Modeling Human Location Data with Mixtures of Kernel Densities

Moshe Lichman
Department of Computer Science
University of California, Irvine
mlichman@ics.uci.edu

Padhraic Smyth
Department of Computer Science
University of California, Irvine
smyth@ics.uci.edu

ABSTRACT

Location-based data is increasingly prevalent with the rapid increase and adoption of mobile devices. In this paper we address the problem of learning spatial density models, focusing specifically on individual-level data. Modeling and predicting a spatial distribution for an individual is a challenging problem given both (a) the typical sparsity of data at the individual level and (b) the heterogeneity of spatial mobility patterns across individuals. We investigate the application of kernel density estimation (KDE) to this problem using a mixture model approach that can interpolate between an individual's data and broader patterns in the population as a whole. The mixture-KDE approach is evaluated on two large geolocation/check-in data sets, from Twitter and Gowalla, with comparisons to non-KDE baselines, using both log-likelihood and detection of simulated identity theft as evaluation metrics. Our experimental results indicate that the mixture-KDE method provides a useful and accurate methodology for capturing and predicting individual-level spatial patterns in the presence of noisy and sparse data.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Application—*Data Mining, Spatial databases and GIS*

Keywords

spatial; kernel density estimation; anomaly/novelty detection; probabilistic methods; social media; user modeling

1. INTRODUCTION

Human location data is increasingly available in the modern mobile world, often in the form of geolocation tags attached to human behavioral data such as phone calls, text messages, social media activities, and more. With the widespread availability of this data there is increasing interest

KDD'14, August 24–27, 2014, New York, NY, USA.

ACM 978-1-4503-2956-9/14/08 ...$15.00.
http://dx.doi.org/10.1145/2623330.2623681.

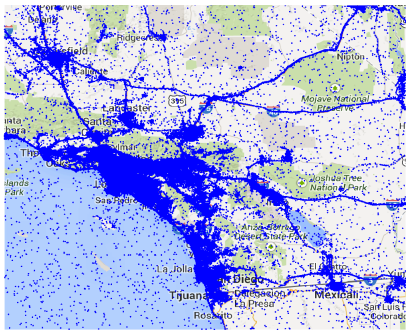

Figure 1: Geolocated Twitter tweets in southern California. Points over the ocean and sparsly populated areas are largely a result of noise in the geolocation data rather than actual locations.

across a variety of fields of study in creating accurate models to characterize the spatial distributions of populations and individuals. For example, Sadilek et al. [23] analyze the spread of infectious diseases through geolocation data from Twitter, opening up potential new approaches for real-time computational epidemiology. Cranshaw et al. [9] use Foursquare check-in data to identify local spatial clusters within urban areas, with potential applications in urban planning to economic development and resource allocation. From a commercial perspective, location-based services and personalization are becoming increasingly important to individual mobile device users, with an increasing number of applications that are location-aware, including maps, localized search results, recommender systems, and advertising [28].

In this paper we focus on the problem of developing accurate individual-level models of spatial location based on geolocated event data. The term "event" here can be interpreted in a broad context—examples include communication events such as phone calls or text messages, check-in events, social media actions, and so on. The goal is to be able to accurately characterize and predict the spatial pattern of an individual's events. The problem is challenging for two main reasons. Firstly, there is often relatively little data for many of the individuals making it difficult to build accurate models at the individual level. Secondly, there is often considerable variety and heterogeneity in the spatial pattern of events of individual users, rendering techniques such as clustering less than ideal for individual-level modeling.

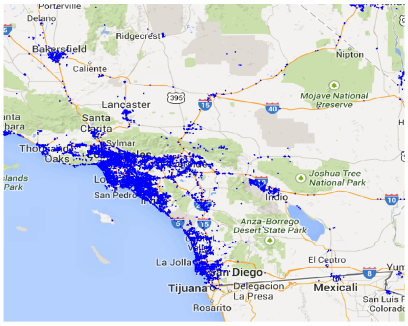

Figure 2: Gowalla checkins in southern California.

The primary contribution of our paper is a systematic approach for individual-level modeling of geolocation data based on kernel density estimates. Kernel density approaches have been relatively unexplored to date in the context of spatial modeling of geolocation data. While in principle they provide a flexible and general framework for spatial density estimation, direct application at the level of individual event data will tend to lead to significant over-fitting (because of the sparsity of data at the individual level). We propose a hierarchical extension of the traditional kernel approach that can avoid the over-fitting problem by systematically smoothing an individual's data towards the population data. We demonstrate how adaptive bandwidth techniques provide better quality density estimates compared to fixed bandwidths. Using two large geolocation data sets from Twitter and Gowalla (see Figures 1 and 2) we show that the proposed kernel method is significantly more accurate than baselines such as Gaussian mixtures for such data, in terms of the quality of predictions on out-of-sample data.

This paper is organized as follows. Section 2 provides an overview of existing approaches for modeling human location data and elaborates on the challenges of developing spatial models in practice. In section 3 we review kernel density estimation and discuss the use of adaptive bandwidth methods and in section 4 we describe our proposed mixture-KDE approach for modeling and predicting individuals' locations. In section 5 we present empirical experiments using two different geospatial/check-in data sets and using both test log-likelihood and accuracy in detection of simulated identity theft. Section 6 discusses scalability and online algorithms for the proposed approach and we conclude with a brief discussion in Section 7.

2. BACKGROUND AND RELATED WORK

2.1 Notation and Problem Definition

In this paper we consider data available in the form of individual-level geotagged events, $\mathbf{E} = \{E_1, \ldots, E_N\}$ where $E_i = \{e_i^1, \ldots, e_i^{n_i}\}$ and e_i^j is the jth event for the ith individual, $1 \leq i \leq N$. Each event e_i^j consists of a tuple $< i, x, y, t >$, where x and y are longitude and latitude respectively and t is a time-stamp, e.g., geotagged tweets based on GPS location.

One approach in analyzing such data is to focus on the problem of sequentially predicting a user's behavior in terms of their short-term trajectory, e.g., predicting where a user i's next event e_i^{j+1} is likely to occur in terms of location $< x, y >$ given their event history $\{e_i^1, \ldots, e_i^j\}$, where events may be minutes or hours apart (e.g., Song et al. [27] and Scellato et al. [24]). In this paper we focus on a different problem, that of modeling a user's spatial patterns over a longer time-period in a more aggregate sense. Specifically, we focus on learning probability density models of the form $f_i(x, y)$ that represent the spatial density of user i's events. Given an event has occurred for individual i, the probability that it lies in any area A is $\int\int f_i(x, y)dxdy$ where the integral is over the region defined by A (see also [16]). In this context we focus in this paper on modeling $f_i(x, y)$ rather than $f_i(x, y, t)$, and only use the time dimension t to order the data. e.g. for online training and prediction. In principle it should be possible to extend the 2-dimensional spatial modeling methods proposed in this paper to include the temporal dimension, allowing for inclusion of circadian and calendar-dependent aspects of an individual's behavior.

2.2 Modeling of Discretized Locations

A widely-used approach in location modeling is to restrict attention to a finite set of known fixed locations, effectively discretizing space and turning the problem into a multivariate data analysis problem where each location represents a single dimension. One can use such representations to generate a sparse matrix consisting of individuals as rows and locations as columns, where each cell i, j contains the count of the number of events for individual i that occurred at location j. The locations (columns) can be defined in different ways. For example, one can define the columns by identifying a set of specific locations such as shops, restaurants, and so forth (e.g., see [5, 6, 9]). An alternative approach is to discretize the spatial domain into disjoint cells (e.g., via clustering), and then associate a discrete set of venues with each cell (as in Cranshaw et al. [10] who used Foursquare checkin venues) or to aggregate the counts of geolocated events within each cell (as in Lee et al. [20] and Frias-Martinez et al. [15]). The advantage of these discretized representations is that they allow the application of broad set of multivariate data analysis tools, such as clustering techniques or matrix decomposition methods. However, they do not explicitly encode spatial semantics and, as such, do not provide the ability to make predictions in continuous space, which is a primary aim in our work.

2.3 Continuous Spatial Models

In the context of continuous models, a number of authors have explored such models for individual location data in prior work. For example, Gonzalez et al. [16] and Brockmann et al. [4] explored general distributional patterns of human mobility from location data. Eagle and Pentland [12], Li et al. [21], and Cho et al. [7] demonstrated how different aspects of individuals' daily routines can be effectively extracted from traces of location data.

A simple approach to modeling an individual's spatial density $f_i(x, y)$ is to use a single Gaussian density function, i.e.,

$$f_G(x, y|\underline{\mu}_i, \Sigma_i) = f_G(e_i|\underline{\mu}_i, \Sigma_i) = \frac{1}{(2\pi)\,|\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(e_i-\underline{\mu}_i)'\Sigma_i^{-1}(e_i-\underline{\mu}_i)} \quad (1)$$

where $e_i = (x, y)$ is a 2-dimensional longitude-latitude pair, $\underline{\mu}_i$ is a 2-dimensional mean vector, and Σ_i is a 2×2 covariance matrix. The unimodal and elliptical density contours

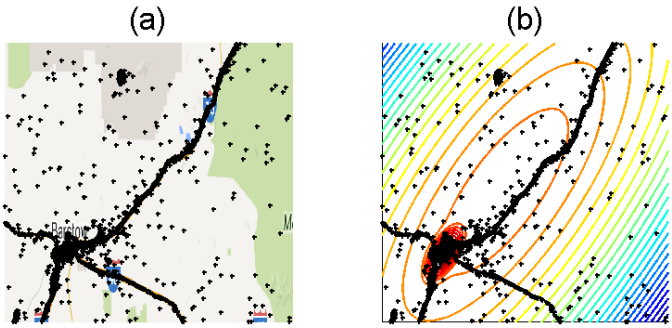

Figure 3: On the left (a): Geotagged events in the area between Los Angeles and Las Vegas near the city of Barstow, CA. (b): The contour lines of a Gaussian mixture model with 2 components. Figure best viewed in color.

of a single Gaussian are too simple to accurately represent human location data in practice. With this in mind, a finite mixture of C Gaussian densities can provide additional flexibility, defined as

$$f_{MG}(e_i|\theta_i) = \sum_{c=1}^{C} \pi_{ic} f_G(e|\underline{\mu}_{ic}, \Sigma_{ic}) \quad (2)$$

with parameters θ_i consisting of the C mixing weights $\pi_1, \ldots, \pi_C$, $\sum_c \pi_c = 1$, and means $\underline{\mu}_{ic}$ and covariance matrices Σ_{ic}, $1 \leq c \leq C$. For example, as described in Cho et al. [7], a two-component ($C = 2$) spatial model may be a useful model for capturing the bimodal variation due to "home" and "work" components in an individual's spatial data.

While the mixture model can provide additional modeling power beyond that of a single Gaussian, it has a number of practical limitations. Firstly, the number of components C required for an accurate density model may vary considerably across different individuals, and automatically and reliably determining the number of components is a non-trivial problem. Secondly, the number of data points per individual is usually skewed towards small counts. For example, in our Twitter data set 60% of the individuals have associated with them 5 or fewer events over a 2 month period (July and August 2013). This makes it challenging, if not impossible, to fit mixture models, even if the number of components C for each individual is known and fixed. A third limitation of the Gaussian mixture approach is a more pragmatic one. Human mobility is constrained by our environment resulting in sharp transitions in spatial densities, due both to natural topography (mountains, oceans) and man-made artifacts (roads, city centers, etc.). Figure 3(a) shows Twitter data for a region near Barstow, California. The spatial density of the data shows significant local variation, including regions of high density for the town of Barstow (bottom left), for the military base (top center), and along the various major roads, with very low density in the surrounding desert. Figure 3(b) shows a fitted mixture density model with two components: it is unable to capture many of the high density patterns and "wastes" considerable probability mass over sparsely populated desert regions.

3. KERNEL DENSITY ESTIMATION FOR SPATIAL LOCATION DATA

To address these limitations, we investigate the use of kernel density estimation (KDE) methods as outlined in detail in the next section. There has been limited prior work investigating the application of KDE methods in the context of human location data. Zhang and Chow [29] illustrated the advantages of KDE techniques (over Gaussian mixture models) for data from location-based social networks, but used 1-dimensional kernel densities on distances rather than 2-dimensional spatial models, and Hasan et al. [18] illustrated the use of 2-dimensional spatial KDE models for exploratory analysis of check-in data. KDE methods have also been used in application areas such as epidemiology [2], ecology [14], and marketing [11], for modeling spatial densities of *populations* of individual entities, but not for modeling spatial densities of individuals themselves.

3.1 Kernel Density Estimation

Kernel density estimation is a non-parametric method for estimating a density function from a random sample of data [25]. Let $E = \{e^1, \ldots, e^n\}$ be a set of historical events where $e^j =< x, y >$ is a two-dimensional location, $1 \leq j \leq n$, and where we have suppressed any dependence on individual i for the moment and dropped dependence on time t. We will refer to E as the training data set. A simple approach for estimating a bivariate density function from such data is to use a single fixed bandwidth h for both spatial dimensions and a Gaussian kernel function $K(\cdot)$. This results in a bivariate KDE of the following form:

$$f_{KD}(e|E,h) = \frac{1}{n}\sum_{j=1}^{n} K_h(e, e^j) \quad (3)$$

$$K_h(e, e^j) = \frac{1}{2\pi h}\exp\left(-\frac{1}{2}(e-e^j)^t \Sigma_h^{-1}(e-e^j)\right) \quad (4)$$

$$\Sigma_h = \begin{pmatrix} h & 0 \\ 0 & h \end{pmatrix}$$

where e is the location for which we wish to compute the density and $h > 0$ is a fixed scalar bandwidth parameter for all events in E. It is well known that the resulting density estimate f_{KD} can be highly sensitive to the value of the bandwidth h, producing densities that are sharply peaked around the training data points e^j when h is too small, and producing an overly smooth estimate that may omit important structure in the data (such as multiple modes) when h is too large [25].

There are a number of techniques that can be used to evaluate the quality of a particular value for the bandwidth h. One straightforward data-driven option is to measure the log-probability (or log-likelihood) of a set of test data points not used in constructing the density estimate, i.e.,

$$L(h) = \frac{1}{n_t}\sum_{r=1}^{n_t} \log f_{KD}(e^r|E,h) \quad (5)$$

where the n_t events e^r are data points not included in the training data E (e.g., a validation set). Larger values of $L(h)$ are preferred since it means that higher probability is being assigned to new unseen data. Hence, a simple approach to

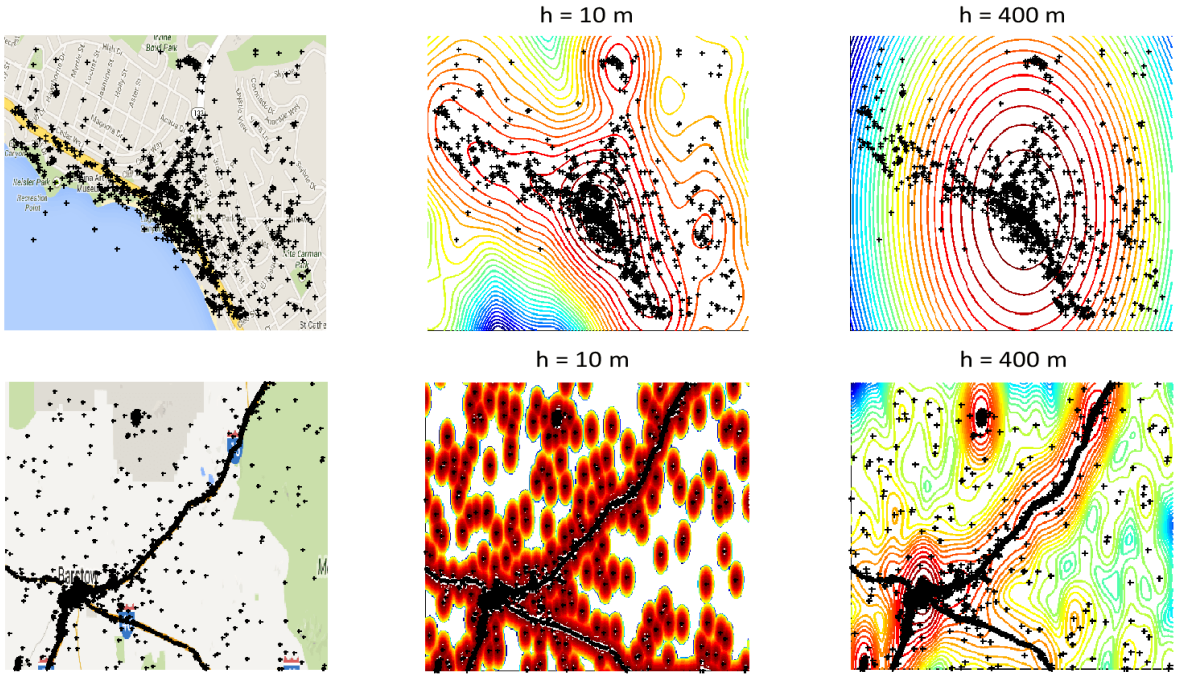

Figure 4: Left plots: Events in the city of Laguna Beach in southern California (top row) and in an area between Los Angeles and Las Vegas (bottom row). Points on the map represent observed events. Middle plots: Contour of the log-probability of a KDE model with fixed $h = 10$ meters. Right Plots: Contour of the log-probability of a KDE model with fixed $h = 400$ meters. Best viewed in color.

bandwidth selection (at least for a single bandwidth parameter h) is to perform a grid-search on h using a validation set. We will use the above out-of-sample log-probability score function $L(h)$ later in the paper for both bandwidth selection and for comparing different types of density models.

One could also use various "plug-in" estimates for h, such as that of Silverman [25, pages 86-88]. These estimates are optimal (e.g. in a mean integrated squared error sense) if the true underlying density being estimated is Gaussian, and can work well in practice for other related smooth non-Gaussian densities. However, for spatial location data we found that "plug-in" estimates for h were much too large and significantly oversmoothed the KDE results in a very poor fit due to the highly multimodal nature of location data.

At this point in the discussion it is worth noting that, in addition to its advantages in terms of flexibility, kernel density estimation has some well known drawbacks that have tended to limit its use in practice in the past, particularly in machine learning and data mining. The first drawback is that it is particularly susceptible to the curse of dimensionality, essentially requiring an exponential number of data points as a function of data dimensionality d. This is not an issue for location-data modeling since we are in the low-dimensional regime of $d = 2$. A second drawback of kernel densities (as with related "neighbor-based" methods) is the need to store all of the training data in memory at prediction time. This was arguably a relevant point 10 years or more ago when memory was relatively expensive, but in current times it is relatively inexpensive (both computationally and financially) to keep millions (or even hundreds of millions) of points accessible in main memory at prediction time. We will return to this point in more detail later in the paper—here it is sufficient to note that kernel density estimation is practical for 2-dimensional problems with millions of data points.

3.2 The Adaptive Bandwidth Method

A limitation of the approach described above is that the smoothing is homogeneous, i.e., the amount of smoothing is constant through the 2-dimensional region since the bandwidth h is fixed for all events. This does not reflect the realities of human-location data where dense urban areas will tend to have high event density and sparsely-populated rural areas will have low event density. This limitation is clearly visible in Figure 4. The two plots in the center use the same small fixed bandwidth of $h = 10$ meters, which works well for the relatively dense area of Laguna Beach (upper plot), but works poorly (overfits) for the rural area near Barstow, CA in the lower plot. If the bandwidth is increased to $h = 400$ meters, as in the two plots to the right, we find that this produces a more acceptable result in the lower plot (the rural area) but is vastly oversmoothing in the upper plot.

One approach to address this issue is to use an adaptive kernel bandwidth, several methods of which have been proposed in the literature, that relaxes the assumption of a constant fixed bandwidth parameter. Breiman et al. [3] suggested adapting the kernel bandwidth h^j *to each data point* e^j. Using this idea, we let h^j be the Euclidean distance to the kth nearest neighbor to e^j in the training data. Hence

	Bandwidth	AvgLogL
Fixed	$h = 10^{-2}$	-0.592
	$h = 10^{-3}$	-0.157
	$h = 10^{-4}$	0.139
	$h = 10^{-5}$	-0.326
Adaptive	$k = 2$	0.046
	$k = 5$	1.275
	$k = 10$	1.196
	$k = 20$	0.354

Table 1: Average log-probability scores on held-out events, comparing the fixed and the adaptive approaches for kernel density estimation for Twitter geolocation data.

we can define an adaptive bandwidth kernel density estimate as:

$$f_{KD}(e|E) = \frac{1}{n}\sum_{j=1}^{n} K_{h^j}(e, e^j) \quad (6)$$

where K_{h^j} is defined as in Equation 4 replacing h with h^j.

Table 1 shows the results from a series of tests on a validation data set, comparing the fixed and adaptive bandwidth approaches using different values for (a) the fixed bandwidth h, and (b) the number of neighbors k (for the adaptive method). We trained the models using 100,000 randomly selected events from our Twitter data set (described in more detail later in the paper) and then computed the log-probability score (Equation 5) using a set of $n_t = 100,000$ randomly selected held-out events. From the results, we can see that the adaptive bandwidth models dominate the performance of the fixed bandwidth methods. As a sidenote, the "plug-in" methods performed significantly worse (results not shown).

4. MODELING AN INDIVIDUAL'S LOCATION DATA

So far, our predictive model $f_{KD}(e|E, h)$, does not depend on the identity of an individual i. However, our primary goal in this work is to be able to build accurate predictive spatial density models at the individual level.

4.1 Mixtures of Kernel Density Models

To address this task, we could apply the adaptive kernel density methods described above at the level of an individual (rather than for aggregations of events across a population of individuals), computing $f_{KD}(e|E_i)$ in Equation 6 where we now condition on just the individual's event data E_i rather than the events for the population E.

A significant challenge with building individual-level models in this manner is the "cold-start" problem, given that we typically have very little data for many of the individuals for whom we wish to make predictions. To address this data sparsity problem we propose a multi-scale kernel model where we use a mixture of (a) an individual's potentially noisy kernel density estimate with (b) more robust coarse-scale models[1] . More specifically we define a mixture-KDE for individual i as

$$P_{MKD}(e|E) = \sum_{c=1}^{C} \alpha_c f_{KD}(e|E^c) \quad (7)$$

where $\alpha_1, \ldots, \ldots, \alpha_C$ are non-negative mixing weights with $\sum_c \alpha_c = 1$, and $f_{KD}(e|E^c)$ is the cth component of the mixture. Here component c is a kernel density estimate computed as a function only of a subset of points (or events) E^c. The component density estimates, f_{KD}, can be any density model, including fixed or adaptive bandwidth KDEs. We use adaptive bandwidth KDEs, with $k = 5$ (following Table 1), for all of the components in the mixture-KDEs used in this paper.

As a specific example consider a model for individual i where $C = 2$, with the first component being the individual-level kernel density with $E^1 = E_i$, and the second component being a population-level kernel density estimate with $E^2 = E$. This mixture will have the effect of smoothing the individual's density towards the population density, with more or less smoothing depending on the relative size of the α weights. Note that this mixture is significantly different in nature to the use of a Gaussian mixture for an individual's data (e.g., as in [7]). In that approach, each component typically represents a different spatial location around which an individual's activity is centered (such as "home" and "work"), whereas in the mixture-KDE each mixture component is responsible for a broader spatial scale of activity.

For C components, where $C > 2$, we can have the first component be an individual level density, and the Cth component be the population density, where the intermediate components $c = 2, \ldots, C-1$ can represent different spatial scales (such as neighborhoods, cities, states, and even countries). Given that the data sets we are using in this paper are from the Southern California region, we chose to use a 3-level model ($C = 3$), with the first and last components being the individual and population level models respectively, and the $c = 2$ model representing (approximately) the "home city" for an individual (more details on this intermediate scale component are provided later in the paper). This process of selecting additional spatial components, that are "between" the individual and the full population, is somewhat arbitrary—we chose a single intermediate component in the work below, but other choices could be explored for other applications.

From a generative perspective, one interpretation of Equation 7 above for the mixture-KDE is as follows. Assuming that the first component is based only on individual i's data, individual i has a probability α_1 of generating events in the future in a manner similar to his/her past behavior, and a probability α_c ($c = 2, \ldots, C$) of generating events in accordance with the larger "subpopulations" defined by events E^c. In this paper, in the absence of additional metadata about the individuals, we defined the larger subpopulations solely on spatial characteristics (component 2 being roughly a city, and component 3 being the whole southern California region). However, one could also define the larger subpop-

[1]A potential alternative option would be a Bayesian hierarchical model—however Bayesian modeling with kernel densities is not straightforward given that kernels don't have parameters that can be "shrunk" as in the usual Bayesian approach.

ulations based on metadata (if it were available), such as demographics, social ties, and so forth.

Another way to interpret the mixture-KDE model is that it provides a relatively simple weighting mechanism to allow us to upweight data points from individual i in the kernel density estimate and downweight points that don't belong to the individual. Given that kernel densities are defined as weighted sums over all points, the mixture-KDE can be thought of as a form of kernel density estimate where the points are given an additional level of weighting depending on what component they belong to. As a sidenote, in the results presented here we allow data points to belong to multiple sets E^c, e.g., a data point for individual i can be included in the kernel density estimate for all C components. The other option would be to only allow points to belong to a single component. Either option is fine in theory: from the weighted kernel perspective it only changes (in effect) the manner in which we are assigning weights to each point in the overall weighted sum.

4.2 Training the Model

Given a set of components $f_{KD}(e|E^c), c = 1 \dots C$, the next step is to compute the mixing weights $\alpha_1, \dots, \alpha_C$ in Equation 7. To do so, we randomly sample a validation set, disjoint from the training set, and use it to learn the weights as follows. For each event in the validation set, its density value under each component c is computed (using the training set for the KDE computation). This results in a fixed set of component density values on the validation data points and we can optimize over the convex set of mixing weights α_c to find the highest-likelihood combination. We used the Expectation-Maximization (EM) algorithm since it is easy to implement and converges quickly (one could use other optimization techniques such as gradient descent)—this is in essence the same as using EM for learning finite mixtures (such as Gaussian mixtures) but where the parameters of the component densities are known and fixed and one is just learning the mixture weights (see also Smyth and Wolpert [26]). An alternative to using fixed α's would be to allow the weights to vary by individual, in particular as a function of the number of data points for each individual. Preliminary exploration of this idea suggested that any additional predictive power that might be attained would likely not be justified by the additional complexity.

5. EXPERIMENTS AND RESULTS

5.1 Data Sets

For our experiments, we use two geolocation/check-in data sets: **Twitter** and **Gowalla**. Twitter is a popular microblogging service that allows the posting of short texts (up to 140 characters) and pictures. Using the Twitter API [1] we collected over 4 million public tweets from 230,450 unique individuals in the Southern California area over the period of July-August 2013. In the experiments in this paper we use data only from weekdays, i.e. Monday trough Friday. To remove repeated and bursty events we replaced tweets occurring with the same hour and within 50 meters of each other with a single effective tweet. Figure 1 shows the spatial distribution of our data set. The Gowalla data is the same data used by Cho et al. [7], containing $145,558$ events from $7,653$ unique individuals on weekdays between January and October, 2010, in the southern California area as shown in Figure 2. Training, validation, and test sets were extracted from both data sets for our experiments (details provided later).

5.2 Models Evaluated

We evaluated each of the following individual-level models in our experiments. By an "individual-level model" we mean a model that is fit to each individual i using only their data E_i, and then used for predicting future events for that individual i.

Gaussian: A Gaussian density model. We used maximum likelihood estimates for the mean and maximum a posteriori (MAP) estimates for the covariance matrices (e.g., see [22], chapter 4.6.2):

$$\mu = \hat{\mu}_{MLE}, \quad \Sigma = \lambda\Sigma_0 + (1-\lambda)\hat{\Sigma}_{MLE}, \quad \lambda = \frac{n_0}{n_0 + n_i}$$

where the prior parameters were set to $n_0 = 3$ and $\sigma = 5$ kilometers (for the diagonal on the prior covariance) via a grid search over the log-likelihood on a validation set across all individuals.

Gaussian Mixture Model (GMM): Two different Gaussian mixture models with $C = 2$ and $C = 4$ components. The models were fit using the EM algorithm. Again, we used maximum likelihood to estimate the μ_c and MAP estimates for the Σ_c. Parameters for the priors were also determined on a validation set.

Fixed KDE: A fixed bandwidth kernel density estimate using Equation 3. The validation set was used to determine a single fixed bandwidth, $h = 5.3$ kilometers, for all users.

Adaptive KDE: An adaptive kernel density estimate using Equation 6. The validation set was used to determine a nearest-neighbor value of $k = 5$ for all users.

In addition, for our log-likelihood experiments we evaluated a single "global" population model (Population KDE) using an adaptive kernel density estimate ($k = 5$) based on **all** data points in the training set (i.e., not an individual-level model).

For our mixture-KDE model we used 3 components, the first and last corresponding to individual i and to the full population, respectively. Each component is an adaptive bandwidth KDE with $k = 5$ neighbors. For the middle component we divided the southern Calfornia area into 81 regions corresponding to a 9×9 array of equal-sized grid boxes. Each individual i was associated with the region that contains the majority of their individual events—thus, the $c = 2$ component in the model represents the scale of a local region or city. A similar approach was used in prior work for finding the "home" location of an individual [7]. Using EM to determine the mixture weights resulted in the following values for the α's: 0.85 for α_1 (the individual level), 0.12 for α_2 (the region level), and 0.03 for α_3 (the population level) for the Twitter data set, and $\alpha_1 = 0.5$, $\alpha_2 = 0.3$ and $\alpha_3 = 0.2$ for the Gowalla data set.

5.3 Evaluation using Log-Likelihood

The training set for the Twitter data consists of all events recorded for the month of July, 2013. The test set for Twit-

	Individual		Event	
Model	Mean	Median	Mean	Median
Gaussian	-0.586	0.151	-0.242	1.357
GMM (C = 2)	-0.469	0.221	0.001	1.676
GMM (C = 4)	-0.474	0.279	-0.015	1.712
Fixed KDE	-1.025	-0.714	-0.869	-0.688
Adaptive KDE	-7.154	0.446	(∗)	-4.908
Population KDE	2.014	0.563	0.784	0.237
Mixture-KDE	**4.279**	**4.312**	**5.293**	**6.302**

Table 2: Average log-probabilities on the test data for individuals and events from the Twitter data set.

	Individual		Event	
Model	Mean	Median	Mean	Median
Gaussian	-1.513	0.008	-1.133	1.027
GMM (C = 2)	-1.532	0.308	-0.956	1.412
GMM (C = 4)	-1.522	-0.479	-0.958	1.471
Fixed KDE	-1.136	-0.749	-1.092	-0.742
Adaptive KDE	(∗)	-0.247	-3.288	1.355
Population KDE	3.388	1.237	4.021	0.923
Mixture-KDE	**6.599**	**6.296**	**6.568**	**2.619**

Table 3: Average log-probabilities on the test data for individuals and events from the Gowalla data set.

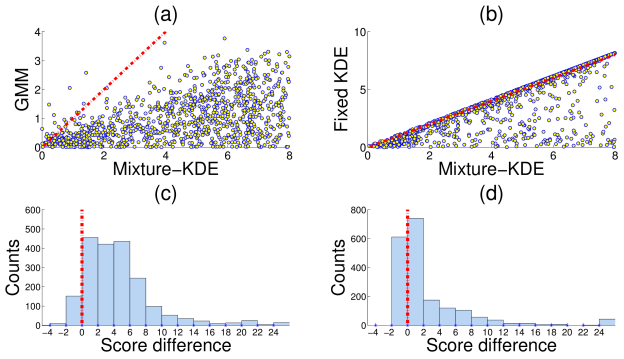

Figure 5: Upper plots (a) and (b): scatter plots for a sample of test set log-probability scores for Twitter individuals with (a) individual Gaussian mixtures ($C = 4$) versus the mixture-KDE, and (b) individual fixed-KDE versus the mixture-KDE. Lower plots (c) and (d): histograms of the score differences per event for the mixture-KDE minus the score of the corresponding model on the y-axis in the upper plot.

ter is all of the events in August 2013 for a randomly selected set of 2000 individuals, selected from the individuals that have at least 2 events in July. To create the Gowalla training set we used the data from the months of January to June, 2010. The test set for Gowalla is all of the events from the months of July to October, 2010 for a randomly selected set of 1000 individuals, selected from the individuals that have at least 2 events in the months of January to June, 2010. A validation set was generated for each of the Twitter and Gowalla data sets by setting aside approximately 37,000 and 25,000 randomly selected events from each training data set.

We built models on the data from our training data set and computed the test set log-probability score of the events in the test set, under each model. We report results both in terms of the mean and median log-likelihood per event, and the mean and median log-likelihood per individual (the latter giving equal weight to individuals, the former to events). The mean and median scores, for both individual and event-level scores, are shown in Tables 2 and 3 for the Twitter and Gowalla data sets respectively. A (∗) indicates that the test log-likelihood was not computed due to numerical underflow. The mixture-KDE model clearly outperforms all other methods, on both data sets, for all metrics, assigning significantly higher log-probability to the test events than any of the other modeling approaches. Figures 5 and 6 show the comparison between the mixture-KDE approach and the GMM (on the left) and fixed KDE (on the right) when looking at each individual separately, again clearly showing the improvement of the mixture-KDE approach over the other methods.

5.4 Evaluation using Simulated Identity Theft

We now compare the different models by using a simulated real-world application based on identity theft. Over 8 million people are victims of identity theft every year in the United States alone, with an annual cost exceeding 4 billion dollars and over 32 million hours spent by individuals to resolve the problem [8]. To simulate identify theft we replaced the geolocation events for an individual over a specific time-window with the geolocation events of a *different individual* and then tested the ability of our models to detect the resulting anomalous spatial patterns of events.

We used the Twitter data set for our experiment since the Gowalla data set did not show any significant differences for this problem between the different models (single Gaussian, mixtures of Gaussians, various forms of KDEs). We believe this may be due to the fact that our data set for Gowalla has fewer individuals than Twitter, and that these individuals use many of the same checkins, limiting the effectiveness of Gowalla data for detecting "identity switching".

Focusing on the Twitter data set, we defined the training set to be all events in the month of July, 2013. The test set consists of two types of event streams in August, 2013: events for normal "control" individuals and events with simulated identity theft. The control individuals are a randomly selected set of 950 individuals that have at least 2 events in July and at least 10 events in August. The individuals with simulated identity theft correspond to a set of 50 randomly selected individuals with at least 2 events in July. For each of these 50, we then replaced their real test data events in August, with the set of events from a different randomly selected individual, among individuals who have at least 10 events in August. In this manner, our test data has 50 event sets where the spatial distribution for each sets will in general look different (in a realistic manner, as if a different individual were using the Twitter account), compared to the event sets for the "true" individual in the training data from July.

To evaluate the different models we computed a surprise index S_i for each individual i, defined as the negative log-probability of individual i's events in the test data set rela-

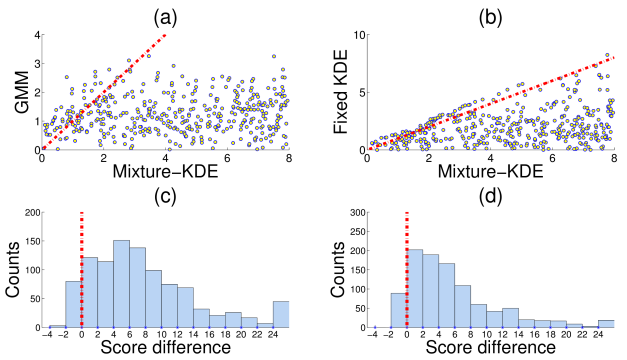

Figure 6: Upper plots (a) and (b): scatter plots for a sample of test set log-probability scores for Gowalla individuals with (a) individual Gaussian mixtures ($C = 4$) versus the mixture-KDE, and (b) individual fixed-KDE versus the mixture-KDE. Lower plots (c) and (d): histograms of the score differences per event for the mixture-KDE minus the score of the corresponding model on the y-axis in the upper plot.

tive to a model constructed on historical data:

$$S_i = -\frac{1}{n_i}\sum_{r=1}^{n_i} \log \hat{f}_i(e_i^r) \tag{8}$$

where e_i^r is the rth event in the test data set for individual i, and $\hat{f}_i$ is the density estimate for individual i constructed using the training data. The larger the surprise score S_i, then the more anomalous the events e_i^r are relative to the model $\hat{f}_i$. In these experiments we used all of the models that we used in the log-likelihood experiments as described earlier in the paper, except for the population model which is unable to generate rankings at the individual level.

We used a grid search on the validation set (defined in a similar manner to the training-test setup above) to determine various parameters of the models, where the parameter values were selected to optimize precision on the identity theft task. In general, optimizing for precision results in different parameter values for the various models compared to optimizing for likelihood. The priors of the Gaussian mixture model resulted in $n_0 = 0.1$ and $\sigma = 5$ kilometers. The optimal parameters for the individual KDE were estimated to be a fixed bandwidth of $h = 3$ kilometers and $k = 5$ for the adaptive method. The optimal mixture weights for the mixture-KDE model were $\alpha_1 = 0.9$ (the individual level), $\alpha_2 = 0.08$ (the region level), and $\alpha_3 = 0.02$ for the population model.

For each model, we then ranked all of the individuals in the test data by their surprise index S_i and computed precision relative to the known ground truth in terms of which individuals correspond to simulated identity theft and which to the controls. Table 4 shows the precision at 20, the fraction of simulated identity theft cases correctly detected in the top 20 ranked individuals. These precision numbers are the result of averaging over 50 different randomly generated test sets, using the methodology described earlier. The rows correspond to different models and the columns correspond to 3 different scenarios: computing the surprise-index per individual based on their first n events (in time) for each individual in the test set, with $n = 1, n = 5, n = 10$. Table 5 shows the same information for a "cold-start" scenario, where now test sets are generated for simulated identity theft and normal individuals (using the same general procedure as before) who are constrained to have between 2 and 5 events in their training data (compared to any number greater than or equal to 2 for the general case).

	$n = 1$	$n = 5$	$n = 10$
Gaussian	**0.612**	0.470	0.325
GMM (C=2)	0.320	0.240	0.160
GMM (C=4)	0.240	0.170	0.130
Fixed KDE	0.500	0.500	0.460
Adaptive KDE	0.432	0.309	0.274
Mixture-KDE	0.531	**0.747**	**0.816**

Table 4: Average precision (over 50 runs) for the top 20 ranked individuals in the test data, as a function of the number of observed test events n_t per individual.

	$n = 1$	$n = 5$	$n = 10$
Gaussian	0.379	0.341	0.240
GMM (C=2)	0.260	0.220	0.150
GMM (C=4)	0.240	0.190	0.150
Fixed KDE	0.330	0.370	0.320
Adaptive KDE	0.296	0.188	0.142
Mixture-KDE	**0.459**	**0.589**	**0.644**

Table 5: Average precision (over 50 runs) for the top 20 ranked individuals in the test data, as a function of the number of observed test events per individual, for "cold-start" individuals (as defined in the text).

The results in the two tables show that the mixture-KDE model dominates all of the other methods, with a Wilcoxon signed rank p-value of $p < 0.02$, except for the Gaussian model in the non-cold-case situation for $n = 1$. This may be due to the fact that for a simulated identity theft case, a sampled new event has a high probability of coming from a popular area.

The mixture-KDE model improves as it sees more events in the test set (as n increases from left to right in the table). However, the other methods all decrease in precision as n increases. On closer inspection we found that this was being caused by their sensitivity to false alarms, i.e., with more data points per individual there is a higher chance that a control individual (a false alarm) will have an event in the test data that is not close spatially to the individual's events in the training data, resulting in a high-surprise score and a high rank for that individual. The mixture-KDE is more robust to this type of variation, consistent with results earlier in the paper in terms of log-likelihood.

6. SCALABILITY AND ONLINE COMPUTATION

6.1 Scalability

Our experience suggests that kernel density models are quite scalable for two-dimensional data, and can likely be scaled to millions of individuals and hundreds of millions of events relatively easily. To compute the density of a new

	Individual		Event	
Model	Mean	Median	Mean	Median
Mixture-KDE	4.279	4.321	5.293	6.302
Online	**4.892**	**4.913**	**5.788**	**6.531**

	Individual		Event	
Model	Mean	Median	Mean	Median
Mixture-KDE	6.599	6.296	6.568	2.619
OnLine	**6.987**	**6.742**	**7.432**	**3.022**

Table 6: Predictive log-probability scores, averaged over individuals and events for a) Twitter (top table) and b) Gowalla (bottom table). "Online" is the online version of the Mixture-KDE model as described in the text.

point e, given a training data set E, we need to compute the contribution of each training point to e's density, as shown in Equation 3. In general, storing the N training data points requires $O(N)$ space and computing the density for a new event will result in time complexity $O(dN)$ where d is the dimension of the data (here $d = 2$). In our implementation of the KDE models for the results in this paper we used k-d trees, as described in [17], to further speedup our KDE implementation. This effectively computes only contributions from nearby points to e, based on a k-d tree partition of the 2-dimensional space, resulting in a significant reduction in computation time. We coded our algorithm for kernel density estimation in Java[2] and ran the code on an 8-core 2.4GHz Intel Xeon CPU with 8 hyper threads and 48 GB of RAM memory. Using one million events as training data points, the average time for computing the density of a new event is 8 milliseconds, making the model tractable for large data sets. Additional speed-ups could be achieved for example by distributed computation since the density contribution from different subsets of points are independent from one another. Hence, one can "split" the training data set into disjoint groups and aggregate the density contributions in parallel.

6.2 Online Prediction

The results presented up to this point in the paper have used a batch approach to training and testing. A useful feature of the kernel density approach, including the mixture-KDE, is that it is quite easy to implement an *online* version that can sequentially be updated as new events arrive in streaming fashion. When a new event e arrives we simply add it to the training set. For the adaptive bandwidth approach, every time the training set changes, we need to find the k neighbors of the new point, as well as potentially needing to find new neighbors for all N existing points. In practice, however, only a very small fraction of existing points will need to have their neighbors updated. Other parameters of the mixture-KDE model, such as mixture weights for the components, are likely to change relatively slowly over time, and can be periodically updated on validation subsets.

Tables 6(a) and 6(b) show predictive log-probability scores for the same training and test data used earlier for likelihood experiments, but now, each sequential test event is included in the training data in an online fashion before computing the log-probability of the next event. The online model shows a significant systematic improvement in predictive scores compared to the batch model, suggesting that online adaptation is beneficial with human location data. This certainly makes intuitive sense, as we expect individual behavior to be non-stationary and changing over time. In a practical application of an online model one would likely incorporate some downweighting (e.g., via exponential weighting) or windowing of events that are further back in time, allowing the model to adapt to changes in individual behavior.

[2]The code is available for download at http://www.datalab.uci.edu/resources/

7. CONCLUSIONS

In this paper we proposed and investigated a systematic framework for modeling human location data at an individual level using kernel density estimation methods. We found that adaptive bandwidth methods had distinct advantages for this type of data over fixed bandwidth methods. To address the problem of data sparsity at the individual level we introduced the idea of mixtures of kernel density estimates at different spatial scales. This allows smoothing of an individual's model towards the aggregate population model, motivated by the desire for better generalization to new data. Experimental results on both Twitter and Gowalla data sets systematically illustrated the benefits of our approach in terms of predictive power: kernel methods were systematically better than mixture models, adaptive bandwidth methods were systematically better than fixed bandwidth methods, and mixing individual and population estimates via the multi-scale mixture-KDE model outperformed all other approaches.

There are a number of extensions that could be further explored. One example is the discrete-location effect in data sets such as Twitter and Gowalla, namely that certain specific longitude-latitude locations are over-represented in the data. This suggests that additional gains in accuracy could be gained by modeling such locations as discrete delta-functions (with probability weights) in the kernel model, rather than using the simpler standard kernel approach. Another aspect we did not pursue in this paper is including time in our models—for many applications (such as identity theft detection) it would beneficial to have a distribution over time as well as space for individual events. Initial experiments in this direction suggestion that including time is not as straightforward as simply extending the spatial kernel densities from 2 to 3 dimensions—the temporal dimension, not surprisingly, has distinctly different characteristics than the spatial dimensions. Nonetheless we anticipate that spatio-temporal kernel density models can be developed in a relatively straightforward manner.

Acknowledgements

This work was supported by a gift from the University Affairs Committee at Xerox Corporation, by the National Science Foundation under award IIS-1320527, and by Office of Naval Research under MURI grant N00014-08-1-1015.

8. REFERENCES

[1] Twitter streaming api. *https://dev.twitter.com/docs/using-search*.

[2] J. Bithell. An application of density estimation to geographical epidemiology. *Statistics in Medicine*, 9(6):691–701, 1990.

[3] L. Breiman, W. Meisel, and E. Purcell. Variable kernel estimates of multivariate densities. *Technometrics*, 19(2):135–144, 1977.

[4] D. Brockmann, L. Hufnagel, and T. Geisel. The scaling laws of human travel. *Nature*, 439(7075):462–465, 2006.

[5] J. Chang and E. Sun. Location 3: How users share and respond to location-based data on social networking sites. In *Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media*, pages 74–80, 2011.

[6] C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In Proceedings of the 26th *AAAI*, pages 17-23, 2012.

[7] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1082–1090, ACM, 2011.

[8] Federal Trade Commission Identity theft survey report, 2006. *URL http://www.ftc.gov/reports/federal-trade-commission-2006-identity-theft-survey-report-prepared-commission-synovate*

[9] J. Cranshaw, R. Schwartz, J. I. Hong, and N. M. Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. In Proceedings of the Sixth *ICWSM*, pages 58-65, 2012.

[10] J. Cranshaw and T. Yano. Seeing a home away from the home: Distilling proto-neighborhoods from incidental data with latent topic modeling. In *CSSWC Workshop at NIPS*, 2010.

[11] N. Donthu and R. T. Rust. Estimating geographic customer densities using kernel density estimation. *Marketing Science*, 8(2):191–203, 1989.

[12] N. Eagle and A. S. Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[13] M. D. Escobar and M. West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[14] J. Fieberg. Kernel density estimators of home range: smoothing and the autocorrelation red herring. *Ecology*, 88(4):1059–1066, 2007.

[15] V. Frias-Martinez, V. Soto, H. Hohwald, and E. Frias-Martinez. Characterizing urban landscapes using geolocated tweets. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 239–248. IEEE, 2012.

[16] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi. Understanding individual human mobility patterns. *Nature*, 453(7196):779–782, 2008.

[17] A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In Proceeding of the 2003 SIAM International Conference of Data Mining, pages 203-211, 2003.

[18] S. Hasan, X. Zhan, and S. V. Ukkusuri. Understanding urban human activity and mobility patterns using large-scale location-based data from online social media. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing*, ACM, 2013.

[19] K. Joseph, C. H. Tan, and K. M. Carley. Beyond local, categories and friends: clustering foursquare users with latent topics. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 919–926. ACM, 2012.

[20] R. Lee, S. Wakamiya, and K. Sumiya. Urban area characterization based on crowd behavioral lifelogs over twitter. *Personal and Ubiquitous Computing*, 17(4):605–620, 2013.

[21] Z. Li, B. Ding, J. Han, R. Kays, and P. Nye. Mining periodic behaviors for moving objects. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1099–1108. ACM, 2010.

[22] K. P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

[23] A. Sadilek, H. A. Kautz, and V. Silenzio. Modeling spread of disease from social interactions. In *Proceedings of the Sixth AAAI International Conference on Weblogs and Social Media (ICWSM)*, pages 322-329, 2012.

[24] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Pervasive Computing*, pages 152–169. Springer, 2011.

[25] B. W. Silverman. *Density Estimation for Statistics and Data Analysis*. CRC press, 1986.

[26] P. Smyth and D. Wolpert. Linearly combining density estimators via stacking. *Machine Learning*, 36(1-2):59–83, 1999.

[27] L. Song, D. Kotz, R. Jain, and X. He. Evaluating next-cell predictors with extensive wi-fi mobility data. *Mobile Computing, IEEE Transactions on*, 5(12):1633–1649, 2006.

[28] S. J. Vaughan-Nichols. Will mobile computing's future be location, location, location? *Computer*, 42(2):14–17, 2009.

[29] J.-D. Zhang and C.-Y. Chow. igslr: personalized geo-social location recommendation: a kernel density estimation approach. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 324–333. ACM, 2013.