PREDICTING MELBOURNE AMBULANCE DEMAND USING KERNEL WARPING[1]

By Zhengyi Zhou and David S. Matteson

Cornell University

Predicting ambulance demand accurately in fine resolutions in space and time is critical for ambulance fleet management and dynamic deployment. Typical challenges include data sparsity at high resolutions and the need to respect complex urban spatial domains. To provide spatial density predictions for ambulance demand in Melbourne, Australia, as it varies over hourly intervals, we propose a predictive spatio-temporal kernel warping method. To predict for each hour, we build a kernel density estimator on a sparse set of the most similar data from relevant past time periods (labeled data), but warp these kernels to a larger set of past data irregardless of time periods (point cloud). The point cloud represents the spatial structure and geographical characteristics of Melbourne, including complex boundaries, road networks and neighborhoods. Borrowing from manifold learning, kernel warping is performed through a graph Laplacian of the point cloud and can be interpreted as a regularization toward, and a prior imposed for, spatial features. Kernel bandwidth and degree of warping are efficiently estimated via cross-validation, and can be made time- and/or location-specific. Our proposed model gives significantly more accurate predictions compared to a current industry practice, an unwarped kernel density estimation and a time-varying Gaussian mixture model.

1. Introduction. A primary goal of emergency medical services (EMS) is to minimize response times to life-threatening emergencies while keeping operational costs low. Accurate spatial-temporal ambulance demand predictions are crucial to optimal operations management of base location, staff, fleet and deployment. These demand predictions are ideally needed at high temporal and spatial granularities. The industry typically predicts for every hour and every 1 km^2 region. We are motivated to predict this demand for the city of Melbourne, Australia.

There are several typical challenges to predicting ambulance demand:

- Ambulance demand is often exceedingly sparse at the temporal and spatial resolution required for prediction. There is zero demand in the vast majority of 1-km^2 regions over a 1-hour period.
- This demand arises from complex urban geography. The city boundary is often highly irregular. Ambulance demand can be very high (coastal and downtown)
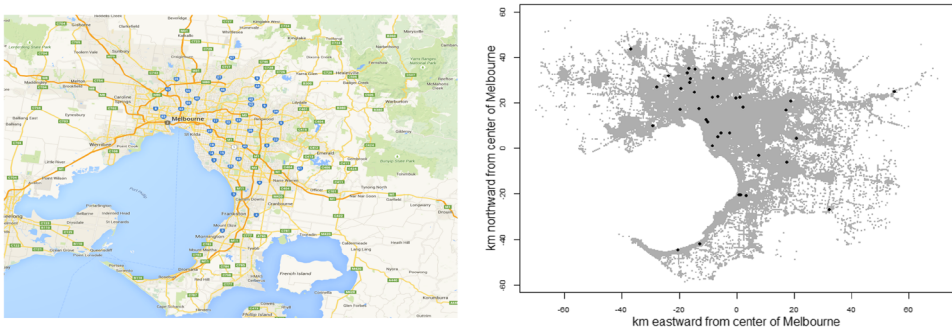
FIG. 1. *Left*: *map of Melbourne* [*Google Maps* (2015)]; *right*: *spatial locations of all* 696,975 *Melbourne ambulance demand incidents from years* 2011–2012 (*in gray*), *and* 38 *demand incidents for a typical* 1-*hour period* (*in black*). *We observe complex boundary and geographical features* (*e.g.*, *highways*, *roads*, *satellite suburbs*).

or very low (suburbs) along the boundary. Within this boundary, demand follows closely the city's infrastructure and terrain; there might be high demand along central highways and zero demand within an internal lake.

- Ambulance demand exhibits spatial and temporal patterns. Weekly seasonality is usually prominent [Channouf et al. (2007), Matteson et al. (2011)]; the industry relies heavily on this seasonality to make predictions. Some studies have also noted daily seasonality and short-term serial dependence at densely populated regions [Zhou et al. (2015)].
- Ambulance demand data for large cities is often large scale. This presents computational challenges, especially since predictions are needed very frequently.

It is particularly difficult to simultaneously resolve these challenges. Overcoming sparsity requires considerable smoothing, while capturing complex spatiotemporal patterns requires fine-resolution modeling. At high granularities, data sparsity makes it difficult to detect spatio-temporal characteristics accurately. At low granularities, differences across regions and times are not sufficiently captured for optimal ambulance planning.

Figure 1 demonstrates these challenges in predicting ambulance demand for Melbourne. We show on the right the locations of 696,975 demand incidents from years 2011 and 2012 (in gray), and those of 38 demand incidents for a typical 1-hour period (in black). On average, 99.6% of the 1-km^2 regions in Melbourne receive zeros calls in any given hour. Comparing to a map of Melbourne on the left [Google Maps (2015)], we observe a highly complex spatial boundary as Melbourne encloses a large bay to its southwest. Demand is high near the bay, but low on the outskirt suburbs. Demand is visibly higher at small satellite suburban neighborhoods and along major highways radiating out from the city center. There is lack of demand due to several reservoirs and a national park to the west and northwest. Consistent with typical patterns, the demand exhibits strong weekly seasonality.

The EMS industry and previous studies have attempted to address some of these challenges. The current industry practice uses a simple averaging formula. Demand in a 1-km^2 spatial region over an hour is typically predicted by averaging a small number of historical counts, from the same spatial region, over the corresponding hours from previous weeks or years [Goldberg (2004)]. For instance, the EMS of Charlotte-Mecklenburg, North Carolina, use a method called MEDIC, in which the prediction is the average of twenty corresponding counts in the same hour of the preceding four weeks for the past five years [Setzler, Saydam and Park (2009)]. Averaging so few historical counts, which are mostly zeros, produces noisy and flickering predictions, resulting in haphazard and inefficient deployment.

Much attention has been given to predicting the aggregate ambulance demand as a temporal process using autoregressive moving average models [Channouf et al. (2007)], factor models [Matteson et al. (2011)] and spectral analysis [Vile et al. (2012)]. Few studies have modeled spatio-temporal ambulance demand well. [Setzler, Saydam and Park (2009)] use artificial neural networks, but fail to improve over the industry method. A recent study by Zhou et al. (2015) predicts ambulance demand for Toronto, Canada, using a time-varying Gaussian mixture model (GMM). This method is more accurate than the industry practice, but, as the authors point out, extending it to incorporate spatial boundaries would be prohibitively expensive. While it may not be essential for Toronto since the city is almost rectangular in shape, it becomes important for Melbourne; it is difficult for ellipsoidal Gaussian components to model demand well on the highly complex spatial domain of Melbourne. Another study by Zhou and Matteson (2015) considers a spatio-temporal weighted kernel density estimation (KDE) to predict Toronto's ambulance demand. It gives similarly accurate predictions as GMM (both much better than industry), showing promise for KDE.

KDE has been used to analyze spatio-temporal data such as crime incidence [Nakaya and Yano (2010)], disease spread [Zhang et al. (2011)] and data streams [Aggarwal (2003)]. It allows for rapid identification of "hotspots" and their evolutions in time and space. However, implementing a naive KDE is not satisfactory for our application. The chosen bandwidth necessarily has to be quite large given the data sparsity, smoothing inappropriately across boundary features and disregarding the underlying urban geography.

Few studies have focused on modeling spatial or spatio-temporal point processes on complex spatial structures. Most studies assume a boundary defined a priori (polygon or pixelated). If not, *ad hoc* methods based on the convex hull of all observed points are typically used [Ripley and Rasson (1977)]. This invariably results in a convex boundary that may be inaccurate where data is sparse. Even with a boundary optimally defined, few methods are equipped to handle complex boundary features. Ramsay (2002) proposes a finite window smoother with known boundary conditions computed using an expensive finite element approach. Building on that, Wood, Bravington and Hedley (2008) model the boundary condition

as a loop of wire and the point process as a soap film suspended from the boundary wire. They represent this smoother as a penalized basis, compute it via multigrid, and select smoothness via generalized cross-validation. They acknowledge the lack of an elegant solution when the boundary conditions are unknown. Apart from boundary, other spatial characteristics, such as neighborhood structures and road networks, are rarely incorporated in modeling. Woodworth et al. (2014) use high-resolution housing data and satellite images to inform the support of density estimates for residential burglary, and obtain improved spatial-only density estimates then without using external spatial information. We propose a method that can efficiently capture and exploit a wide range of spatial characteristics with or without using external spatial information. We can infer spatial characteristics from the point process data when external spatial information is not readily available or directly relevant. We apply our method to perform spatio-temporal prediction on sparse data.

We draw from theory and methods developed in manifold learning. Manifold learning, a branch of machine learning, is concerned with estimating and exploiting the underlying structures of data. The assumption is that data in a high-dimensional space resides on or near a lower-dimensional sub-manifold. In practice, we do not have access to this sub-manifold, but we can approximate it from a point cloud, that is, a mass of historical data. The most common method is to construct an adjacency graph of this point cloud and make use of the properties and structures of this graph. This idea has led to many popular learning methods, including isomap [Tenebaum, de Silva and Langford (2000)], local linear embedding [Roweis and Saul (2000)], Hessian eigenmaps [Donoho and Grimes (2005)] and Laplacian eigenmaps [Belkin and Niyogi (2003)] [see van der Maaten, Postma and van den Herik (2009) for some review]. These methods were initially designed for data representation or visualization, but have been adapted for semi-supervised classification [Belkin and Niyogi (2004), Zhu et al. (2005), Zhou et al. (2003)] and clustering [Ng, Jordan and Weiss (2001), Shi and Malik (2000)].

In particular, a variant of Laplacian eigenmaps, kernel warping, has been proposed for semi-supervised classification [Belkin and Niyogi (2004), Sindhwani, Niyogi and Belkin (2005), Smola and Kondor (2003)]. Using a small number of labeled data and a larger number of labeled and unlabeled point cloud data, the method classifies new examples by constructing kernels on the labeled data that warp to the geometry of the point cloud. This geometry is represented by the adjacency graph of the point cloud. Smoothing orthogonal to this geometry is penalized heavily, whereas smoothing along this geometry is not. This method is designed for high-dimensional classification, and has good performance on text and image data.

Drawing from this idea, we propose a novel method for modeling spatio-temporal point processes against complex spatial structures and features. To predict ambulance demand for a future time period, we have a sparse set of historical data that is very relevant for this prediction (labeled data). We fit a KDE on them,

but warp the kernels to a larger set of historical data regardless of their relevance to this predictive task (point cloud). This point cloud describes our belief about the spatial structure on which the labeled data lies. It captures exterior and interior boundaries without needing to explicitly define boundaries and boundary conditions. It also incorporates a wide range of complex spatial similarities and discontinuities, such as roads, city blocks and neighborhoods of varying shapes and densities. Intuitively, this warping can be thought of as a regularization that penalizes radical departure from and encourages flow of information along our intuition of the geography. In a Bayesian sense, it can also be thought of as imposing a prior based on how similar or different the point process is across different locations. Such a regularization or prior is especially beneficial when the labeled data is sparse. We select the kernel bandwidth and the degree of warping efficiently via cross-validation. Both of these parameters can be made time- and/or location-specific.

We implement this method on ambulance demand data from Melbourne in years 2011 and 2012. Altogether there are 696,975 realized events. Each event contains the time and location that the ambulance was dispatched to. The proposed kernel warping model gives significantly more accurate predictions than previous approaches, including the MEDIC method as an industry practice, unwarped KDE and GMM.

We develop the kernel warping model in Section 2. We construct an unwarped KDE in Section 2.1, warp the kernels to the point cloud in Section 2.2, and allow for time- and location-specific warping in Section 2.3 for the Melbourne data. Some details on computation are included in Section 2.4. We show the empirical results for predicting Melbourne ambulance demand in Section 3, and conclude in Section 4.

2. Model. We model Melbourne's ambulance demand on a continuous spatial domain $\mathcal{S} \subseteq \mathbb{R}^2$ and a discretized temporal domain of one-hour intervals $\mathcal{T} = \{1, 2, \ldots\}$. Let $\mathbf{s}_{t,i}$ be the location of the ith ambulance demand arising from the tth time period, for $i \in \{1, \ldots, n_t\}$, where n_t is the total number of ambulances demanded in the tth period. Since a nonhomogeneous Poisson process (NHPP) is a natural model for a spatial point process [Diggle (2003), Møller and Waagepetersen (2004)], we assume $\{\mathbf{s}_{t,i} : i = 1, \ldots, n_t\}$ for each time period t independently follow an NHPP over $\mathcal{S}$, with positive intensity function γ_t. We decompose the intensity function as $\gamma_t(\mathbf{s}) = \delta_t f_t(\mathbf{s})$ for $\mathbf{s} \in \mathcal{S}$. Here, $\delta_t = \int_{\mathcal{S}} \gamma_t(\mathbf{s}) \, d\mathbf{s}$ is the aggregate demand intensity over the spatial domain, and $f_t(\cdot)$ is the continuous spatial *density* of the demand at time t such that $f_t(\mathbf{s}) > 0$ and $\int_{\mathcal{S}} f_t(\mathbf{s}) \, d\mathbf{s} = 1$. Therefore, for each t, $n_t | \gamma_t \sim \text{Poisson}(\delta_t)$ and $\mathbf{s}_{t,i} | \gamma_t, n_t \overset{\text{i.i.d.}}{\sim} f_t(\cdot)$ for $i \in \{1, \ldots, n_t\}$. The usual practice is to model $\{\delta_t\}$ and $\{f_t\}$ separately. As mentioned before, numerous studies have proposed sophisticated and accurate methods for estimating $\{\delta_t\}$. We thus focus on predicting the spatio-temporal demand density $\{f_t\}$, which is more challenging and less studied.

2.1. *Spatio-temporal KDE.* Suppose we want to predict Melbourne's ambulance demand for a future 1-hour period u. Given the prominent weekly seasonality, the most relevant observations are from the corresponding hour of the week for the past M weeks. They constitute the labeled data for this predictive task. This approach is aligned with the industry practice, and is shown to work well in Zhou et al. (2015). We choose the sliding window width M a priori. With a larger M, we have more training data, but each training is slower and less adaptive to recent changes in demand patterns (e.g., summer vs. winter). The industry and recent studies have considered M between 4 and 8. For Melbourne, we set $M = 8$, resulting in an average labeled data size of about 300 points (ranging from 100 to 450 for different periods). Let $\mathcal{T}_u = \{u - 168m : m \in \{1, \ldots, M\}\}$ denote the set of labeled time periods, in which 168 is the number of 1-hour periods in a week.

Starting with a simple KDE on the labeled data, we predict, for any $\mathbf{x} \in \mathcal{S}$,

$$(2.1) \qquad f_u(\mathbf{x}) = \frac{1}{\sum_{t \in \mathcal{T}_u} n_t} \sum_{t \in \mathcal{T}_u} \sum_{i=1}^{n_t} k(\mathbf{x}|\mathbf{s}_{t,i}, \boldsymbol{H}).$$

Here, k is the chosen bivariate spatial kernel with bandwidth matrix $\boldsymbol{H}$. We use the bivariate Gaussian kernel centered at the label data with a covariance matrix of $\boldsymbol{H}$, but the Epanechnikov kernel with bounded support could be used to provide computational savings. We choose bandwidth $\boldsymbol{H}$ via the plug-in method [Wand and Jones (1994)] or smoothed cross-validation [Duong and Hazelton (2005)]. When data show large variations in density, using one fixed bandwidth may not be optimal [Cacoullos (1966), Scott (1992)]. A bandwidth too large wipes out local features where we have sufficient data; a bandwidth too small leads to spurious peaks where data is sparse. In the case of Melbourne, data density varies substantially in space (downtown vs. neighborhoods) and time (midnight vs. rush hours); we may be motivated to consider a spatial- and/or time-varying $\boldsymbol{H}$.

2.2. *Kernel warping.* We would like to warp each kernel k in equation (2.1) to a larger set of point cloud data that describes the spatial boundary and characteristics of Melbourne. We choose the point cloud data, construct an adjacency graph on the point cloud, define the graph Laplacian matrix, and warp the kernel to this Laplacian matrix. We discuss in detail each step.

Step 1 [*Choosing the point cloud*]: Typically in Laplacian eigenmap and kernel warping applications, all labeled and unlabeled data are used as the point cloud. In the context of spatial statistics and our application, there are several points of consideration:

(a) *Which points?* We consider all observations in the near past, regardless of the time period. If we use the same sliding window width of $M = 8$ previous weeks, we are choosing from about 50,000 points.

(b) *How many points?* There is a trade-off: using more points in the cloud leads to better approximation of the geography but slower computation. Since we are in a low-dimensional space of $\mathbb{R}^2$, we may not need a very large number of points to depict the most salient boundary and spatial structures. In our application, we find 1000 spatial points represent Melbourne's geography reasonably well.

(c) *Points or mesh?* Alternative to using past observations, we can also use past data to define a pixelated spatial domain of Melbourne and use the included pixels as the point cloud. Doing so, we lose some resolution and information on data density, but may gain computationally if it can reduce the number of point cloud data significantly. A regularly spaced point cloud also induces a sparse, band-diagonal graph Laplacian matrix (to be discussed later), leading to further savings.

(d) *Representing roads?* If roads are particularly important in the application, we can select road intersections or midpoints of road segments for all or a subset of roads as the point cloud.

(e) *Global or local?* We can have one global point cloud for the entire spatial domain. We can also discretize the spatial domain into several regions with separate local point clouds. Local point clouds can provide computational advantages if they are smaller. They may also offer accuracy advantages if they depict finer-grain characteristics or allow for customized degree of warping at each locale. We discuss this further in Section 2.3.

In our application, we randomly sample 1000 historical observations as the point cloud for each "component" (to be explained in Section 2.3). We denote the set of point cloud data as $\{z_i\}$ for $i \in \{1, \ldots, Z\}$. See Figure 2(a) for an example cloud of 1000 points over the entire city of Melbourne. For our application, we find that predictive accuracy is not sensitive to the random sampling of the point cloud data. If it were, a larger point cloud might be needed, or predictions might be repeated and averaged over several point cloud samples.
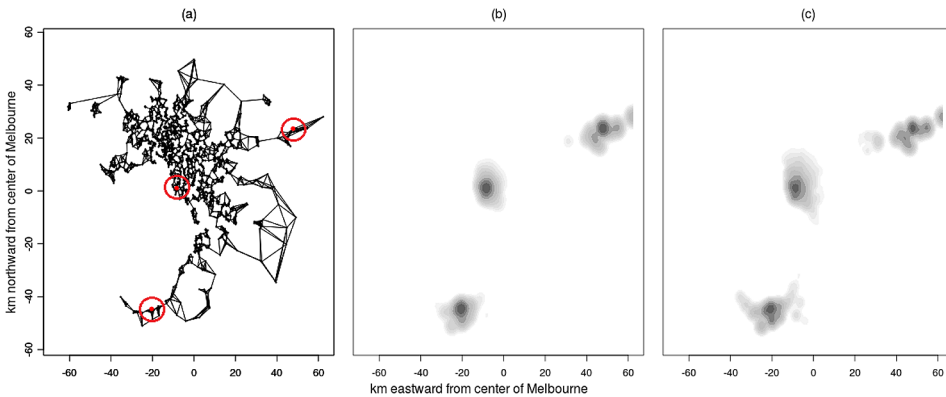

FIG. 2. *Examples of kernel warping*: (a) *the adjacency graph of a sample point cloud of size* 1000; *three observations are circled*; (b) *and* (c), *warped kernels centered at the three observations with degrees of deformation* $\lambda = 0.5$ *and* 2, *respectively.*

Step 2 [*Constructing the adjacency graph*]: We construct a graph with nodes at each point in the point cloud and edges connecting points that are close. We represent this graph using a symmetric, positive semidefinite adjacency matrix A.

(a) *Which nodes to connect?* Knowledge about the spatial domain (e.g., road connectivity, inside a building vs. outside) or regularity of the point cloud (e.g., regular mesh) may inform a natural way to define how nodes should be connected. Without such knowledge, we can connect nodes z_i and z_j if z_i is among the n nearest neighbors of z_j *or* z_j is among the n nearest neighbors of z_i (symmetric relation). This requires us to choose n. In our experience, n should be big enough to ensure that the point cloud is sufficiently connected instead of being very fragmented, but small enough to emphasize local relationships. A second way is to connect nodes if the (Euclidean) distance between them is smaller than a threshold. This requires us to choose the threshold.

(b) *Weighted edges?* In the simplest case, we can set $A_{ij} = 1$ if nodes z_i and z_j are connected and 0 otherwise. Another idea suggested in Belkin and Niyogi (2003) is to define weighted edges depending on the distance between points, that is, $A_{ij} = \exp\{-\|z_i - z_j\|^2/r\}$ if z_i and z_j are connected and 0 otherwise. The authors note that they do not have a principled way of choosing r; we find it reasonable to choose r empirically by fitting an exponential distribution on all distances between connected nodes. They also note that in practice a binary adjacency graph works well, and we agree.

In the Melbourne application, we use $n = 5$ nearest neighbors and binary weights to construct A. Figure 2(a) shows the adjacency graph of a sample point cloud of size 1000. Again, we find our predictive accuracy to be insensitive toward any reasonable variations in these choices.

Step 3 [*Constructing the Laplacian matrix*]: The graph Laplacian matrix L is defined to be $L = D - A$, in which D is the diagonal degree matrix, with its diagonal entries being the column (or, equivalently, row) sum of A, that is, $D_{ii} = \sum_j A_{ij}$. L is a symmetric, positive semidefinite matrix. If the graph has multiple connected components, L can be rearranged into a block diagonal matrix, where each block is the respective Laplacian matrix for each connected component. See Merris (1994) for an introduction of the Laplacian matrix and its important properties.

Here is the intuition of the Laplacian matrix. The (discrete) point cloud adjacency graph is an empirical approximation to our target (continuous) manifold of Melbourne geography. The (discrete) graph Laplacian matrix L is then an approximation to the (continuous) Laplace–Beltrami operator on this manifold. The Laplace–Beltrami operator is a manifold generalization of the Laplace operator, which is a linear second order differential operator on functions (in our case, kernels). See more discussion on the Laplace–Beltrami operator in Belkin and Niyogi (2005). This L induces a semi-norm on kernels which penalizes changes between adjacent nodes. There is a close analogy to heat flow; the heat (partial differential) equation has a Laplace operator in space. Intuitively, L guides how information

(heat) spreads on the spatial structure (manifold approximated by graph) from any initial KDE (initial heat distribution).

Step 4 [*Warping the kernels*]: We warp each kernel k from equation (2.1) to the point cloud to generate a new warped kernel $\tilde{k}$. For any $\mathbf{x} \in \mathcal{S}$ and any $\mathbf{s}$ in the set of labeled data,

$$(2.2) \qquad \tilde{k}(\mathbf{x}, \mathbf{s}|\boldsymbol{H}) = k(\mathbf{x}, \mathbf{s}|\boldsymbol{H}) - \boldsymbol{k}_\mathbf{x}^T (\boldsymbol{I} + \lambda L \boldsymbol{K})^{-1} \lambda L \boldsymbol{k}_\mathbf{s},$$

in which $\boldsymbol{k}_\mathbf{x} = [k(\mathbf{x}, \mathbf{z}_1|\boldsymbol{H}), \dots, k(\mathbf{x}, \mathbf{z}_Z|\boldsymbol{H})]$ and $\boldsymbol{k}_\mathbf{s} = [k(\mathbf{s}, \mathbf{z}_1|\boldsymbol{H}), \dots, k(\mathbf{s}, \mathbf{z}_Z|\boldsymbol{H})]$ are vectors of kernels evaluated at $\mathbf{x}$ or $\mathbf{s}$ and the point cloud data $\{\mathbf{z}_i\}$. Matrix $\boldsymbol{K} = [k(\mathbf{z}_i, \mathbf{z}_j|\boldsymbol{H})]_{i,j \in \{1,\dots,Z\}}$ is a symmetric matrix of kernels evaluated at all pairs of point cloud data, and $\boldsymbol{I}$ is a Z by Z identity matrix. The parameter $\lambda > 0$ represents the degree of deformation. When $\lambda = 0$, we have $\tilde{k} = k$. When $\lambda \to \infty$, $\tilde{k}$ approaches a positive constant on the point cloud (steady state heat distribution).

Equation (2.2) is obtained by warping the Reproducing Kernel Hilbert Space (RKHS) associated with the chosen kernel. We modify the RKHS with a point-cloud semi-norm λL. This deforms the kernel k along a finite-dimensional subspace given by the point cloud data. The modified RKHS is shown to be another RKHS, that is, $\tilde{k}$ is a properly defined kernel. See Sindhwani, Niyogi and Belkin (2005) and Belkin, Niyogi and Sindhwani (2006) for more details (they use the point cloud semi-norm of λL^p; we consider the simplified case where $p = 1$).

There are three interpretations of this type of kernel warping. The first is that of heat flow as mentioned before. We allow information (heat) to spread along the graph of the point cloud (approximately the manifold of Melbourne's geography). The second interpretation is a graph regularizer. Variations between adjacent nodes in the graph are penalized, and thus violations of the spatial structure implied by the point cloud are penalized. See Smola and Kondor (2003) and Bousquet, Chapelle and Hein (2005) for more discussion. Last, in the Bayesian framework, kernel warping can informally be thought of as imposing a data-dependent informative prior to describe our belief of the data geometry.

We replace the regular Gaussian kernel k in equation (2.1) with the new warped kernel $\tilde{k}$ defined in (2.2) to predict the density of ambulance demand f_u at a future time period u. We set a priori the sliding window width M, the point cloud data type/size, the number of nearest neighbors n and the weights used to construct the Laplacian matrix. We estimate the Gaussian kernel bandwidth $\boldsymbol{H}$ and the degree of deformation λ.

We show in Figures 2(b) and (c) examples of warping kernels. Three kernels of bandwidth $\boldsymbol{H} = \text{diag}(2, 2)$ are placed on three observations circled in Figure 2(a). They are warped toward the point cloud in (a) with degree of deformation $\lambda = 0.5$ (b) and 2 (c). With a larger λ, the kernels conform to the spatial boundary and characteristics to a greater extent.

2.3. *Spatio-temporal kernel warping.* Melbourne's ambulance demand shows substantial density variations with patterns in time (midnight vs. rush hour) and in space (downtown vs. neighborhoods). It may be beneficial to allow bandwidth H and degree of deformation λ to vary with time and space. Ideally, we would like to find, in time and space, pockets of the point process with similar characteristics, and apply similar smoothing and deformation.

We discretize time according to our modeling aims, that is, into 1-hour time periods. For each hour, we further discretize the spatial domain into a small number of regions, as motivated by the behavior of labeled data for that time period. We call each subregion of each hour a *component*, and perform estimations and predictions independently on each component. The spatial discretization splits a global point cloud into local ones, cuts all edges connecting across regions, and decomposes the Laplacian matrix into blocks. Labeled data are also matched into components. We estimate a separate set of H and λ for each component by cross-validation (details in Section 2.4).

We discretize spatially by clustering. For any given future time period, we cluster on its labeled data (about 300 observations). We allow different numbers of clusters and clustering configurations for each time period. In our application, this gives more accurate predictions than imposing a universal clustering configuration across time. We also obtain better results by clustering on labeled data rather than clustering on the point cloud data (the point cloud is much more similar across time than the labeled data). In the case of Melbourne, spatial characteristics across time are different enough that the gain in personalized modeling exceeds the loss in stablization offered by a common arrangement.

We choose to cluster using K-means based on Euclidean distance. K-means is fast, clusters all points and gives even clusters. Even cluster sizes are desirable because a very small cluster does not provide enough labeled data to reliably estimate parameters via cross-validation. To avoid this, we set a threshold minimum number of points in any cluster. We set the threshold at 15 points, which in practice limits the number of clusters to be below 8. If we fail to clear this threshold, we lower the number of clusters. Density-based clustering algorithms such as DBSCAN [Ester et al. (1996)] and shared nearest neighbors [Ertöz, Steinbach and Kumar (2003)] do not classify all points, and do not allow easy specification of the number of clusters. Graph-based clustering algorithms such as affinity propagation [Frey and Dueck (2007)] and spectral clustering [Ng, Jordan and Weiss (2001)] do not cluster on Euclidean distance and may be less intuitive for spatial point patterns. In our case, hierarchical clustering gives very uneven cluster sizes.

For each time period, we binary search for the best number of clusters based on validation likelihood. Increasing the number of clusters leads to, on the one hand, an additional 1000 points to the cloud and the flexibility to customize parameters locally, but, on the other hand, sparser labeled data for each cluster and reduced stability in parameter estimation. It is an empirical question for each time period whether we have enough labeled data to afford this increase in complexity. For

Melbourne, we find the number of clusters to be largely proportional to the size of labeled data.

2.4. *Computation*.   We estimate the kernel bandwidth H and the degree of deformation λ for each spatio-temporal component. To reduce the dimensionality, we parametrize H to be a scalar multiple of the plug-in bandwidth H_{pi} obtained if we fit an unwarped KDE for the same component; that is, we define $H = \alpha H_{pi}$, and estimate α. Alternatively, we can define a radial bandwidth $H = \mathrm{diag}(\beta, \beta)$, reducing the Gaussian kernel to a radial basis function. We use $H = \alpha H_{pi}$ because this parametrization gives slightly better performance in our preliminary analysis. To estimate a full H is more difficult because H needs to be positive semidefinite.

We choose H and λ for each component using 5-fold cross-validation. We partition the training data into 5 equally sized subsamples. For each candidate parameter value, we jointly fit the model on 4 subsamples and validate data likelihood on the remaining subsample. The process is repeated 5 times with each subsample used exactly once for validation, and we compute the average of the 5 validation likelihoods. To find the parameter values that maximize this average validation likelihood, we implement a surrogate, derivative-free optimization procedure called the stochastic radial basis function (RBF) method [Regis and Shoemaker (2007, 2009)]. It is a fast algorithm for global optimization of computationally expensive objective functions. Each iteration builds an the RBF model to approximate the expensive function, selects subsequent candidate points and evaluates them in parallel. We choose this approach because our objective function (likelihood) evaluation is not instantaneous. It takes between 0.5 and 4 seconds, depending on the sizes of the labeled data and point cloud (Python code on a personal computer). We also do not have simple derivative computations. In our experience, 100 such evaluations are sufficient to provide a good optimum, competitive to those found by grid search, pattern search or evolutionary algorithms. However, a wide range of optimization tools can be applied here.

In our application, we find a typical optimal α to be between 0.05 and 0.3. We need a concentration of heat which is then spread or warped to the point cloud. A typical optimal λ is between 0 and 2. Most time periods choose between 1 and 3 spatial components. We warm start the binary search for the number of clusters based on the size of labeled data. The best configuration is usually found within 3 searches.

Given the prominent weekly seasonality, we believe that the corresponding parameter values are also similar from week to week. In fact, we believe that the nature of deformation and smoothing does not vary significantly over several months, and thus only estimate the parameters for a one-week cycle once every few months. With the most recent weekly set of parameter values, we predict forward in an online fashion with a sliding window of $M = 8$ weeks, making use of the most recent 8 weeks of data available. Each prediction is instantaneous.

The most expensive part of the computation is evaluating kernels between all pairs of point cloud data and taking the inverse of a large matrix. Several local point clouds of reasonable sizes (<2000) are computationally more efficient than one massive global point cloud. There are ways to optimize this computation, including using right division instead of inversion, saving precomputed kernel evaluation matrices and vectors, exploiting the sparse, banded-diagonal Laplacian matrix, using a tree-based algorithm for fast KDE computation [Gray and Moore (2003)], and using a lookup table for Gaussian densities. The computation is also "embarrassingly parallelizable," across validation likelihood evaluations and across spatio-temporal components. Without using most of the above-mentioned optimizations and using a point cloud size of 1000, it typically takes a small number of hours to train on 8 weeks of data and predict for 4 weeks (672 hourly periods) on a personal computer (Python).

3. Predicting ambulance demand for Melbourne. We would like to predict ambulance demand in Melbourne for every 1-hour period in March 2011. There are two stages to this computation. In the first stage, we estimate all parameters for a weekly cycle. The parameters include the spatial clustering configuration for each 1-hour period, as well as the parameters λ (degree of warping) and α (in bandwidth $\boldsymbol{H} = \alpha \boldsymbol{H}_{pi}$) for each spatial component in each 1-hour period. This estimation only needs to be performed very infrequently, and, in our case, once. For this estimation, we use Melbourne ambulance demand data from 8 weeks in January and February 2011. In the second stage, we use the estimated weekly set of parameter values to predict future ambulance demand on a sliding window of 8 weeks for each 1-hour period in March 2011.

Figure 3 shows the predictive density estimated by kernel warping for two time periods on March 2, 2011 (Wednesday). We have only about 150 labeled data to predict for 2–3 am (a), and cross-validate to use only 1 spatial component. We have almost 400 labeled data for 2–3 pm (b) and cross-validate to choose 5 spatial components.

We consider two variations in estimation: (i) spatio-temporal kernel warping (S-T param), in which we separately estimate parameters for each 1-hour period and spatial region (via clustering, Section 2.3), and (ii) temporal kernel warping (T param), in which we separately estimate parameters for each 1-hour period (no spatial clustering). We show in Figure 4 the predictive densities produced by these two approaches for the same time period. The densities look similar, with slightly more details when we use spatio-temporal kernel warping (we cross-validate to select 3 spatial clusters).

We compare the proposed kernel warping models to the following:

(a) The MEDIC method, which is an industry practice implemented in Charlotte-Mecklenburg, NC (Section 1): We implement this method as far as we have data. The cell count in a 1-km^2 region and a 1-hour period is predicted by the average of corresponding cell counts in the preceding 8 weeks.
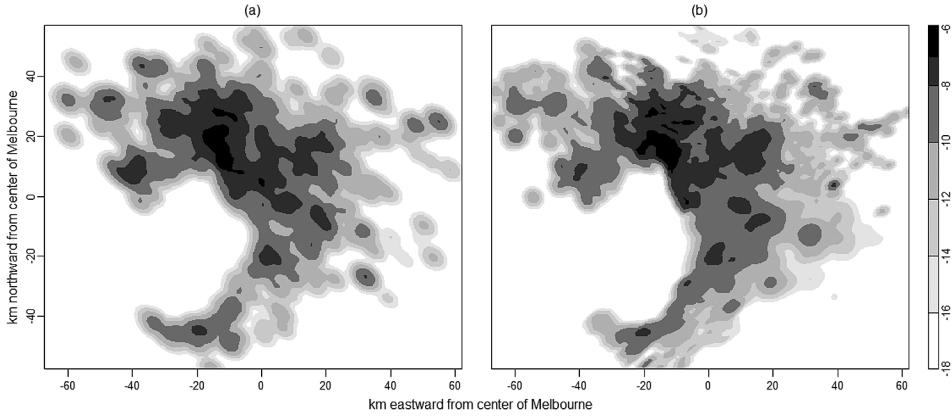
FIG. 3. *Log predictive densities using spatio-temporal kernel warping for March* 2, 2011 (*Wednesday*) *at* (a) 2–3 *am* (*night*), *and* (b) 2–3 *pm* (*day*). *For time period* (a), *we have sparse data and cross-validate to choose* 1 *spatial component. For time period* (b), *we have more data and choose* 5 *spatial components.*

(b) Unwarped KDE, as in equation (2.1): The bandwidth H is chosen via the plug-in method (PI) [Wand and Jones (1994)] and smoothed cross-validation (SCV) [Duong and Hazelton (2005)]. This H is separately estimated for each time period, but does not vary in space.

(c) Gaussian mixture model (GMM) [Zhou et al. (2015)], in which the means and covariances of Gaussian components are fixed through time, and the mixture weights vary in time: We also use labeled data from the last 8 weeks, and consider
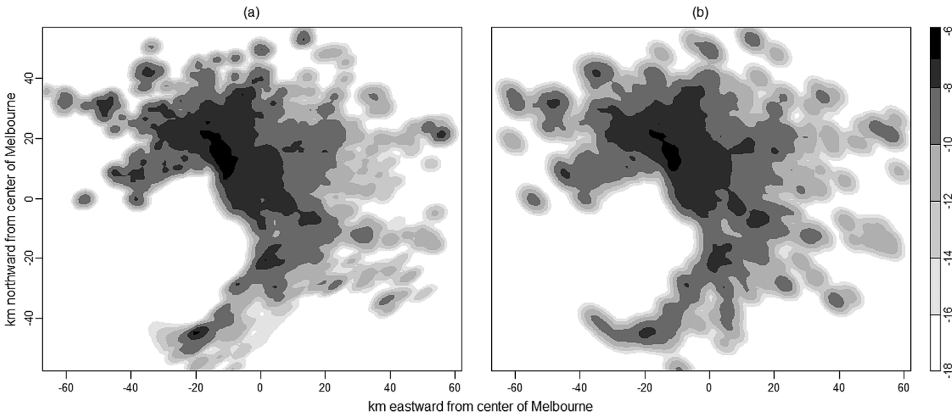

FIG. 4. *Log predictive densities for March* 2, 2011 (*Wednesday*) *at* 10–11 *am using* (a) *spatio-temporal kernel warping* (3 *spatial clusters*), *and* (b) *temporal kernel warping. The density in* (a) *shows slightly more details.*
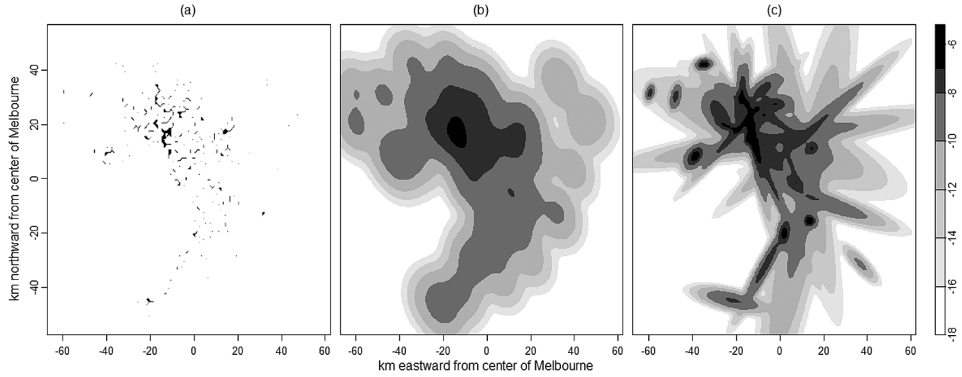
FIG. 5.   *Log predictive densities using comparison methods for* 2–3 *pm on March* 2, 2011 (*Wednesday*): (a) *the MEDIC method* (*an industry practice*); (b) *unwarped KDE with bandwidth selected by the plug-in method* (*PI*); (c) *time-varying Gaussian mixture model with* 30 *components. These densities are to be compared to Figure* 3(b), *which is the prediction using kernel warping for the same period.*

15, 30 and 50 components. The computational expenses are a few times higher than that of the kernel warping method.

Figure 5 shows the log predictive density using the MEDIC method, unwarped KDE (PI) and GMM (30 components) for March 2, 2011 at 2–3 pm. These densities are comparable with Figure 3(b), which shows the log predictive density for the same period predicted by the proposed kernel warping. Even with 400 labeled data, the MEDIC method gives exceedingly noisy predictions, while unwarped KDE and GMM produce over-smoothed densities that do not adapt well to the spatial features of Melbourne.

We use several performance metrics to compare the statistical predictive accuracies of different methods. First, we use average negative log score (LS) [Good (1952)]. This metric is advocated for being a strictly proper scoring rule closely related to Bayes factor and Bayes information criterion [Gneiting and Raftery (2007)]. It is the average negative log likelihood of test data. For each test time period u in the set of all test time periods $\mathcal{T}_{\text{test}}$,

$$\text{LS}(u) = \frac{1}{n_u} \sum_{i=1}^{n_u} - \log \hat{f}_u(\tilde{\mathbf{s}}_{u,i}),$$

in which $\{\tilde{\mathbf{s}}_{u,i}\}$ are the test data and $\hat{f}_u(\cdot)$ is the predictive density for period u obtained by various methods. For the MEDIC method, we floor the cell counts at 10^{-3} and divide over the total count in each period to obtain a nonzero discrete predictive density (to avoid infinite LS). The result is not sensitive to reasonable perturbation of the amount of padding we put.

Second, we compare accuracy in cell counts for every 1-km^2 region and 1-hour period. For the proposed kernel warping, unwarped KDE and GMM, we discretize

TABLE 1
Mean predictive accuracies across all 1-*hour periods in March* 2011 *of the proposed kernel warping and competing methods. Kernel warping outperforms the competing methods*

Prediction method		LS	RMSE	RMSE$_B$
Kernel warping	S-T param	7.53	0.0500	0.0498
	T param	7.56	0.0518	0.0514
(a) MEDIC		10.11	0.0589	0.0996
(b) Unwarped KDE	PI	8.14	0.0562	0.0950
	SCV	8.15	0.0562	0.0950
(c) GMM	15 comp	7.96	0.0562	0.0949
	30 comp	7.87	0.0561	0.0948
	50 comp	7.93	0.0561	0.0949

continuous predictions in space to each 1 km^2, and convert to counts by multiplying the total count for the period as predicted by the MEDIC method. We compute the root-mean-square error, both within the smallest rectangle enclosing all data (plotting window in Figures 1, 3–5) (RMSE) and within a pixelated data-driven boundary of Melbourne B (RMSE$_B$). For each test time period $u \in \mathcal{T}_{\text{test}}$,

$$\text{RMSE}(u) = \sqrt{\frac{1}{C} \sum_{c=1}^{C} (y_{u,c} - \hat{y}_{u,c})^2},$$

where C is the number of 1 km^2 cells in the rectangular observation window, and $y_{u,c}$ and $\hat{y}_{u,c}$ are the actual and predicted count for period u and cell c, respectively. For RMSE$_B$, we use cells c within the pixelated boundary B and C as the number of 1 km^2 cells within this boundary.

We show in Table 1 the mean predictive accuracies of various methods, averaged across all test time periods $\mathcal{T}_{\text{test}}$ (all 1-hour periods in March 2011). A smaller LS, RMSE or RMSE$_B$ indicates better predictive accuracy. Both versions of kernel warping have a significant advantage over the comparison methods in all performance measures. Between the two versions of kernel warping, allowing parameters to be location-specific (in addition to being time-specific) provides additional benefits, even though a large number of time periods choose to use only 1 spatial component. We show in Figure 6 the box-plots illustrating the variations of some of these metrics across time periods. Kernel warping has not only the best mean performance, but also the smallest variations across time periods. We further perform the Diebold–Mariano test [Diebold and Mariano (1995)] on the 672 LS and RMSE$_B$ scores over all times in the test period. For each pair of competing methods and our method (S-T param or T param) in Table 1, we test the null hypothesis that the two methods have the same predictive accuracy against the alternative hypothesis that our method is more accurate. We reject the null hypoth-
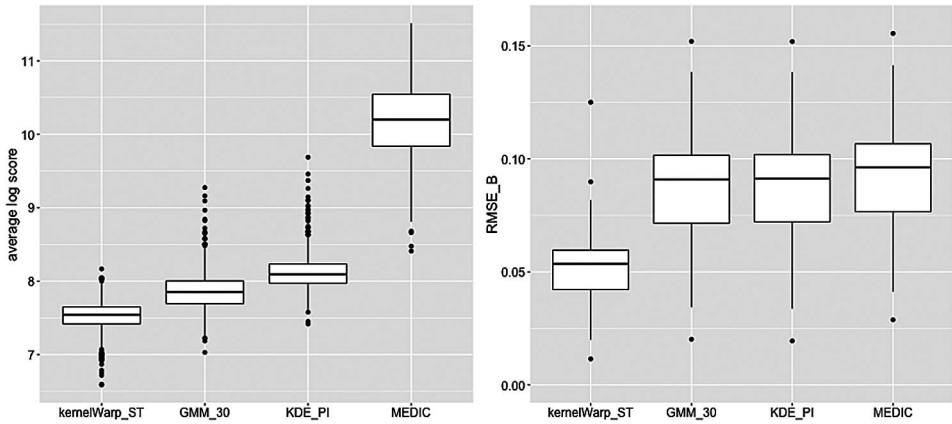
FIG. 6. *Box-plots of predictive accuracies of kernel warping (S-T parameters), GMM (30 comp), KDE (PI bandwidth) and the MEDIC method (an industry practice) over 672 test periods, as measured by average negative log score LS (left) and* RMSE$_B$ *(right). Small values indicate better performance.*

esis for all pairs of our vs. competing methods with exceedingly small p-values ($< 2 \times 10^{-16}$), showing that our improvement in predictive accuracy is significant.

We show in Table 2 the impact of using point clouds of various sizes. Increasing the size of the global point cloud (in T param) brings slight improvements in predictive accuracy and large increases in computational cost. Compared to that, we recommend using the spatial-temporal kernel warping (S-T param), which breaks the spatial region into several spatial components and samples point clouds locally. It has similar performance as using a large global point cloud, but is much faster to compute, as it is faster to invert three $1k \times 1k$ matrices than to invert one $3k \times 3k$ matrix.

TABLE 2
Predictive accuracies using warped and unwarped KDE with a point cloud (unlabeled data) of various sizes

Prediction method		LS	RMSE$_B$
Kernel warping	S-T param (1k each, avg 2.5k)	7.53	0.0498
	T param 1k	7.56	0.0514
	T param 2k	7.54	0.0514
	T param 3k	7.52	0.0514
Unwarped KDE	PI	8.14	0.0950
	PI + 1k	7.95	0.0876
	PI + 2k	7.87	0.0875
	PI + 3k	7.84	0.0875

We also see in Table 2 that adding randomly sampled unlabeled data to un-warped KDE (PI + 1k, + 2k, + 3k) produces small benefits, diminishing as the size of unlabeled data increases, while the extra computational effort is nontrivial. With these additions, the unwarped KDE is still not competitive compared to kernel warping. While using more data ought to provide an advantage, the unlabeled data starts to overwhelm the labeled data. We lose some time-specific information and predict quite similarly across all periods. There is no simple way to distinguish between the labeled and unlabeled data in unwarped KDE. An attractive feature of the warped KDE is that it allows for a way to incorporate unlabeled data to inform the geometry of the data, while not diluting the set of labeled data most relevant to the predictive task at hand.

4. Conclusions. Fine-resolution spatio-temporal ambulance demand predictions are critical to optimal ambulance planning. Typical challenges include data sparsity at the prediction resolution and incorporation of complex urban spatial domains. These challenges are especially prominent in Melbourne. They create a tension; overcoming sparsity requires considerable smoothing, while representing complex spatial features requires fine-resolution modeling. Most current industry practices and earlier studies are ill-equipped to address these challenges simultaneously. We propose a kernel warping method that smooths intelligently toward geographical characteristics. We demonstrate that our proposed method predicts ambulance demand in Melbourne more accurately than the state of the art in the practice and research of ambulance demand prediction.

To predict ambulance demand for any hour, we use a spatio-temporal kernel density estimator on the sparse set of most similarly labeled data, but warp these kernels to a larger point cloud drawn from all historical observations regardless of labels. We construct an adjacency graph on this point cloud to approximate Melbourne's spatial boundaries, neighborhoods and road networks in a data-driven manner. Kernels on labeled data are warped to encourage flow along and penalize flow orthogonal to this graph.

Kernel warping circumvents the need to define boundaries and boundary conditions, which are often difficult in the practice of modeling point patterns on complex spatial domains. It also captures and exploits finer-grain internal spatial structures other than boundary features, which can be prominent in various heterogeneous environments such as cities, buildings, mountains and forests. Kernel warping is not limited to density estimation. It can be adapted to model a wide range of functions and surfaces. It can be used to perform a broad set of tasks including prediction, classification, clustering and visualization. Inferences on uncertainty, if desired, can be obtained by assessing cross-validation variance and warping kernels to different samples of point clouds. There is much flexibility in designing the point cloud and its Laplacian, and it can be customized to the needs of different applications (e.g., exterior/interior boundaries, road/train network, etc.). We offer

some discussions on these in the context of spatial and spatio-temporal point patterns. We also offer efficient estimation of kernel bandwidth and degree of warping local to time periods and locations via cross-validation. The proposed method is straightforward to implement and easy to experiment with. The tools we have developed can be easily generalized to model a wide range of spatial or spatio-temporal point processes on complex spatial domains.

REFERENCES

AGGARWAL, C. C. (2003). A framework for diagonosing changes in evolving data streams. In *ACM SIGMOD International Conference on Management of Data* 575–586. ACM, New York.

BELKIN, M. and NIYOGI, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **15** 1373–1396.

BELKIN, M. and NIYOGI, P. (2004). Semi-supervised learning on Riemannian manifolds. *Mach. Learn.* **56** 209–239.

BELKIN, M. and NIYOGI, P. (2005). Towards a theoretical foundation for Laplacian-based manifold methods. In *Learning Theory. Lecture Notes in Computer Science* **3559** 486–500. Springer, Berlin. MR2203282

BELKIN, M., NIYOGI, P. and SINDHWANI, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *J. Mach. Learn. Res.* **7** 2399–2434. MR2274444

BOUSQUET, O., CHAPELLE, O. and HEIN, M. (2005). Measure based regularization. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge.

CACOULLOS, T. (1966). Estimation of a multivariate density. *Ann. Inst. Statist. Math.* **18** 179–189. MR0210255

CHANNOUF, N., L'ECUYER, P., INGOLFSSON, A. and AVRAMIDIS, A. N. (2007). The application of forecasting techniques to modeling emergency medical system calls in Calgary, Alberta. *Health Care Manag. Sci.* **10** 25–45.

DIEBOLD, F. and MARIANO, R. (1995). Comparing predictive accuracy. *J. Bus. Econom. Statist.* **13** 253–263.

DIGGLE, P. J. (2003). *Statistical Analysis of Spatial Point Patterns*, 2nd ed. Arnold, London.

DONOHO, D. L. and GRIMES, C. (2005). Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. In *Proceedings of the National Academy of Sciences* **102**. National Academy of Sciences, Washington DC.

DUONG, T. and HAZELTON, M. L. (2005). Cross-validation bandwidth matrices for multivariate kernel density estimation. *Scand. J. Stat.* **32** 485–506. MR2204631

ERTÖZ, L., STEINBACH, M. and KUMAR, V. (2003). Finding clusters of different sizes, shapes and densities in noisy, high dimensional data. In *Proceedings of the SIAM International Conference on Data Mining* 47–58. SIAM, Philadelphia.

ESTER, M., KRIEGEL, H. P., SANDER, J. and XU, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noice. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* 226–231. ACM, New York.

FREY, B. J. and DUECK, D. (2007). Clustering by passing messages between data points. *Science* **315** 972–976. MR2292174

GNEITING, T. and RAFTERY, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *J. Amer. Statist. Assoc.* **102** 359–378. MR2345548

GOLDBERG, J. B. (2004). Operations research methods for the deployment of emergency service vehicles. *EMS Management Journal* **1** 20–39.

GOOD, I. J. (1952). Rational decisions. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **14** 107–114. MR0077033

GOOGLE MAPS (2015). Map of Melbourne, Australia. Web.

GRAY, A. G. and MOORE, A. W. (2003). Nonparametric density estimation: Toward computational tractability. In *Proceedings of the SIAM International Conference on Data Mining*. SIAM, Philadelphia.

MATTESON, D. S., MCLEAN, M. W., WOODARD, D. B. and HENDERSON, S. G. (2011). Forecasting emergency medical service call arrival rates. *Ann. Appl. Stat.* **5** 1379–1406. MR2849778

MERRIS, R. (1994). Laplacian matrices of graphs: A survey. *Linear Algebra Appl.* **197/198** 143–176. MR1275613

MØLLER, J. and WAAGEPETERSEN, R. P. (2004). *Statistical Inference and Simulation for Spatial Point Processes. Monographs on Statistics and Applied Probability* **100**. Chapman & Hall/CRC, Boca Raton, FL. MR2004226

NAKAYA, T. and YANO, K. (2010). Visualising crime clusters in a space–time cube: An exploratory data analysis approach using space–time kernel density estimation and scan statistics. *Transactions in GIS* **14** 223–239.

NG, A., JORDAN, M. and WEISS, Y. (2001). On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems* 849–856. MIT Press, Cambridge.

RAMSAY, T. (2002). Spline smoothing over difficult regions. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **64** 307–319. MR1904707

REGIS, R. G. and SHOEMAKER, C. A. (2007). A stochastic radial basis function method for the global optimization of expensive functions. *INFORMS J. Comput.* **19** 497–509. MR2364007

REGIS, R. G. and SHOEMAKER, C. A. (2009). Parallel stochastic global optimization using radial basis functions. *INFORMS J. Comput.* **21** 411–426. MR2546962

RIPLEY, B. D. and RASSON, J.-P. (1977). Finding the edge of a Poisson forest. *J. Appl. Probab.* **14** 483–491. MR0451339

ROWEIS, S. T. and SAUL, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science* **290** 2323–2326.

SCOTT, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York. MR1191168

SETZLER, H., SAYDAM, C. and PARK, S. (2009). EMS call volume predictions: A comparative study. *Comput. Oper. Res.* **36** 1843–1851.

SHI, J. and MALIK, J. (2000). Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **22** 888–905.

SINDHWANI, V., NIYOGI, P. and BELKIN, M. (2005). Beyond the point cloud: From transductive to semi-supervised learning. In *Proceedings of the 22nd International Conference on Machine Learning* 824–831. ACM, New York.

SMOLA, A. J. and KONDOR, R. (2003). Kernels and regularization on graphs. In *Learning Theory and Kernel Machines, Lecture Notes in Computer Science* 144–158. Springer, Berlin.

TENEBAUM, J. B., DE SILVA, V. and LANGFORD, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science* **290** 2319–2323.

VAN DER MAATEN, L. J. P., POSTMA, E. O. and VAN DEN HERIK, H. J. (2009). Dimensionality reduction: A comparative review. *J. Mach. Learn. Res.* **10** 66–71.

VILE, J. L., GILLARD, J. W., HARPER, P. R. and KNIGHT, V. A. (2012). Predicting ambulance demand using singular spectrum analysis. *Journal of the Operations Research Society* **63** 1556–1565.

WAND, M. P. and JONES, M. C. (1994). Multivariate plug-in bandwidth selection. *Comput. Statist.* **9** 97–116. MR1280754

WOOD, S. N., BRAVINGTON, M. V. and HEDLEY, S. L. (2008). Soap film smoothing. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **70** 931–955. MR2530324

WOODWORTH, J. T., MOHLER, G. O., BERTOZZI, A. L. and BRANTINGHAM, P. J. (2014). Non-local crime density estimation incorporating housing information. *Philos. Trans. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **372** 20130403, 15. MR3268065

ZHANG, Z., CHEN, D., LIU, W., RACINE, J. S., ONG, S. H., CHENG, Y., ZHAO, G. and JIANG, Q. (2011). Nonparametric evaluation of dynamic disease risk: A spatio-temporal kernel approach. *PLoS ONE* **6**.

ZHOU, Z. and MATTESON, D. S. (2015). Predicting ambulance demand: A spatio-temporal kernel approach. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York.

ZHOU, D., BOUSQUET, O., LAL, T. N., WESTON, J. and SCHOELKOPF, B. (2003). Learning with local and global consistency. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge.

ZHOU, Z., MATTESON, D. S., WOODARD, D. B., HENDERSON, S. G. and MICHEAS, A. C. (2015). A spatio-temporal point process model for ambulance demand. *J. Amer. Statist. Assoc.* **110** 6–15. MR3338482

ZHU, X., KANDOLA, J., GHAHRAMAMI, Z. and LAFFERTY, J. (2005). Nonparametric transforms of graph kernels for semi-supervised learning. In *Advances in Neural Information Processing Systems*. MIT Press, Cambridge.

CENTER FOR APPLIED MATHEMATICS   DEPARTMENT OF STATISTICAL SCIENCE
CORNELL UNIVERSITY   CORNELL UNIVERSITY
ITHACA, NEW YORK 14853   ITHACA, NEW YORK 14853
USA   USA
E-MAIL: zz254@cornell.edu   E-MAIL: matteson@cornell.edu