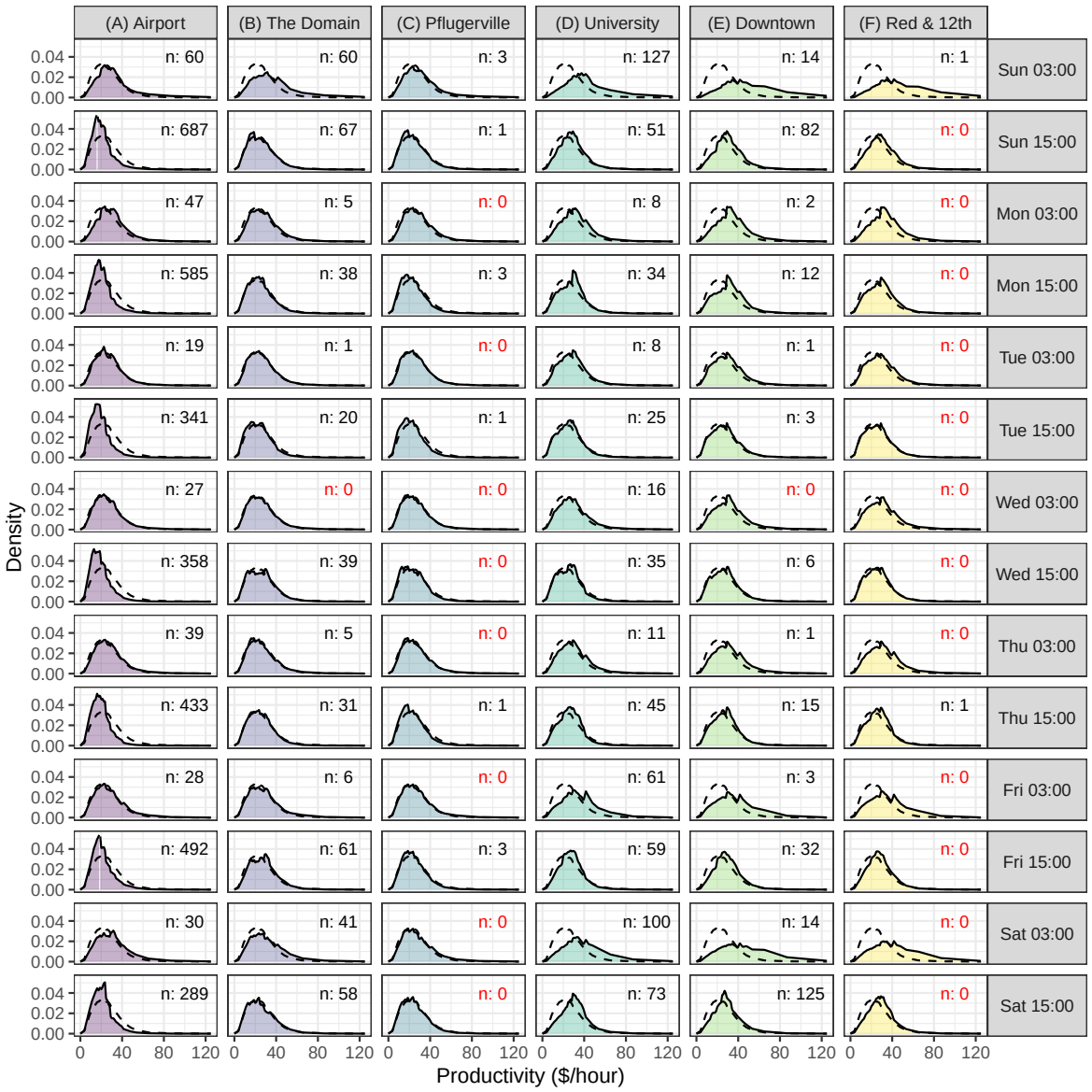


Figure 5: Driver Productivity by Time and Location. The global distribution is shown in dashes; the number of observed data points in the corresponding node of the graph is shown in the upright corner of each density plot (n). Time is shown every 12 hours.

1. *Tail probabilities*: what is the probability of not exceeding a specific salary? We compare to standardized living wages in Travis County, TX (Nadeau, 2017).

2. *Quantiles*: how many dollars per hour constitute the α-level quantile? We are interested in assessing the risk of the α% worst performers.

We provide answers to some of these questions in this section; the others are included in the supplemental material.

Tail probabilities: the risk of not attaining a living wage. We seek to estimate the probability that a driver will obtain a minimum living salary. Table 2 shows estimated living wages for families living in Austin in 2017 (Nadeau, 2017). To these wages, we must add the activity-specific additional costs such as the fixed fee of $0.99 charged per trip charged

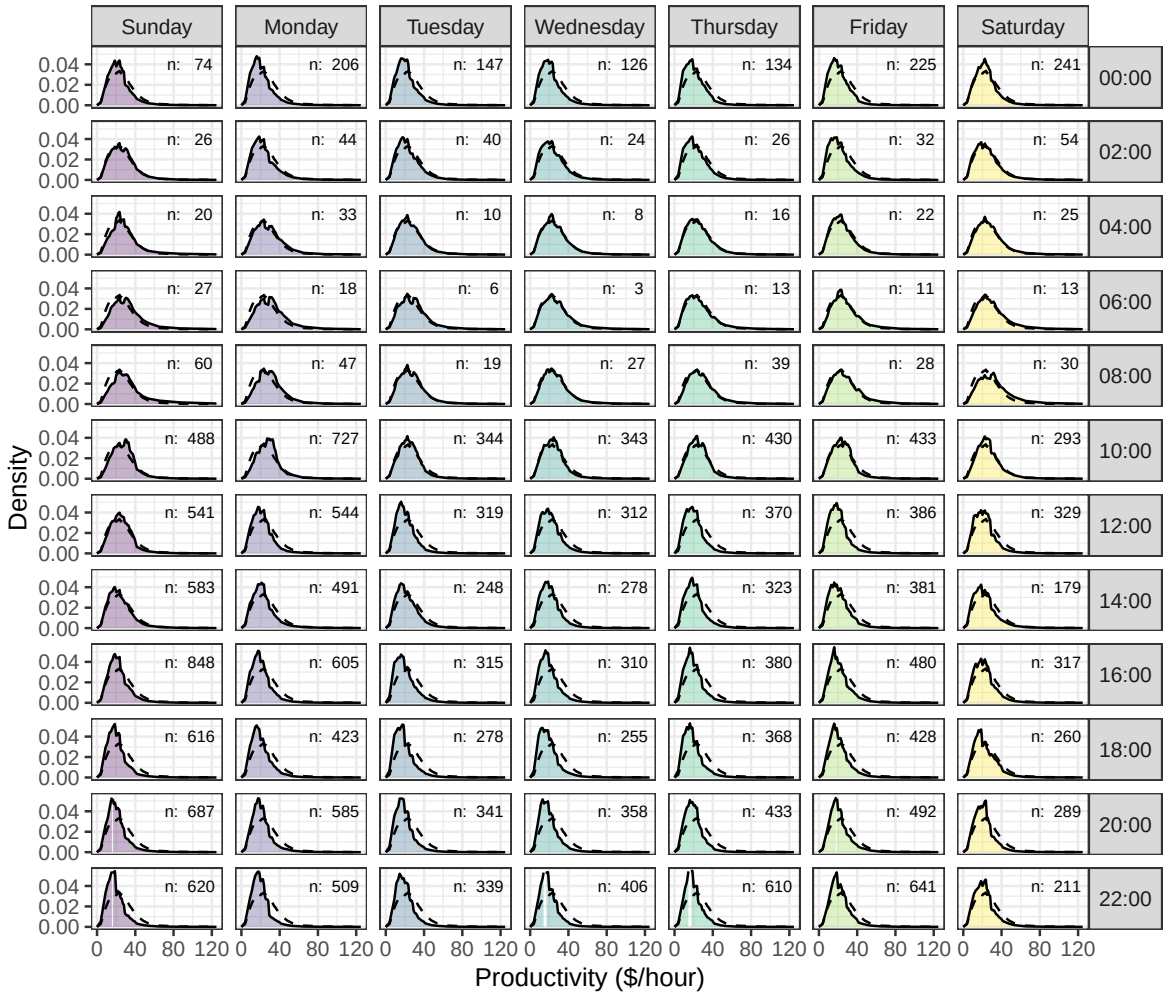


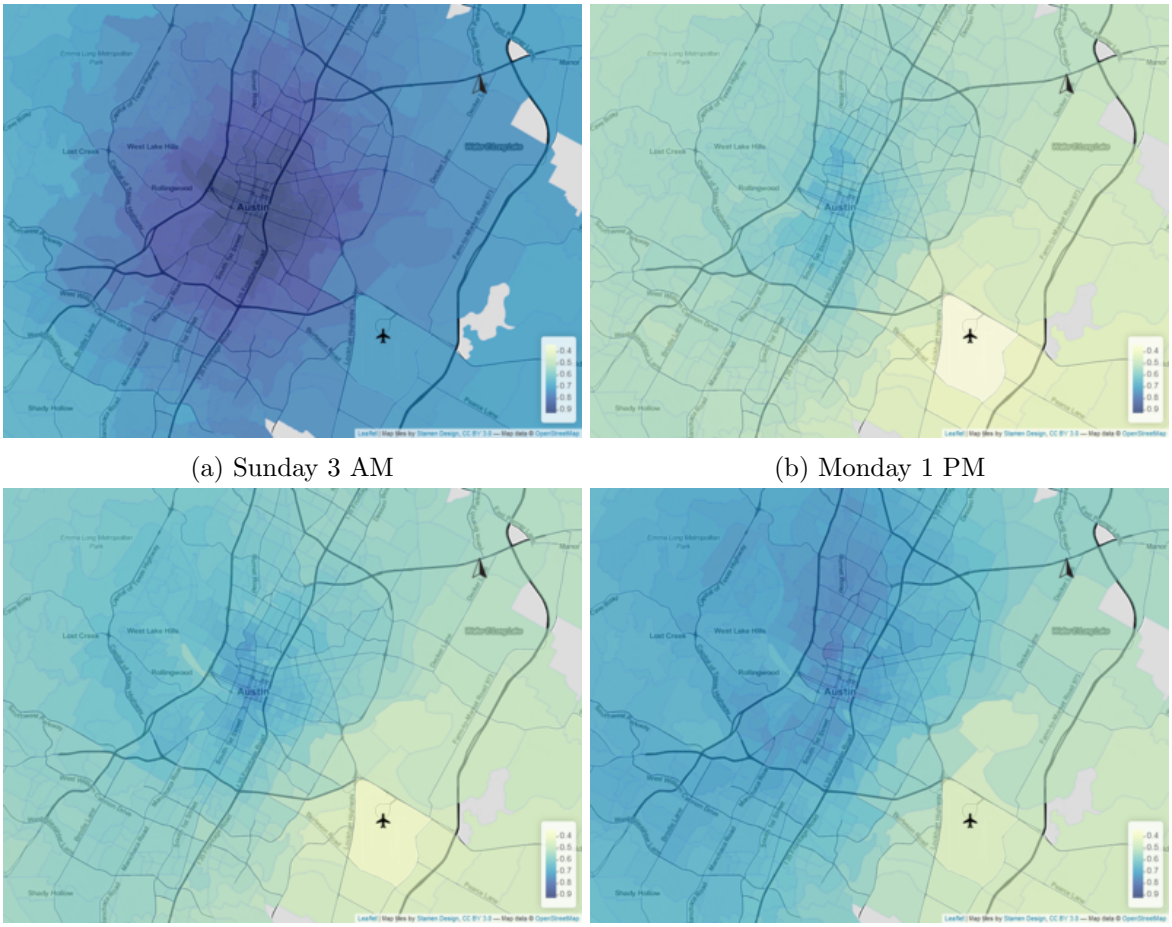
Figure 6: Driver Productivity of the Airport TAZ. The global distribution (dashed); the number of arriving flights during the observation period (a); the number of departing flights (d); the number of observations in the dataset (n). Time is shown every two hours for the seven days of the week.

by RideAustin as well as car maintenance, which on average lies around $6.40 hourly pretax and $4.78 hourly after tax deductions (Mishel, 2018; Hall and Krueger, 2016). Since a driver completes more than one trip per hour, we rounded up the costs to $6.00. The final reference values including costs are also presented in Table 2.

Table 2: Hourly living wages in Austin, TX (Nadeau, 2017). The costs are calculated using the $0.99 RideAustin fee per ride and an estimation of $4.78 hourly maintenance cost after tax deductions (Mishel, 2018).

# adults	1 adult	2 adults	2 adults (1 working)	1 adult
# children	0 children	2 children	2 children	2 children
living wage	$12.56	$15.64	$26.73	$28.74
living wage+costs	$18.56	$21.64	$32.73	$34.74

Figure 7 shows the results for the case of two working adults with two children ($21.64)



(a) Sunday 3 AM

(b) Monday 1 PM

(c) Friday 4 PM

(d) Saturday 6 PM

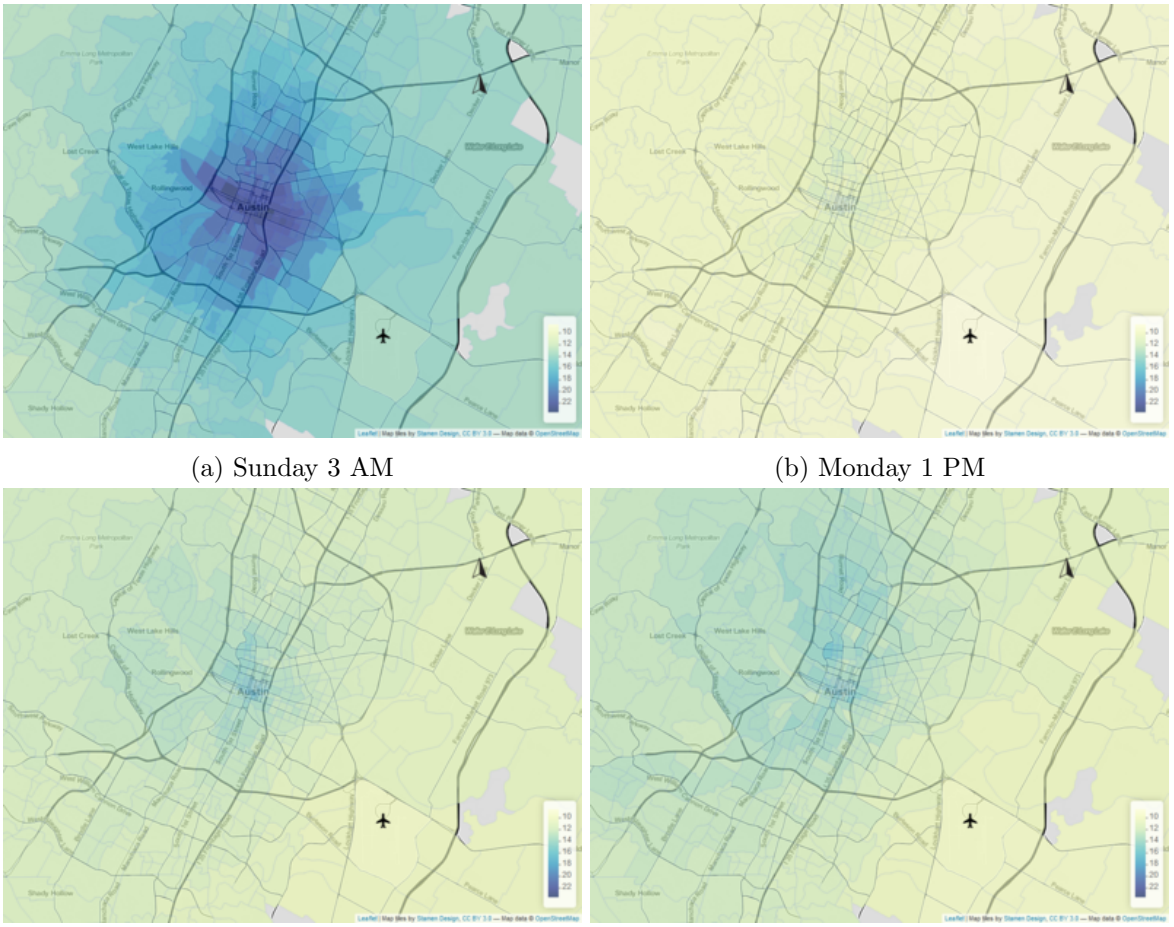
Figure 7: Probability of exceeding $21.64 in the next hour given a current location (living wage with costs for two working adults with two children).

for different times of the week and regions of the city. The remaining cases are included in Appendix D. We observe that during a Sunday morning when the traffic is low and there is a high demand (*c.f.* Figure 4d) the probability of exceeding the living wage is close to 90% near downtown, and it decreases to around 70% as a driver lies farther away from central Austin. In contrast, during Monday 6 PM, with moderate demand but high traffic, the probability ranges from 40% to 60%, being worst at the airport. These results suggest that drivers are at a high risk of not making a living wage.

Quantiles: How bad are the worst performers doing? As a company seeking to guarantee the well-being of its workers, it makes sense to target the population at specific levels of risk. One may ask, what is the expected income of the lowest $100\alpha\%$ percent? That is, for each location and time, we seek to find the quantity

$$q_\alpha := \min_{q \in [0,125]} P(\texttt{productivity} > q) > \alpha$$

Usual interesting values for α are $\{0.1, 0.25, 0.5, 0.75, 0.9\}$. The case $\alpha = 0.1$ is shown in Figure 8, the rest are included in the Appendix. We can see that in a typical Monday 6 PM, the rush hour, the lowest 10% quantile is around $12 to $15, being as low as $10 in the airport area. This result should be contrasted with Table 2, which states that a living wage of a single working



(a) Sunday 3 AM (b) Monday 1 PM

(c) Friday 4 PM (d) Saturday 6 PM

Figure 8: Lower 10% quantile of productivity for different times and locations.

adult with no children is above $18. The highest value is attained around the central part of the city during weekend mornings when there are low traffic and high demand (*c.f.* Figure 4d).

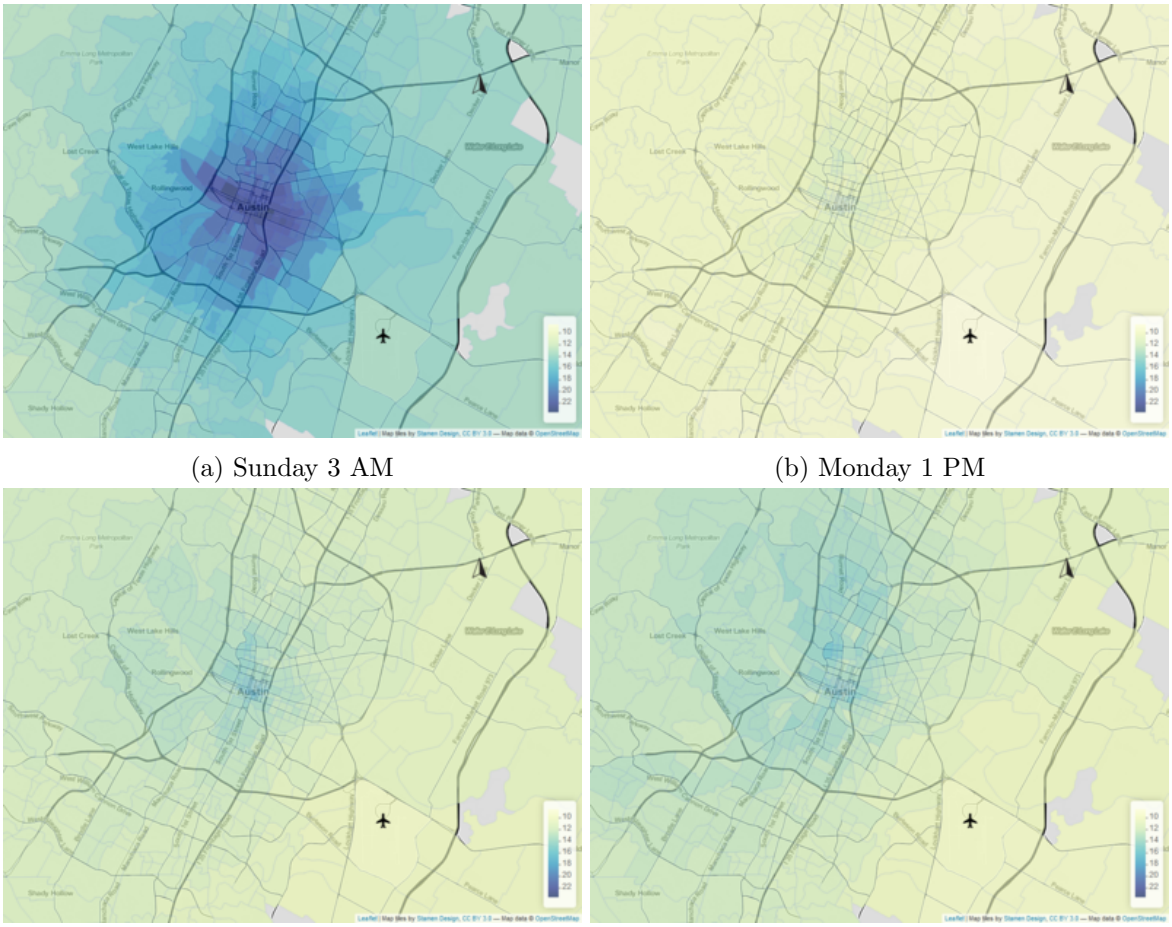
Given that we have quantiles, a natural measure of spread to consider is the inter-quartile range

$$\text{IQR} = q_{0.75} - q_{0.25}.$$

This quantity is preferred over standard deviation for skewed distributions, such as our measure of productivity. Figure 9 shows the IQR for different times and locations. It must be pointed out that this is a measure of spread and does not take into account the uncertainty arising from the estimation procedure, but only the variability in the estimated densities. This figure complements our previous inquiry using tail probabilities and quantiles in the sense that it shows that the highest reward observed Sunday 3 AM when there are high demand and low traffic, comes accompanied by higher variability, and not only a shift in location.

4 Conclusions

In this study, we presented a methodology for estimating the probability distribution of the productivity of a ride-sourcing driver as a function of space and time. We used information from more than 1.4 million trips in Austin, Texas, to provide a case study for the application of



(a) Sunday 3 AM (b) Monday 1 PM

(c) Friday 4 PM (d) Saturday 6 PM

Figure 9: IQR of productivity for different times and locations.

the proposed methodology. To the best of our knowledge, this article is the first academic study presenting a large-scale empirical analysis of spatiotemporal effects in the productivity of drivers of TNCs. Previous studies were either of theoretical nature, focused on mean spatial effects only and not spatiotemporal densities, or were applied to Taxi data and therefore not specifically tailored to TNC data. Our proposed methodology specifically focused on a spatio-temporal extension of the spatial density smoothing technique of Tansey et al. (2017), which is based on the Graph-fused Lasso (GFL). To do so, we focused on addressing two important challenges: first, enabling interpolation in regions with missing data; second, allowing for different effects in the spatial and temporal dimensions. Our proposed methodology is based on the Graph-fused Elastic Net (GFEN), which has an additional ℓ_2-total variation penalty for enabling interpolation. Furthermore, it has separate penalty parameters for the spatial and temporal dimensions that allow addressing both challenges while preserving many of the benefits of the GFL in terms of the behavior of the smoothing. We also presented an extended algorithm that allows training the GFEN at large scale.

The analysis suggests that the method offers several advantages for spatiotemporal evaluations. For example, its ability to interpolate enabled us to provide hourly estimates, even in locations and times with no observations. A model with high temporal resolution could help to detect periodic events, such as arrivals' peaks at the airport. Also, it can be used to detect the locations and periods with the lowest (or highest) productivity values. In addition, hav-

ing full density estimates enabled useful insights that could not be possible using mean effects only. For example, the estimation of the probability of not exceeding a specific salary (e.g., a living wage), and a value-at-risk calculation which answers the question of what is the salary threshold defining the worst performer drivers specified by a percentile. This methodology can help transportation engineers, policy-makers, and other ride-sourcing stakeholders to address the multiple challenges that trip-level information presents. The method can also be extended to other ride-sourcing metrics such as idle or deadheading time, driver reaching time, among others. Furthermore, the methodology can be extended to analyze metrics from different modes, such as public transit and taxis services.

A direction for future research is to include covariates in the analysis in order to obtain conditional density estimates. For example, it would be useful in our analysis to include weather conditions such as rain as a covariate. This extension can be easily implemented by replacing the binomial model in (2) with a logistic regression model or any binary prediction model. The drawback of this approach is that with the obvious strategy of smoothing each parameter separately, the complexity of the smoothing problem would increase proportionally to the complexity of the conditional estimation model. Therefore, an interesting area of research is to find an efficient smoothing framework for this task. Another limitation of our proposed model is that the addition of the GMRF penalty makes it more sensible to outliers. An interesting venue of research would be to replace the GMRF penalty with a Huber type of loss, which would encourage smoothness but would be robust to big outliers. To do this in a way that is compatible with the current algorithmic strategy for scalability, we would require a fast solver for the one-dimensional fused Huber problem. To the best of our knowledge, no such method exists.

References

Allcroft, D. J. and Glasbey, C. A. (2003), 'A latent gaussian markov random-field model for spatiotemporal rainfall disaggregation', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **52**(4), 487–498.

Ando, T. (2011), 'Predictive bayesian model selection', *American Journal of Mathematical and Management Sciences* **31**(1-2), 13–38.

Arias, P. and Morel, J.-M. (2018), 'Video denoising via empirical bayesian estimation of space-time patches', *Journal of Mathematical Imaging and Vision* **60**(1), 70–93.

Arlot, S., Celisse, A. et al. (2010), 'A survey of cross-validation procedures for model selection', *Statistics surveys* **4**, 40–79.

Barbero, A. and Sra, S. (2018), 'Modular proximal optimization for multidimensional total-variation regularization', *The Journal of Machine Learning Research* **19**(1), 2232–2313.

Bashtannyk, D. M. and Hyndman, R. J. (2001), 'Bandwidth selection for kernal conditional density estimation', *Computational Statistics and Data Analysis* **36**(3), 279–298.

Bentley, J. L. (1975), 'Multidimensional binary search trees used for associative searching', *Communications of the ACM* **18**(9), 509–517.

Besag, J. (1974), 'Spatial interaction and the statistical analysis of lattice systems', *Journal of the Royal Statistical Society B* **24**, 192–236.

Bian, B. (2018), 'Search frictions, network effects and spatial competition- taxis versus uber', *Working paper, Pennsylvania State University Department of Economics* .

Bimpikis, K., Candogan, O. and Daniela, S. (2016), 'Spatial pricing in ride-sharing networks', *Working paper, Stanford Graduate School of Business* .

Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J. et al. (2011), 'Distributed optimization and statistical learning via the alternating direction method of multipliers', *Foundations and Trends® in Machine learning* **3**(1), 1–122.

Brown, R. A. (2015), 'Building a balanced kd tree in o (kn log n) time', *Journal of Computer Graphics Techniques (JCGT)* **4**(1), 50–68.

Buchholz, N. (2015), Spatial equilibrium, search frictions and efficient regulation in the taxi industry, Technical report, Technical report, University of Texas at Austin.

Campbell, H. (2017), 'Why can't Uber drivers see the passenger's destination before accepting a trip?'.
URL: *https://maximumridesharingprofits.com/cant-uber-drivers-see-passengers-destination-accepting-trip/*

Castro, F., Besbes, O. and Lobel, I. (2018), 'Surge pricing and its spatial supply response', *Columbia Business School Research Paper No. 18-25* .
URL: *https://ssrn.com/abstract=3124571*

Chambolle, A. and Darbon, J. (2009), 'On total variation minimization and surface evolution using parametric maximum flows', *International journal of computer vision* **84**(3), 288.

Chen, J. and Tang, C.-K. (2007), Spatio-temporal markov random field for video denoising, *in* '2007 IEEE Conference on Computer Vision and Pattern Recognition', IEEE, pp. 1–8.

Cook, C., Diamond, R., Hall, J., List, J. A. and Oyer, P. (2018), The gender earnings gap in the gig economy: Evidence from over a million rideshare drivers, Technical report, National Bureau of Economic Research.

Cradeur, J. (2019), 'Big changes could be coming to Uber's destination filter'.
URL: *https://therideshareguy.com/changes-to-ubers-destination-filter/*

Cressie, N. (1993), *Statistics for Spatial Data*, J. Wiley.

Cressie, N. and Huang, H.-C. (1999), 'Classes of nonseparable, spatio-temporal stationary covariance functions', *Journal of the American Statistical Association* **94**(448), 1330–1339.

Cressie, N., Shi, T. and Kang, E. L. (2010), 'Fixed rank filtering for spatio-temporal data', *Journal of Computational and Graphical Statistics* **19**(3), 724–745.

Data World (2017), 'RideAustin Dataset'. [dataset].
URL: *https://data.world/ride-austin*

Finley, A. O., Banerjee, S. and Gelfand, A. E. (2012), 'Bayesian dynamic modeling for large space-time datasets using gaussian predictive processes', *Journal of geographical systems* **14**(1), 29–47.

Fryzlewicz, P. and Nason, G. (2002), A wavelet-fisz algorithm for poisson intensity estimation, Technical report, Department of Mathematics, University of Bristol.

Gelfand, A. E., Kottas, A. and MacEachern, S. N. (2005), 'Bayesian nonparametric spatial modeling with dirichlet process mixing', *Journal of the American Statistical Association* **100**(471), 1021–1035.
URL: *https://doi.org/10.1198/016214504000002078*

Getreuer, P. (2012), 'Rudin-osher-fatemi total variation denoising using split bregman', *Image Processing On Line* **2**, 74–95.

Hall, J. V. and Krueger, A. B. (2016), An analysis of the labor market for uber's driver-partners in the united states, Technical Report 22843, National Bureau of Economic Research.
URL: *http://www.nber.org/papers/w22843*

He, F., Wang, X., Lin, X. and Tang, X. (2018), 'Pricing and penalty/compensation strategies of a taxi-hailing platform', *Transportation Research Part C: Emerging Technologies* **86**, 263–279.

Hochbaum, D. S. (2001), 'An efficient algorithm for image segmentation, markov random fields and related problems', *Journal of the ACM (JACM)* **48**(4), 686–701.

Imrich, W. and Klavzar, S. (2000), *Product graphs: structure and recognition*, Wiley.

Jansen, M. (2006), 'Multiscale poisson data smoothing', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 27–48.

Johnson, N. A. (2013), 'A dynamic programming algorithm for the fused lasso and l 0-segmentation', *Journal of Computational and Graphical Statistics* **22**(2), 246–260.

Katzfuss, M. and Cressie, N. (2011), 'Spatio-temporal smoothing and em estimation for massive remote-sensing data sets', *Journal of Time Series Analysis* **32**(4), 430–446.

Katzfuss, M. and Cressie, N. (2012), 'Bayesian hierarchical spatio-temporal smoothing for very large datasets', *Environmetrics* **23**(1), 94–107.

Lavine, M. et al. (1992), 'Some aspects of polya tree distributions for statistical modelling', *The annals of statistics* **20**(3), 1222–1235.

Lavine, M. et al. (1994), 'More aspects of polya tree distributions for statistical modelling', *The Annals of Statistics* **22**(3), 1161–1176.

Li, P., Banerjee, S., Hanson, T. A. and McBean, A. M. (2015), 'Bayesian models for detecting difference boundaries in areal data', *Statistica Sinica* pp. 385–402.

Li, S., Tavafoghi, H., Poolla, K. and Varaiya, P. (2019), 'Regulating TNCs: Should Uber and Lyft Set Their Own Rules?', *arXiv e-prints* p. arXiv:1902.01076.

Lyft (2019), 'How to use destination mode'.
URL: *https://help.lyft.com/hc/en-us/articles/115013081128-How-to-use-Destination-Mode*

Ma, H., Fang, F. and Parkes, D. C. (2018), 'Spatio-temporal pricing for ridesharing platforms', *arXiv preprint arXiv:1801.04015* .

Mauldin, R. D., Sudderth, W. D., Williams, S. et al. (1992), 'Polya trees and random distributions', *The Annals of Statistics* **20**(3), 1203–1221.

Mishel, L. (2018), Uber drivers' compensation, wages, and the scale of uber and the gig economy, Working Paper 145552, Economic Policy Institute.
URL: *https://epi.org/145552*

Nadeau, C. A. (2017), Living wage calculator: User's guide, update 2017, Technical report, Open Data Nation. Online; accessed 1 March 2019.
URL: *http://livingwage.mit.edu/counties/48453*

Parisotto, S. and Schönlieb, C.-B. (2019), Total directional variation for video denoising, *in* 'International Conference on Scale Space and Variational Methods in Computer Vision', Springer, pp. 522–534.

Perea, C. (2017), 'Uber drops destination filters back to two trips per day'.
URL: *https://therideshareguy.com/uber-drops-destination-filters-back-to-2-trips-per-day/*

Ramdas, A. and Tibshirani, R. J. (2016), 'Fast and flexible admm algorithms for trend filtering', *Journal of Computational and Graphical Statistics* **25**(3), 839–858.
URL: *https://doi.org/10.1080/10618600.2015.1054033*

Reich, B. J. and Fuentes, M. (2007), 'A multivariate semiparametric bayesian spatial modeling framework for hurricane surface wind fields', *Ann. Appl. Stat.* **1**(1), 249–264.
URL: *https://doi.org/10.1214/07-AOAS108*

Rodríguez, A., Dunson, D. B. and Gelfand, A. E. (2010), 'Latent stick-breaking processes', *Journal of the American Statistical Association* **105**(490), 647–659.
URL: *https://doi.org/10.1198/jasa.2010.tm08241*

Romanyuk, G. (2017), Ignorance is strength: Improving the performance of matching markets by limiting information, Technical report, Working Paper, Harvard Univeristy, Cambridge, MA.

Rue, H. and Held, L. (2005), *Gaussian Markov random fields: theory and applications*, Chapman and Hall/CRC.

Rushworth, A., Lee, D. and Sarran, C. (2017), 'An adaptive spatiotemporal smoothing model for estimating trends and step changes in disease risk', *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **66**(1), 141–157.

Samuels, A. (2017), 'Uber, Lyft returning to Austin on Monday'.
URL: *https://www.texastribune.org/2017/05/25/uber-lyft-returning-austin-monday/*

Shaheen, S., Cohen, A. and Zohdy, I. (2016), Shared mobility: current practices and guiding principles, Technical Report FHWA-HOP-16-022, U.S. Depatment of Transportation.
URL: *https://ops.fhwa.dot.gov/publications/fhwahop16022/index.htm*

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. and de Freitas, N. (2016), 'Taking the human out of the loop: A review of bayesian optimization', *Proceedings of the IEEE* **104**(1), 148–175.

Smith, C. (2019), 'Amazing uber stats and facts (2019)'.
URL: *https://expandedramblings.com/index.php/uber-statistics/*

Snoek, J., Larochelle, H. and Adams, R. P. (2012), 'Practical Bayesian Optimization of Machine Learning Algorithms', *arXiv e-prints* p. arXiv:1206.2944.

Stroud, J. R., Müller, P. and Sansó, B. (2001), 'Dynamic models for spatiotemporal data', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **63**(4), 673–689.

Tansey, W., Athey, A., Reinhart, A. and Scott, J. G. (2017), 'Multiscale spatial density smoothing: An application to large-scale radiological survey and anomaly detection', *Journal of the American Statistical Association* **112**(519), 1047–1063.
URL: *https://doi.org/10.1080/01621459.2016.1276461*

Tansey, W. and Scott, J. G. (2015), 'A fast and flexible algorithm for the graph-fused lasso', *arXiv preprint arXiv:1505.06475* .

Tibshirani, R. J. and Taylor, J. (2011), 'The solution path of the generalized lasso', *Ann. Statist.* **39**(3), 1335–1371.
URL: *https://doi.org/10.1214/11-AOS878*

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005), 'Sparsity and smoothness via the fused lasso', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(1), 91–108.

Wang, Y.-X., Sharpnack, J., Smola, A. J. and Tibshirani, R. J. (2016), 'Trend filtering on graphs', *Journal of Machine Learning Research* **17**(105), 1–41.
URL: *http://jmlr.org/papers/v17/15-147.html*

Welch, G., Bishop, G. et al. (1995), 'An introduction to the kalman filter'.

Willett, R. M. and Nowak, R. D. (2007), 'Multiscale poisson intensity and density estimation', *IEEE Transactions on Information Theory* **53**(9), 3171–3187.

Wohlberg, B. (2017), 'Admm penalty parameter selection by residual balancing', *arXiv preprint arXiv:1704.06209* .

Xu, G., Liang, F. and Genton, M. G. (2015), 'A bayesian spatio-temporal geostatistical model with an auxiliary lattice for large datasets', *Statistica Sinica* pp. 61–79.

Yang, H., Fung, C., Wong, K. and Wong, S. C. (2010), 'Nonlinear pricing of taxi services', *Transportation Research Part A: Policy and Practice* **44**(5), 337–348.

Yasuda, M., Watanabe, J., Kataoka, S. and Tanaka, K. (2018), 'Linear-time algorithm in bayesian image denoising based on gaussian markov random field', *IEICE TRANSACTIONS on Information and Systems* **101**(6), 1629–1639.

Zha, L., Yin, Y. and Xu, Z. (2018), 'Geometric matching and spatial pricing in ride-sourcing markets', *Transportation Research Part C: Emerging Technologies* **92**, 58–75.

Zhao, L. and Hanson, T. E. (2011), 'Spatially dependent polya tree modeling for survival data', *Biometrics* **67**(2), 391–403.

Zou, H. and Hastie, T. (2005), 'Regularization and variable selection via the elastic net', *Journal of the royal statistical society: series B (statistical methodology)* **67**(2), 301–320.

Zuniga-Garcia, N., Tec, M., Scott, J. G., Ruiz-Juri, N. and Machemehl, R. B. (2019), 'Evaluation of Ride-Sourcing Search Frictions and Driver Productivity: A Spatial Denoising Approach', *arXiv e-prints* p. arXiv:1809.10329.

A Algorithmic details of the GFEN

For ease of presentation, in this section we will focus on the case of a general graph, which corresponds to the GFEN objective (9). The extension to the spatio-temporal case will be straightforward. In this case, it will be convenient to assume that $\mathcal{T}$ itself can be written as a disjoint union $\mathcal{T} = \mathcal{T}_S \cup \mathcal{T}_T$ where the trails of $\mathcal{T}_S$ and $\mathcal{T}_T$ consists of spatial and temporal edges respectively. This assumption is not strictly necessary; however, it simplifies computation since the total variation penalization hyperparameters will always be constant for a given trail.

The first step is to rewrite the GFEN objective as a constrained optimization problem using a set of slack variables $\mathbf{z}_{\tau,p} = (z^{(v)}_{\tau,p})_{v\in\tau}$ exactly one for each trail[7] $\tau \in \mathcal{T}$ and for each norm $p \in \{1, 2\}$, obtaining

$$\begin{aligned} \underset{\beta}{\text{minimize}} \quad & \sum_{v\in V} l(\mathbf{y}^{(v)}, \beta^{(v)}) + \sum_{\tau\in\mathcal{T}} \sum_{p\in\{1,2\}} \lambda_p \mathrm{TV}_p(\mathbf{z}_{\tau,p}, \tau) \\ \text{subject to} \quad & \mathbf{z}_{\tau,p} = \beta[\tau] \quad \text{for all} \quad \tau \in \mathcal{T}, p \in \{1, 2\} \end{aligned} \tag{12}$$

where $\beta[\tau] = (\beta^{(v)})_{v\in\tau}$. A direct application of the ADMM algorithm (Boyd et al., 2011) yields the following iterative updates:

$$\begin{aligned} \beta^{(v)}_{[k+1]} &= \underset{\beta}{\text{argmin}}\ l(\mathbf{y}^{(v)}, \beta) + \alpha \sum_{\{\tau:\, v\in\tau\}} \sum_{p\in\{1,2\}} (\beta - z^{(v)}_{\tau,p,[k]} + u^{(v)}_{\tau,p,[k]})^2 && \forall v \in V \\ \mathbf{z}_{\tau,1,[k+1]} &= \underset{\mathbf{z}}{\text{argmin}}\ \|\mathbf{z} - \beta_{[k+1]}[\tau] - \mathbf{u}_{\tau,1,[k]}\|^2 + \lambda_1 \mathrm{TV}_1(\mathbf{z}, \tau) && \forall \tau \in \mathcal{T} \\ \mathbf{z}_{\tau,1,[k+1]} &= \underset{\mathbf{z}}{\text{argmin}}\ \|\mathbf{z} - \beta_{[k+1]}[\tau] - \mathbf{u}_{\tau,2,[k]}\|^2 + \lambda_2 \mathrm{TV}_2(\mathbf{z}, \tau) && \forall \tau \in \mathcal{T} \\ \mathbf{u}_{t,p,[k+1]} &= \mathbf{u}_{t,p,[k]} + \beta_{[k+1]}[\tau] - \mathbf{z}_{t,p,[k+1]} && \forall p \in \{1, 2\} \quad \forall \tau \in \mathcal{T} \end{aligned} \tag{13}$$

where α is the scalar of the ADMM step-size parameter, and $\mathbf{u}_{t,p}$ are the ADMM dual variables. The algorithm depends on a random initialization of the parameters. Step 1 corresponds to a binomial negative log-likelihood model with a quadratic regularization. Following Tansey et al. (2017), we can substitute the full minimization in Step 1 with a single iteration of Newton's method. Step 2 corresponds to the fused lasso problem for chain graphs. We leverage available linear-time solvers such as (Johnson, 2013) and (Barbero and Sra, 2018), which have comparative performance. We prefer the latter since it can handle different values of λ_1 for each edge, although in the current formulation it is assumed to be constant. Step 3 can be solved in linear time using the Kalman smoothing algorithm (Welch et al., 1995, see). Finally, step 4 is the dual variable update of the ADMM algorithm in scaled form and does not require any sophisticated computation. There are different strategies to dynamically change the value α to accelerate convergence; empirically, we found that the method of Wohlberg (2017) worked best for our problem.

In Section 2.3, we argued that the smoothing step could be performed in an embarrassingly parallel for every node of the tree. The algorithm above offers additional possibilities for parallelism since steps 2-4 can be performed in parallel for each trail. In a high-performance computing environment, a useful strategy would be to distribute the smoothing problem for each tree node into different computation nodes using distributed memory parallelism. While steps 2-4 can be parallelized for each trail using multi-threading, i.e., shared-memory parallelism. Multi-threading will work better if the trail decomposition used has trails of balanced lengths; see (Tansey and Scott, 2015) for a discussion on trail decomposition strategies.

[7] With a slight abuse of notation, we say that $v \in \tau$ if v appears in some edge of τ.

B Hyperparameter Tuning with Bayesian Optimization

Hyperparameter tuning can be performed using in-sample and out-of-sample criteria. In-sample tuning for the graph-fused lasso is typically done using information criteria such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC) (Tibshirani et al., 2005). These methods rely on the fact the degrees of freedom are easy to compute in the graph-fused lasso since it reduces to counting the number of plateaus (Tansey and Scott, 2015). However, in the case of the GFEN, the ℓ_2-norm penalty adds smoothness and makes the approach of counting plateaus unfeasible. One solution that appears in the GMRF literature is to use an approximate information criterion such as the Deviance Information Criterion (DIC). However, that solution is known to favor over-fitted models and assumes an approximately normal distribution predictive distribution (Ando, 2011). Below we describe an alternative out-of-sample tuning approach based cross-validation, for a survey see Arlot et al. (2010).

The overall idea is that we can use k-cross-validation for the out-of-sample likelihood prediction. Given a set of hyperparameters λ for the model. We divide our data $\mathbf{y}$ into k equally sized testing sets or folds $\{\mathbf{y}^{\{j\}}\}_{j=1}^{k}$. For each fold j, we use the training data $\mathbf{y}^{\{j\}}_{\text{train}} = \bigcup_{j' \neq j} \mathbf{y}^{\{j'\}}$ to learn the parameters of a statistical model. We then use thos parameters compute the average of the out-of-sample negative likelihood, i.e. the out-of-sample loss, using the evaluation set $\mathbf{y}^{\{j\}}_{\text{test}} = \mathbf{y}^{\{j\}}$. Finally, the estimate out-of-sample loss corresponding to θ is the average over all test sets $\mathbf{y}^{\{j\}}_{\text{test}}$. To compute the out-of-sample loss in the context of our graph density smoothing model we would to the following: for each point $y_i \in \mathbf{y}^{\{j\}}_{\text{test}}$, we would identify the corresponding leaf node $B(y_i)$ to which y_i belongs and compute $-\log \hat{P}(y_i \in B(y_i) \mid \hat{\beta}^{\{j\}})$ using expression (2) and the relevant parameters for the vertex to which y_i belongs. Averaging over the losses of all the out-of-sample points for every fold we obtain the cross-validation estimate for the negative loglikelihood

$$\hat{l}_\lambda := \frac{1}{N} \sum_{j=1}^{k} \sum_{y_i \in \mathbf{y}^{\{j\}}_{\text{test}}} -\log \hat{P}(y_i \in B(y_i) \mid \hat{\beta}^{\{j\}}) \tag{14}$$

where N is the total number of data points, and λ is the vector of total variation penalization parameters. The hyperparameters with the lowest values of $\hat{l}_\lambda$ will be preferred.

Since for a spatio-temporal GFEN, we have to tune for four hyperparameters a grid-search strategy is not recommended. For example, if we were to try 10 different values for each hyperparameter we would then need to train $k \times 10^4 = 40,000$ models to select the best hyperparameters using grid-search! Instead, we use a Gaussian Process (Snoek et al., 2012) to guide the search. The assumptions of this approach are the following:

1. Suppose we have observed out-of-sample losses $\hat{l}_1, \ldots, \hat{l}_n$ corresponding to hyperparameters $\lambda_1, \ldots, \lambda_n$. The Gaussian Process assumption is that $\hat{l}_1, \ldots, \hat{l}_n$ follow a multivariate Gaussian distribution.
2. Moreover, the multivariate distribution is assumed to have the following form

   $$(\hat{l}_1, ..., \hat{l}_n) \sim \text{Normal}(0, K + \sigma^2 I),$$

   where $K := (K_{ij})_{i,j=1}^{n}$ is some Kernel matrix depending on $\lambda_1, \ldots, \lambda_n$ and σ^2 models the uncertainty in the observations $\hat{l}_j$. A typical example of kernel matrix K is the radial kernel $K_{ij} = \exp(-a\|\lambda_i - \lambda_j\|_2^2)$ where a controls the degree of correlation between similar hyperparameters.
3. Given a new point l_* corresponding to an untested hyperparameter λ_*, the fact that $(l_*, \hat{l}_1, \ldots, \hat{l}_n)$ is multivariate Gaussian can be used to easily compute the predictive mean

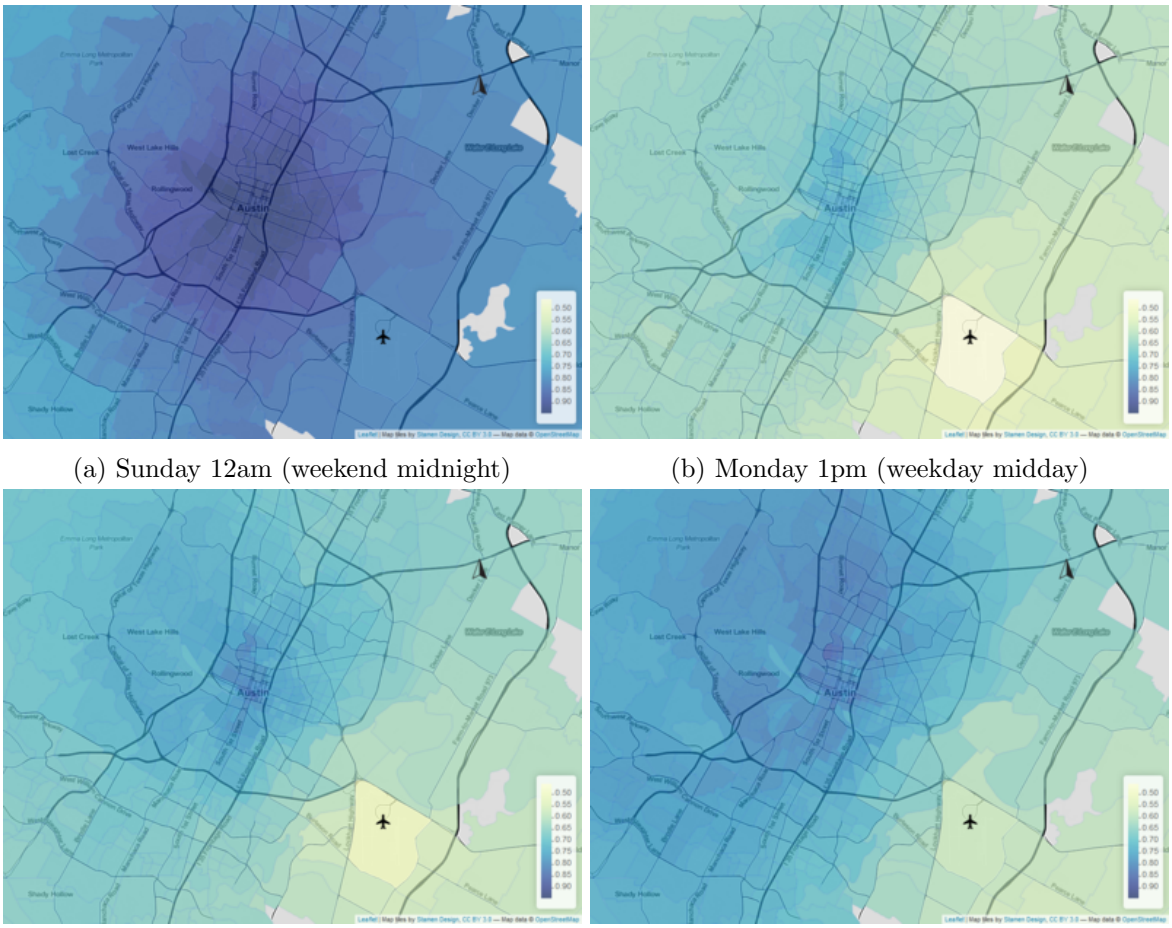
value of l_* given the observed $\hat{l}_1, \ldots, \hat{l}_n$. More precisely, given a series of candidate untested hyperparameters, we select the λ_* that has the lowest expected loss $E[l_* \mid \hat{l}_1, \ldots, \hat{l}_n]$.

The above steps give a high-level description of the idea of Bayesian optimization. For a detailed explanation, we refer the reader to the review by Shahriari et al. (2016). In Section 3 we will provide additional details about our implementation for the RideAustin dataset.

C Choosing a Tree Splitting Scheme

The estimated densities using a binary tree $B^{(K)}$ will assign a constant density to every point of a leaf node B_γ. Therefore, the quality, or more precisely, the resolution, will be limited by the choice of tree. Under infinite streams of data, the depth of the tree K can be increased until the leaf nodes B_γ are very small. However, with finite data, a bad partitioning scheme might lead to a poor resolution in regions of high data concentration and too much resolution on regions without likely values. To improve the construction, we suggest using a quantile method based on the global empirical distribution formed by aggregating the data from all the vertices. This is illustrated in Figure 4a, where we create quantile for the global distribution of the productivity variable in the RideAustin dataset. We will provide the details of the definition of the productivity variable in Section 3. For example, with a depth $K = 2$, the first splitting value would be the median, and then the bottom half would be split with the first quartile, and the top half would be split with the third quartile. More generally, for $\gamma \in \{0, 1\}^k$, we can define $B_\gamma = [q_a, q_b)$ where q_a and q_b are global distribution quantiles corresponding respectively to $a = \sum_{j=1}^{k} \gamma_j 2^{-j}$ and $b = a + 2^{-k}$. This approach to splitting the output space using a balanced binary tree is commonly applied in the literature of kd-trees, as in Bentley (1975), and Brown (2015).

D Tail probabilities of not exceeding living wage



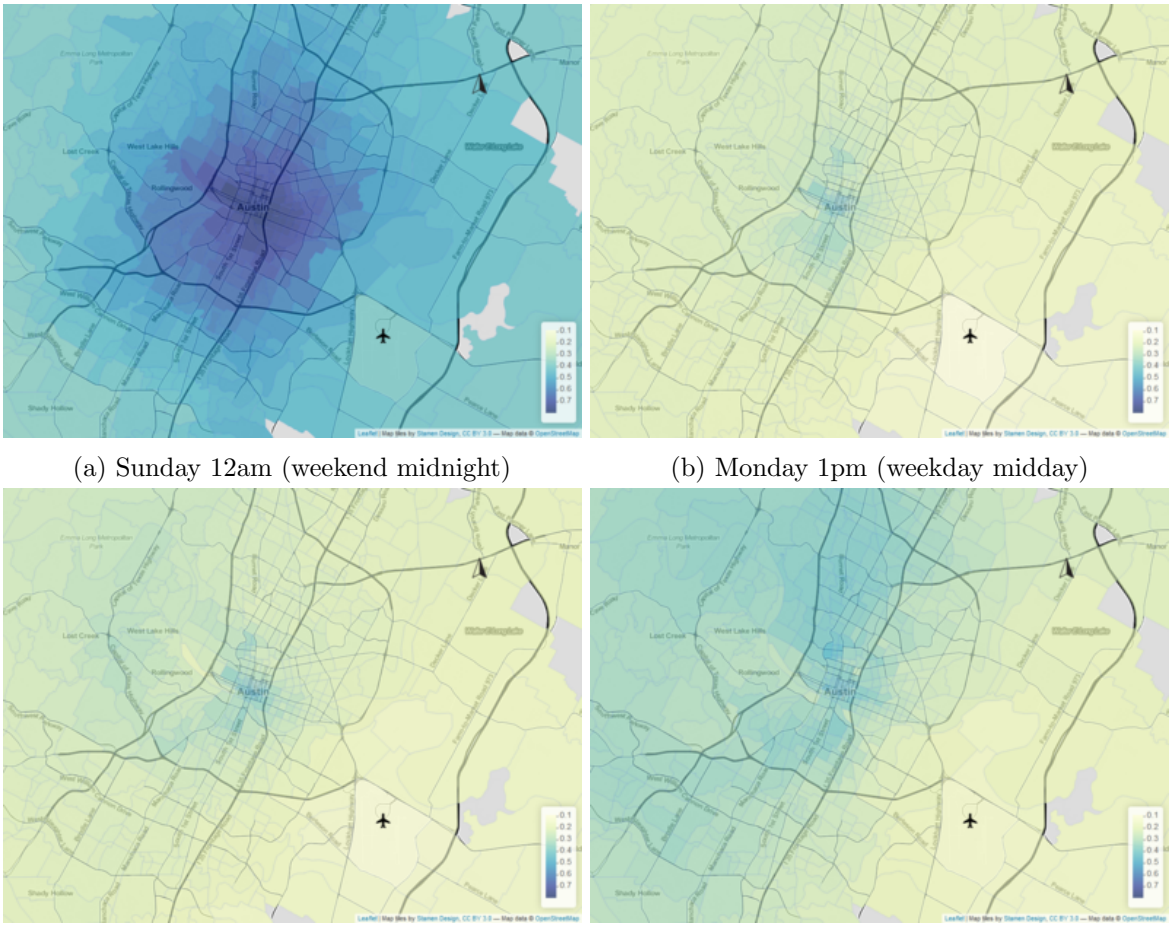
(a) Sunday 12am (weekend midnight)

(b) Monday 1pm (weekday midday)

(c) Friday 4pm (afternoon)

(d) Saturday 6pm (weekend night beginning)

Figure D.1: Probability of exceeding $18.56 in the next hour given a current location (living wage with costs for one single working adult with no children).



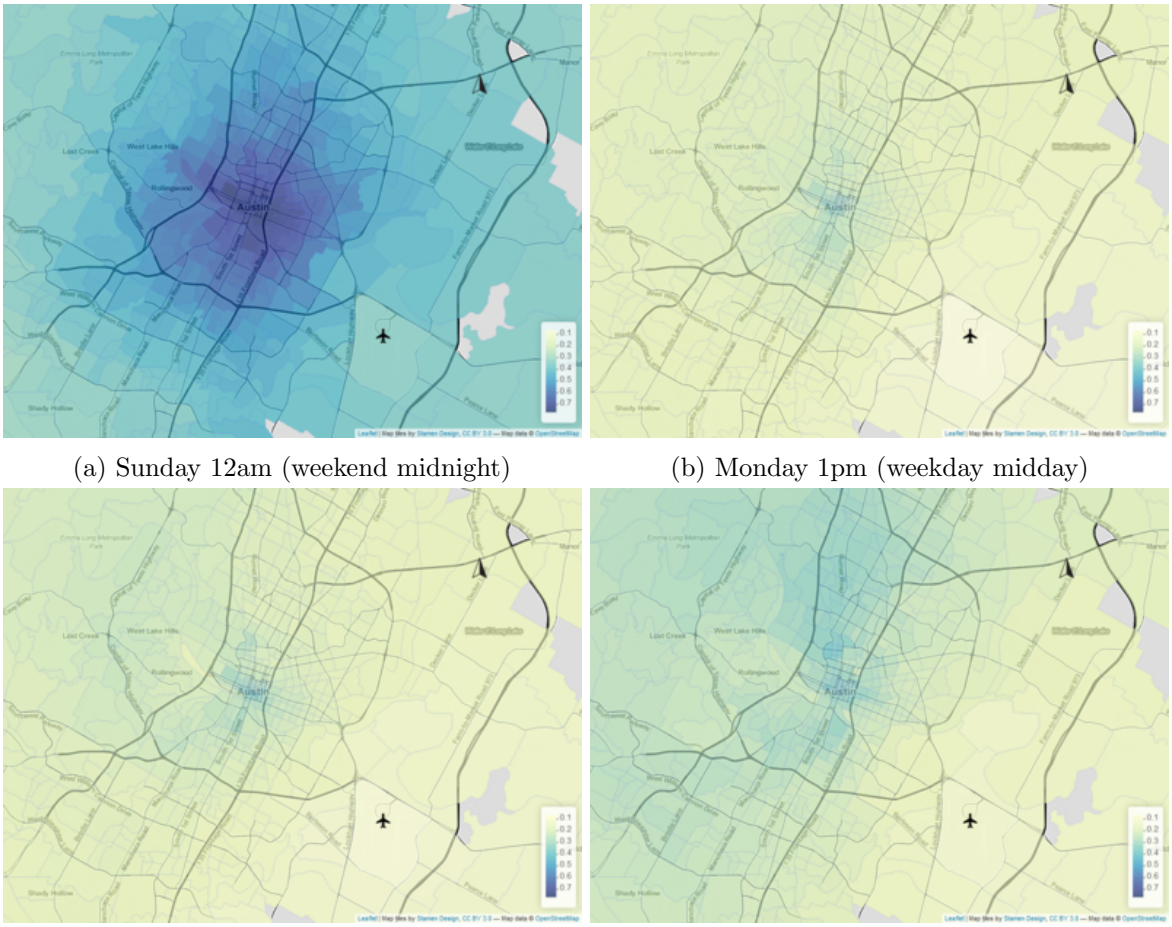
(a) Sunday 12am (weekend midnight)

(b) Monday 1pm (weekday midday)

(c) Friday 4pm (afternoon)

(d) Saturday 6pm (weekend night beginning)

Figure D.2: Probability of exceeding $32.73 in the next hour given a current location (living wage with costs for two adults, one working, and two children).



(a) Sunday 12am (weekend midnight)

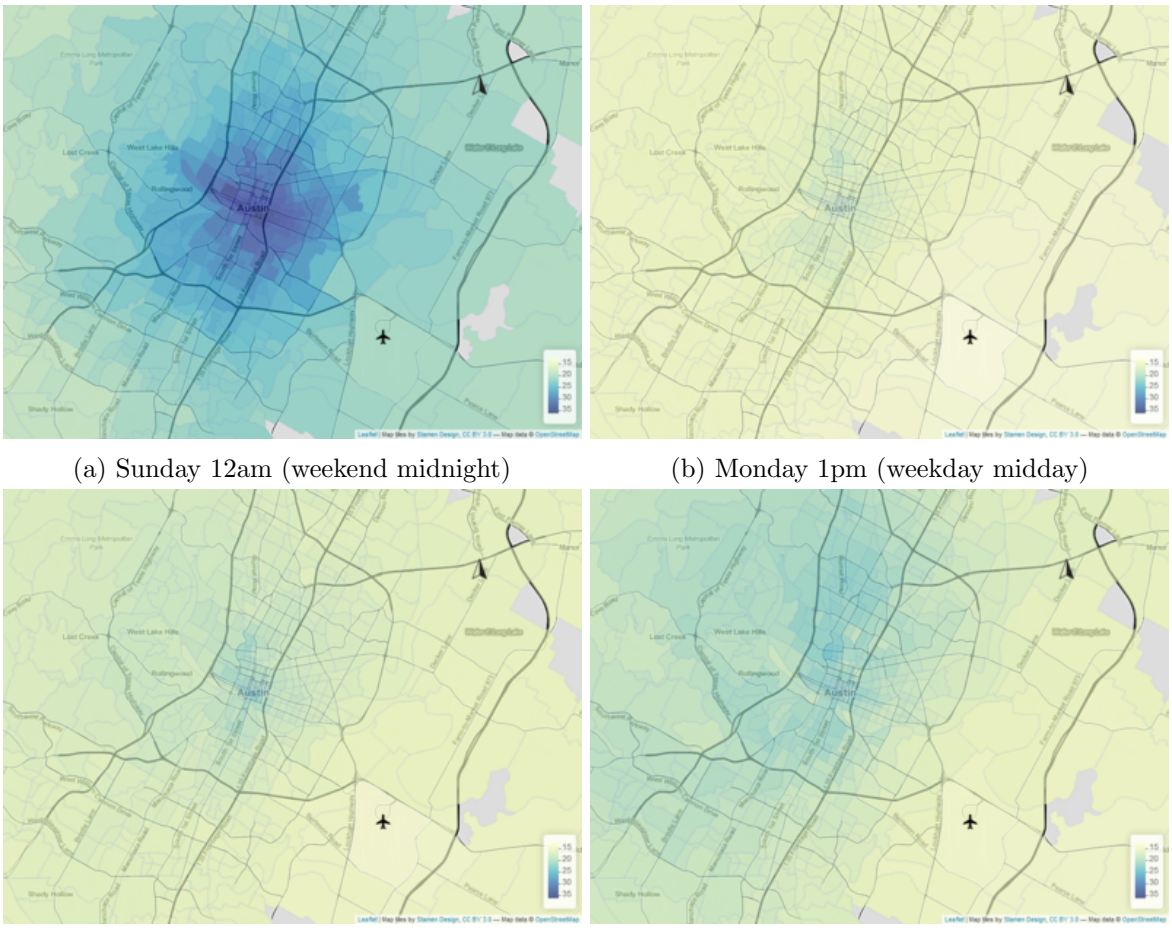
(b) Monday 1pm (weekday midday)

(c) Friday 4pm (afternoon)

(d) Saturday 6pm (weekend night beginning)

Figure D.3: Probability of exceeding $34.74 in the next hour given a current location (living wage with costs for one adult with two children).

E Quantiles



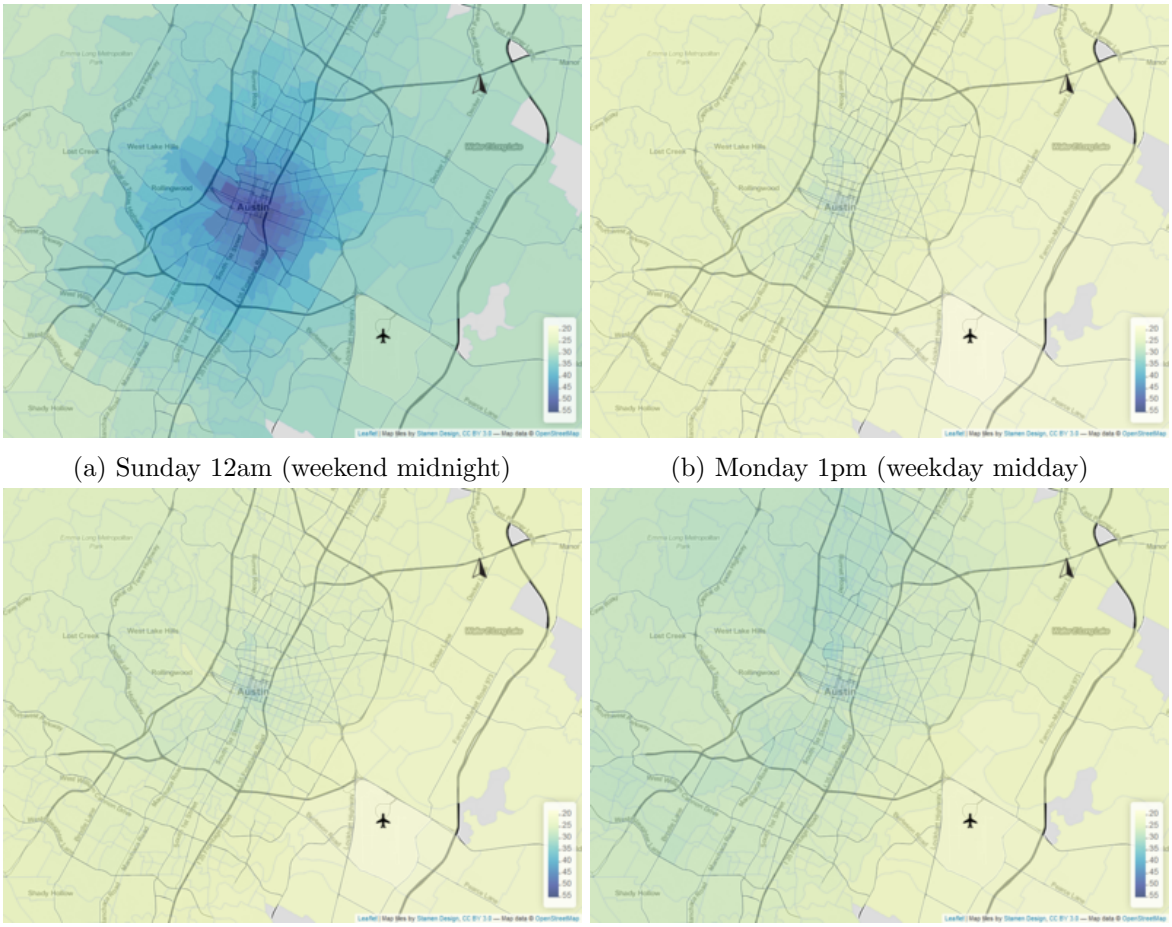
(a) Sunday 12am (weekend midnight)

(b) Monday 1pm (weekday midday)

(c) Friday 4pm (afternoon)

(d) Saturday 6pm (weekend night beginning)

Figure E.1: Lower 25% quantile of productivity for different times and locations.



(a) Sunday 12am (weekend midnight)

(b) Monday 1pm (weekday midday)

(c) Friday 4pm (afternoon)

(d) Saturday 6pm (weekend night beginning)

Figure E.2: Median of productivity for different times and locations.