

Modeling Urban Behavior by Mining Geotagged Social Data

Emre Çelikten, Géraud C Le Falher, Michael C Mathioudakis

▶ **To cite this version:**

Emre Çelikten, Géraud C Le Falher, Michael C Mathioudakis. Modeling Urban Behavior by Mining Geotagged Social Data. IEEE Transactions on Big Data, 2016, pp.14. 10.1109/TBDATA.2016.2628398 . hal-01406676

HAL Id: hal-01406676

https://inria.hal.science/hal-01406676

Submitted on 1 Dec 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Modeling Urban Behavior by Mining Geotagged Social Data

Emre Çelikten
Computer Science
Aalto University
Helsinki, Finland
emre.celikten@aalto.fi

Géraud Le Falher
Inria, Univ. Lille
CNRS UMR 9189 – CRIStAL
F-59000 Lille, France
geraud.le-falher@inria.fr

Michael Mathioudakis
Helsinki Institute for Information Technology
Aalto University
Helsinki, Finland
michael.mathioudakis@hiit.fi

November 9, 2016

Abstract

Data generated on location-based social networks provide rich information on the whereabouts of urban dwellers. Specifically, such data reveal who spends time where, when, and on what type of activity (e.g., shopping at a mall, or dining at a restaurant). That information can, in turn, be used to describe city regions in terms of activity that takes place therein. For example, the data might reveal that citizens visit one region mainly for shopping in the morning, while another for dining in the evening. Furthermore, once such a description is available, one can ask more elaborate questions. For example, one might ask what features distinguish one region from another – some regions might be different in terms of the type of venues they host and others in terms of the visitors they attract. As another example, one might ask which regions are similar across cities.

In this paper, we present a method to answer such questions using publicly shared Foursquare data. Our analysis makes use of a probabilistic model, the features of which include the exact location of activity, the users who participate in the activity, as well as the time of the day and day of week the activity takes place. Compared to previous approaches to similar tasks, our probabilistic modeling approach allows us to make minimal assumptions about the data – which relieves us from having to set arbitrary parameters in our analysis (e.g., regarding the granularity of discovered regions or the importance of different features).

We demonstrate how the model learned with our method can be used to identify the most likely and distinctive features of a geographical area, quantify the importance features used in the model, and discover similar regions across different cities. Finally, we perform an empirical comparison with previous work and discuss insights obtained through our findings.

1 Introduction

Cities are massive and complex systems, the organisation of which we often find difficult to grasp as individuals. Those who live in cities get to know aspects of them through personal experiences: from the cramped bar where we celebrate the success of our favorite sports team to the quiet café where we read a book on Sunday morning. As our daily lives become more digitized, those personal experiences leave digital traces, that we can analyse to understand better how we experience our cities.

In this work, we analyze data from location-based social networks with the goal to understand how different locations within a city are associated with different kinds of activity – and to seek similar patterns across cities. To offer an example, we aim to automatically discover a decomposition

of a city into (potentially overlapping) regions, such that one region is possibly associated, say, with shopping centers that are active in the morning, while another is associated with dining venues that are active in the evening. We take a probabilistic approach to the task, so as to relieve ourselves from having to make arbitrary decisions about crucial aspects of the analysis – e.g., the number of such regions or the granularity level of the analysis. This probabilistic approach also provides a principled way to argue about the importance of different features for our analysis – e.g., is the separation of regions mostly due to the different categories of venues therein, or is it due to the different visitors they attract?

Our work belongs to the growing field of Urban Computing [1] and shares its motivation. First, as an ever-increasing number of people live in cities [2], understanding how cities are structured is becoming more crucial. Such structure indeed affects the quality of life for citizens (e.g., how much time we spend commuting), influences real-life decisions (e.g., where to rent an apartment or how much to price a house), and might reflect or even enforce social patterns (e.g. segregation of citizens in different regions). Second, switching perspective from the city to the people, the increasing amount of data produced by urban dwellers offer new opportunities towards understanding how citizens experience their cities. This understanding opens possibilities to improve the citizens' enjoyment of cities. For instance, by matching similar regions across cities, we could improve the relevance of out-of-town recommendations for travelers.

The data we use were generated on `Foursquare`, a popular location-based social network, and provide rich information about the offline activity of users. Specifically, one of the main functionalities of the platform is enabling its users to generate *check-ins* that inform their friends of their whereabouts. Each check-in contains information that reveals *who* (which user) spends time *where* (at what location), *when* (what time of day, what day of week), and doing *what* (according to the kind of venue: shopping at a grocery store, dining at a restaurant, and so on). The dataset consists of a total of **11.5 million `Foursquare` checkins**, generated by users around the globe (Section 2.1).

In the rest of the paper, we proceed as follows. For each city in the dataset, we learn a probabilistic model for the geographic distribution of venues across the city. The trained models associate different regions of the city with venues of different description. These venue descriptions are expressed in terms of data features such as the venue category, as well as the time and users of the related check-ins (Section 2.2). From a technical point of view, we employ a *sparse-modeling* approach [3], essentially enforcing that a region will be associated with a distinct description only if that is strongly supported by data.

Once such a model is learned for each city in our dataset through an expectation–maximization algorithm (Section 2.3), we examine how features are spatially distributed within a city, illustrating the insights it provides for some of the cities in our dataset (Section 3). Subsequently, we make use of the learned models and address two tasks.

- The first is to understand which features among the ones we consider are more significant for distinguishing regions in the same city (Section 4). Somewhat surprisingly, we find that *who* visits a venue has higher distinguishing power than other features (e.g., the category of the venue). This is a finding that is consistent across the cities that we trained a model for.

- The second is to find similar regions across different cities (Section 5). To quantify the similarity of two regions, we define a measure that has a natural interpretation within the probabilistic framework of this work. First, we discuss its properties and describe how one would employ them in an algorithmic search for similar regions across cities. Subsequently, we employ it on our dataset and find that the regions automatically detected in our model provide well-matching regions.

Having provided the results of our analysis, we compare our modeling approach to previously used approaches. Our empirical evaluation in Section 6 shows that our approach outperforms previous attempts [4, 5, 6] in terms of predictive performance as well as finding more distinctly described regions.

Finally, we review related work (Section 7), and discuss possible extensions and improvements of this work in Section 8. Our code and anonymized versions of our dataset will be made publicly available online[1].

2 Data & Model

2.1 Dataset

Our dataset consists of geo-tagged activity from `Foursquare`, a popular location-based social network that, as of 2016, claims more than 50 million users[2]. It enables users to share their current location with friends, rate and review venues they visit, and read reviews of other users. `Foursquare` users share their activity by generating *check-ins* using a dedicated mobile application[3]. Each check-in is associated with a web page that contains information about the user, the venue, and other details of the visit. Each venue is also associated with a public web page that contains information about it — notably the city it belongs to, its geographic coordinates and a category, such as *Food* or *Nightlife Spot*.

According to `Foursquare`'s policy, check-ins are private by default, i.e., they become publicly accessible only at the users' permission. This is the case, for example, when users opt to share their check-ins publicly via `Twitter`[4], a popular micro-blogging platform. We were thus able to obtain `Foursquare` data by retrieving check-ins shared on `Twitter` during the summer 2015. In order to have data for the whole year, we add data from a previous work [7] collected in the same way. We did not apply any filtering during data collection, but for the purposes of this work, we focus on the 40 cities with highest volume of check-ins in our data. The code for collecting the data is made publicly available on *github*[1].

The data from the 40 selected cities consist of approximately 6.7 million check-ins with 498 thousand unique users, in total. As a post-processing step, we removed the check-ins of users who contributed check-ins at less than five different venues, resulting in 6.3 million check-ins and approximately 284 thousand unique users (i.e. more than 7,000 unique users per city) in our working dataset. Details of the dataset can be found in Table 1.

The data represents three main types of entities, i.e., users, checkins, and venues, and we take a venue-centric view of the data. Specifically, we associate the following information with each venue.

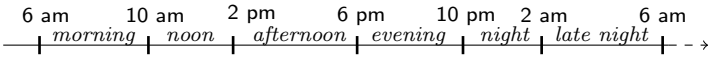
- A geographic `location`, expressed as a longitude-latitude pair of geographic coordinates.
- The `category` of the venue, as specified by `Foursquare`'s taxonomy (e.g., 'Art Gallery', 'Irani Cafe', 'Mini Golf'). If more than one categories are associated with one venue, we keep the one that is designated as the 'main category'.
- A list of all check-ins associated with this venue in the working dataset. Each check-in is a triplet that contains the following data:
  - The unique identifier of the `user` who performed it;

[1] At `http://mmathioudakis.github.io/geotopics/`.
[2] According to `https://foursquare.com/about/`.
[3] The Swarm application, `http://www.swarmapp.com`.
[4] `http://twitter.com`

Table 1: Number of check-ins and venues for the 18 (out of 40) cities with most data. They cover a large part of the world (Americas, Europe, Middle East and Asia).

City	Check-ins	Venues	City	Check-ins	Venues
Ankara	104,002	16,983	Mexico City	122,561	28,779
Barcelona	213,859	20,353	Moscow	397,008	51,871
Berlin	141,161	18,544	New York	1,007,377	75,721
Chicago	306,296	27,949	Paris	284,776	28,489
Istanbul	578,042	69,008	Rio de Janeiro	47,743	13,394
Izmir	190,303	20,529	San Francisco	432,625	22,384
Kuala Lumpur	147,103	22,594	Seattle	103,575	10,591
London	234,744	26,453	Washington	412,863	20,122
Los Angeles	367,624	36,086	Tokyo	214,493	38,117
For the 40 cities; 6,335,350 Check-ins and 749,097 Venues in total					

- The `day of the week` when the check-in occurred, expressed as a categorical variable with values *Monday*, *Tuesday*, ..., *Sunday*;
- The `time of the day` when the check-in occurred, expressed as a categorical variable with values *morning*, *noon*, ..., *late night*, defined as below



According to this view, each venue is a single data point described in terms of five **features** – namely `location`, `category`, `users`, `times of day` and `days of week`, and it takes a list of values for each feature. For the first two – `location` and `category` – the size of the list is always 1 – i.e., each venue is associated with a single location and a single category. Moreover, `location` values are *continuous* two-dimensional, while for all other features the values are *categorical*. For the categorical features, i.e., `category`, `times of day`, `days of week`, and `users`, we'll be using the term *dimensionality* to refer to the number of values they can take. For example, the dimensionality of `times of day` is always 6, that of `days of week` 7, that of `category` is about 700, and that of `users` has an average of more than $10,000$ within each city in the dataset.

2.2 Model Definition

Our analysis is based on a generative model that describes the venues we observe in a city. More precisely, each data point generated by the model corresponds to a single venue, and is associated with a list of values for each *feature* described in Section 2.1.

Remember that our goal is to uncover associations between geographic locations and other features of venues. Such associations are captured as k *topics* in the model – i.e., each data point is assigned probabilistically to one topic and different topics generate data venues with different distributions of features. As an example, one topic might generate venues (data points) that are located in the south of a city (feature: `location`) and are particularly popular in the morning (feature: `time of the day`), while another might generate venues that are located in the north of a city (feature: `location`) and predominantly restaurants, bars, and night-clubs (feature: `category`).

Specifically, to generate one data point, the model performs the following steps:

- Select one (1) out of k available topics $\{1, 2, \ldots, k\}$ according to a multinomial probability distribution $\theta = (\theta_1, \theta_2, \ldots, \theta_k)$. Let the selected topic be z.
- Generate a geographic location $loc = (x, y)$ from a bivariate Gaussian distribution with center $c = c_z$ and variance matrix $\Sigma = \Sigma_z$.

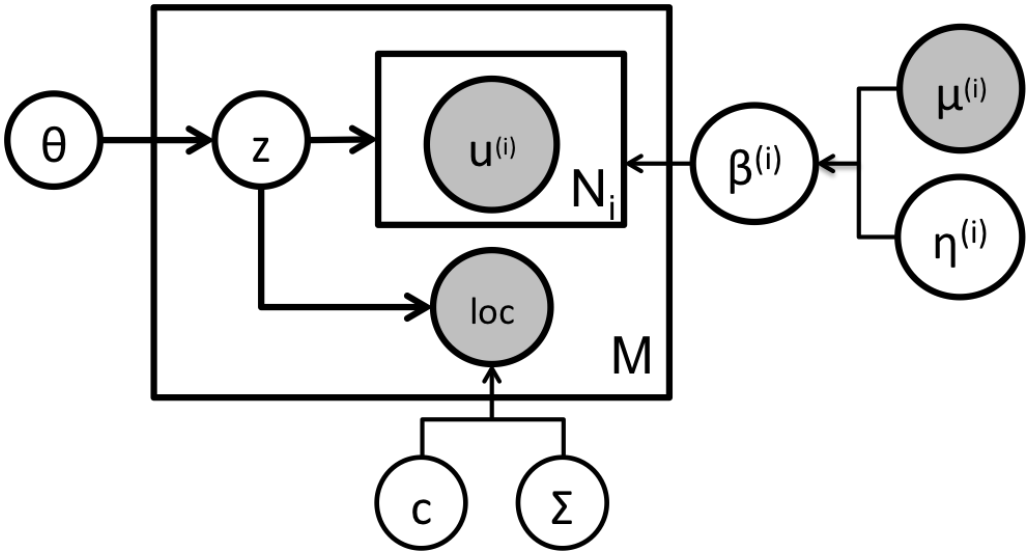


Figure 1: Generative Model. Note that only the i-th categorical feature is depicted.

- For the i-th categorical feature, generate a list $\mathbf{u} = \mathbf{u_i}$ of $N = N_i$ items, where N_i is specified as input for this data point. Each element in the list is selected randomly with replacement from a set $U = U_i = \{u_1, u_2, \ldots, u_m\}$ according to multinomial probability $\beta = \beta_z^{(i)} = (\beta_{z\,|1}^i, \beta_{z\,|2}^i, \ldots, \beta_{z\,|m}^i)$, with $\beta_{z\,|j}^i \geq 0$ and $\sum_{j=1..m} \beta_{z\,|j}^i = 1$.

The model is depicted in the plate diagram of Figure 1. The procedure described above is repeated M times to generate a dataset of size M (i.e., M venues). We stress that there is a different multinomial distribution $\beta = \beta_z^{(i)}$ for the z-th topic and i-th feature. We will be using non-subscript notation (β instead of $\beta_z^{(i)}$) when we might refer to any such distribution vector – and do the same with other notation symbols.

Moreover, we assume that, for the i-th feature, the probability distribution β_z^i is derived from a probability distribution μ^i and a deviation vector $\eta = \eta_z^i$ according to the following formula:

$$\beta_z^i \propto \exp(\mu^i + \eta_z^i). \tag{1}$$

Firstly, the distribution μ models the 'global' log-probability that the model generate an element $u \in U$. The model makes the assumption that all such probability distributions μ^i are equally likely a-priori (uniform prior). Secondly, the deviation vector η_z^i quantifies how much the distribution β_z of topic z deviates from global distribution μ^i. Following standard practice [3], the model makes the assumption that each value of vector η is selected at random with prior probability

$$\log p(\eta_z^{(i)}) = -\lambda \cdot |\eta_z^{(i)}| + \text{constant} \tag{2}$$

for some coefficient λ, provided as input. The model thus penalizes large deviations from 'global' vectors μ, thus leading to 'sparse' vectors η_z^i. The motivation for favoring sparce vectors η is that we wish to associate different topics with different distributions β only if we have significant support from the data. For the remaining parameters of the model, we make the following assumptions: all centers c_z are equally likely (uniform prior), the value of Σ_z has a standard Jeffreys prior [8],

$$\log p(\Sigma_z) = -\log \|\Sigma_z\| + \text{constant} \tag{3}$$

and all θ vectors are equally likely (uniform prior).

2.3 Learning

We learn one instance I of the model for each city in our dataset. Formally, learning corresponds to the optimization problem below.

Problem 1 *Given a set of data points $D = \{d_1, d_2, \ldots, d_{_M}\}$, and the generative model I, find global log-probabilities μ, topic distribution θ, deviation vectors η, bivariate Gaussian centers c and covariances Σ that maximize the probability*

$$L = p(\{\mu\}, \theta, \{\eta\}, \{c\}, \{\Sigma\} \mid D, I).$$

We perform the optimization by partitioning the dataset into a training and test dataset and following a standard validation procedure. During training, we keep k and λ fixed and optimize the remaining parameters of the model on the training dataset (80% of all data points). We then evaluate the performance of the model on the test dataset (20% of all data points), by calculating the log-likelihood of the test data under the model produced during training. We repeat the procedure for a range of values for k and λ to select an optimal configuration.

A max-likelihood vector μ is computed once for each feature from the raw relative frequencies of observed values of that feature in the dataset. For fixed k and λ, the maximum-likelihood value of the remaining parameters can be computed with a standard expectation–maximization algorithm. The steps of the algorithms are provided below,

E-Step

$$\begin{aligned} \log q_d(z) \quad := \quad & \sum_{i=1..m} n_d^{(i)} \log \beta_z^{(i)} + \log \mathcal{N}(loc_d; c_z, \Sigma_z) + \\ & + \log \theta_z + constant;\ \sum_z q_d(z) = 1 \end{aligned}$$

M-Step

$$\begin{aligned} \theta_z \quad &:= \quad \sum_{d \in D} q_d(z),\ \text{normalized to } \sum_z \theta_z = 1 \\ c_z \quad &= \quad \frac{\sum_d q_d(z) \cdot loc_d}{\sum_d q_d(z)} \\ \Sigma_z \quad &:= \quad \frac{\sum_{d \in D} q_d(z)(loc_d - c_z)(loc_d - c_z)^T}{\sum_{d \in D} q_d(z) + 4} \\ \{\eta_z^{(i)}\} \quad &:= \quad \underset{\{\eta_z^{(i)}\}}{\text{argmax}} \sum_{d \in D} q_d(z) \cdot n_d^{(i)} \cdot \log \beta_z^{(i)} - \lambda \cdot |\eta_z^{(i)}|, \end{aligned}$$

with $n_d^{(i)}$ the number of times the i-th element appears in data point d, and the latter optimization (for η values) performed numerically.

To optimize with respect to k and λ, we experiment with a grid of values and select the pair of values with the best performance on the test set. We found that $\lambda \approx 1$ worked well for all cities we experimented with, while improvement reached a plateau for values of k near $k \approx 50 - 55$. Figure 2 shows the training plots for the city of Paris; similar patterns are observed for the other cities in the dataset.

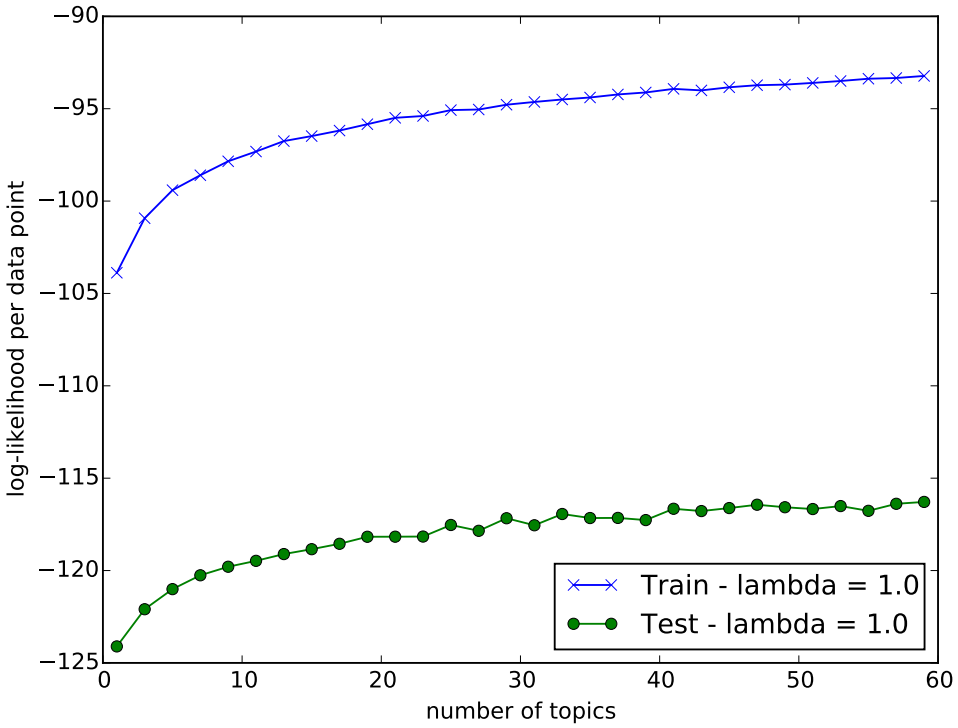


Figure 2: Log-likelihood per data point on the training and test datasets of Paris, for $\lambda = 1$ and increasing k.

2.4 Practical Issues and Discussion

In training, we came across memory issues during the estimation of model variables related to feature `users` – particularly for the corresponding multinomial deviation vector η. We believe that was due to the large dimensionality of the `users` feature. Indeed, on average, our data contain activity of more than $10,000$ users per city – while the dimensionality of features `time of the day`, `day of the week`, `category` is much lower 6, 7, and ≈ 400, respectively.

To deal with the issue, we use SVD to reduce the dimensionality of feature `users`. Our goal is to partition the users $\{u_1, u_2, \ldots, u_M\}$ that appear in one city into d groups $U_1, U_2, \ldots, U_d$, in such a way that U_i groups together users that check-in at the same venues. SVD captures nicely such semantics and this property has been used in many settings (e.g., latent semantic indexing [9]). Once the partition is produced, we treat all users in group U_i, $i = 1..d$, as the same user (a 'super-user'), thus reducing the dimensionality of feature `users` to d.

Specifically, for one city, we consider the users-venue matrix M. The (i, j) entry of matrix M contains the number of check-ins observed for the i-th user at the the j-th venue in the city. Subsequently, we use SVD to compute the d right-eigenvectors $\mathbf{v_1}, \mathbf{v_2}, \ldots, \mathbf{v_d}$ of M. Note that the dimensionality of each such eigenvector $\mathbf{v}$ is equal to the dimensionality of the `users` feature for a given city. Finally, we partition the users into d groups: we assign the i-th user to group

$g \in \{0, 1, \ldots, d\}$ such that

$$g = \underset{j \in \{1,2,\ldots,d\}}{\text{argmax}} \{\mathbf{v_1}[i], \mathbf{v_2}[i], \ldots, \mathbf{v_d}[i]\}$$

This provides naturally the partition $U_1, U_2, \ldots, U_d$ we aimed to identify.

Note that in our experimentation we employed SVD for a large range of dimensionality values $d \in [10^0, 10^3]$ and found that increasing d beyond $d = 1000$ did not significantly improve the model performance. We thus settled for $d = 1000$.

Flexible Number of Features

The model defined in Section 2.2 is employed on data that exhibit a particular set of categorical features, as described in Section 2.1. Nevertheless, this does not preclude that the model be applied on data that exhibit a different (smaller or larger) number of categorical features. Suppose, for example, that one wishes to train a model that takes into account only the location and category of venues, but not the check-in information (time and user of the check-in). A simple way to achieve this is to provide a single data point for each venue. The data point would contain the venue information (location and category) and some placeholder value for all the other features. This allow us to learn a model only for the category and location of venues. In practice, then, one could simply remove non-available features from the model – i.e., one would have vectors $\beta^{(i)}$ only for the features at hand (see Figure 1) and train on them.

Location

The geographic component of the model we use does not match the notion of neighborhood as conceived in a more administrative sense, e.g., as a set of roads or other boundaries that enclose a geographical area. However, our goal in the paper is *not* to discover such neighborhoods, but rather to discover geographical patterns that represent well, in a probabilistic sense, the activity observed in the data at hand. The geographical patterns we look for are associated with more loosely-defined regions, represented by two-dimensional Gaussians. Moreover, note that the model allows regions to overlap and at the same time accommodate a number of categorical features.

In practice, this means that we might discover *topics* in the data that have highly overlapping geographical regions, but are differentiated based on other features. For example, in training the model, we might discover that there are two topics in the same region – one consisting of venues that operate early in the morning, and of venues that operate late in the evening. This is a basic difference from related work such as [4], that aims to directly partition the area of a city into non-overlapping regions.

Dataset Biases

One challenge that arises in using Foursquare checkins as our dataset is how to assess whether the activity (checkins) recorded in the data are representative of the actual activity in the city. In other words, are Foursquare data representative of what people actually do in the city? Such an assessment is the material of future work that is beyond the scope of this paper, as it would require additional technical tools and additional ground-truth data. Until then, we are careful about the claims we make about the empirical findings of our method: these empirical findings solely represent the urban activity contained within the Foursquare dataset.

3 Likely and distinctive feature values

Following the learning procedure defined in Section 2, we learn a single model for each city in our data. The value of these model instances is that they offer a principled way to answer questions that we cannot answer from raw data alone. To provide an example, suppose that an acquaintance from abroad visits our city and asks "if I stay at location l of the city, what is the most common venue category I find there?". Raw data do not provide an immediate answer to the question. They do allow us, for example, to provide answers of the form "within radius r from l, the most common venue category is c with n out of N venues". However such answers would depend on quantity r, that was not provided as input – and would probably never be, if our friend does not have any knowledge about the city. Selecting too small a value for r (e.g., a few meters), would make the answer sensitive to the exact location l; selecting too large a value for r (e.g., a few kilometers), would make the answer insensitive to the exact location l.

The unsupervised learning approach we take allows us to avoid such arbitrary choices in a principled manner. It learns topics associated with Gaussian distributions as regions, whose size is learned from the data; and under a model instance I, it allows us to answer our friend's question by simply considering the probability

$$p(\text{category} = c|\text{location} = l; I)$$

that at the given location we find a venue of category c, and answering with the category that is associated with the highest probability value.

Given such model instances, we explore the geographical distribution of venues for the corresponding cities. Due to space constraints, we provide only a few examples here and provide a complete list of findings from this section on the project's webpage[1].

Most likely feature value Suppose a venue is placed at a given location $loc = (x, y)$ – what is the category most likely associated with it? In other words, we are asking for the category that maximizes the expression

$$p(\text{category} = c \mid \text{location} = l; I) \tag{4}$$

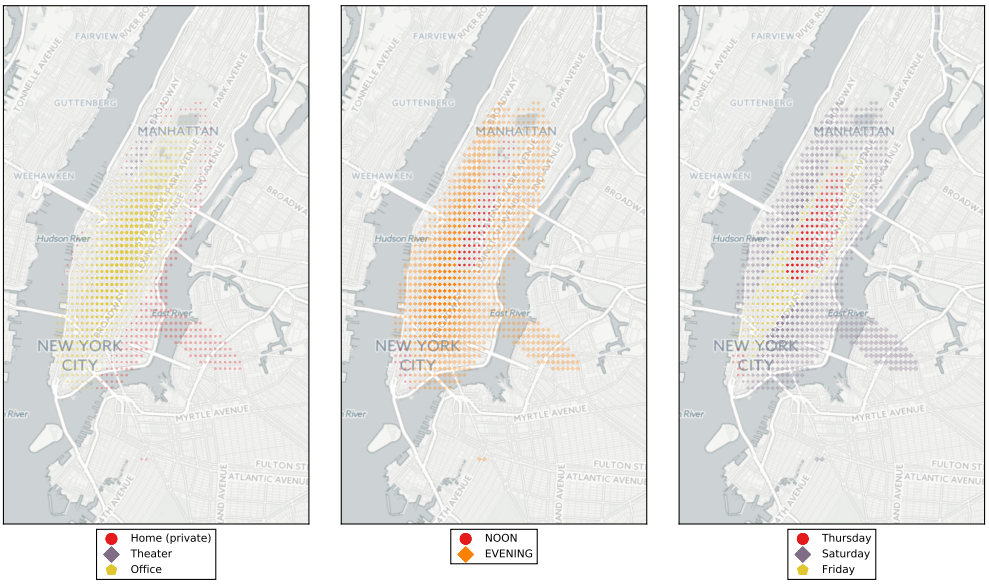
that we just discussed above. We use our model to answer this question for *New York*. The results are shown in Figure 3(a). We can ask a similar question for the remaining categorical features represented in our model. For example, suppose a venue is placed at a given location, what is the most likely time a check-in occurs at that venue? The results for *New York* are given in Figure 3(b).

Most distinctive feature value Looking again at Figure 3(b), we see that *evening* check-ins dominate the map: for many locations in Manhattan, a venue placed there is most likely to receive a check-in during the evening. One simple explanation for this is that overwhelmingly many check-ins in our data for this city occur in the evening, as we see in table 2.

Table 2: New York City check-ins in thousands

morning	noon	afternoon	evening	night	late night
106	219	240	333	118	25

Nevertheless, some areas of the city are more highly associated with morning check-ins than others. In formal terms, for a given location, let us consider the ratio of the probability that the `time of day` a check-in occurs takes a particular value ('morning', 'noon', etc) over the probability that a check-in takes that value over the entire city. Arguably, that ratio expresses how distinctive



(a) category (b) time of day (c) day of week

Figure 3: Most likely `category` and checkin `time of day, day of week` across Manhattan. Note that the transparency of each point is equal to the probability that a venue is located at that point.

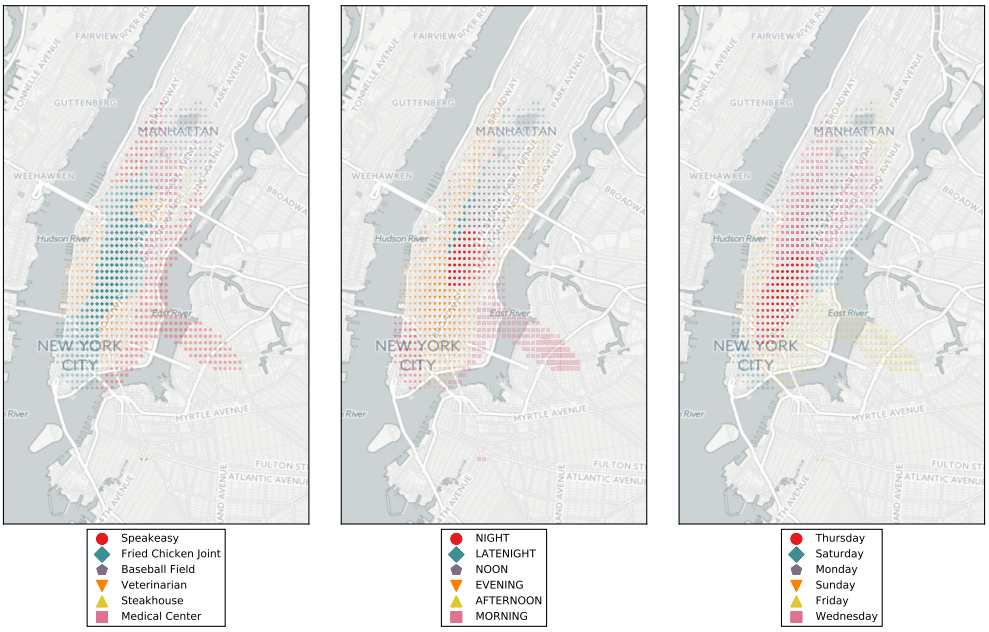
that value is for this particular location. Formally, it is expressed as follows.

$$\frac{p(\text{time of day} = t \mid \text{location} = l; I)}{p(\text{time of day} = t \mid I} \tag{5}$$

For example, suppose that a venue at a particular location *loc* receives a check-in in the morning with probability 30%; and that on average across the city venues, a venue would receive a check-in in the morning with probability only 1%. Then, we can say that location *loc* is associated with venues that are distinguished for the relatively high frequency of morning check-ins. Figure 4(b) indicates the most distinctive time of check-in across New York. We can ask a similar question for other categorical features. For example, *what is the most distinctive category for the same location?* The results for *New York* are shown in Figure 4(a).

4 Feature analysis

In the previous section, we employed the model instance of a city to ask questions about the geographical distribution of a given feature. In this section, we study the importance of each feature in distinguishing the different topics that define a model instance. To further illuminate the question, let us remember that the model is built upon a set of features $\{X\}$ and that the distribution of each feature is allowed to vary across topics. In plain terms, the question we ask is the following: *if we were forced to fix the distribution of feature X across topics, how much would that hurt the predictive performance of the model?* This question is important, not just for the



(a) category (b) time of day (c) day of week

Figure 4: Most distinctive `category` and checkin `time of day, day of week` across Manhattan. The transparency of each point is equal to the probability that a venue is located at that point.

purposes of feature selection in case one wanted to employ a simpler model, but also because it allows us to suggest on what features future work should be focused, in order to understand better urban activity.

To be more specific, let us first consider the categorical features in our model: `category`, `users`, `time of the day`, `day of the week`. Within each topic of a model instance, the distribution β of each of the aforementioned features deviates from the overall distribution μ by a vector η (Equation 1). To quantify to which extent a single categorical feature X contributes to the variance across topics, we perform a simple ablation study. That is, we select the same training set as for our best model, keep the value of parameters k and λ, and train a model instance by fixing the η of feature X equal to zero.

$$\text{for categorical feature} X: \text{ set } \eta = 0$$

Subsequently we compare the log likelihood of both models (the best one and the ablated one) over all the data for the given city and measure the log-likelihood drop between the two. The higher this drop, the higher the importance of that feature in explaining the variance across topics.

We perform a similar procedure for the `location` feature. Specifically, for the ablated model, we replace the bivariate Gaussian distribution G_z associated with each model with a distribution G_0 that remains fixed across topics. G_0 is set to be the mixture of Gaussians G_z across topics z, with mixture proportions equal to topic proportions θ_z.

$$\text{for } \texttt{location}: \text{ set } G_0 = \text{mixture}\{G_z, \theta_z\}$$

Results are summarized for all cities in Figure 5. The immediate observation is that `users` prominently stand out as a feature and that this is consistent across all cities. This suggests that,

at least for the urban activity represented in our dataset, **who** visits a venue has a more important role to play in distinguishing different venues, than where the venues are located and when they are active. At this point, we should also stress that there is very little overlap in the users that

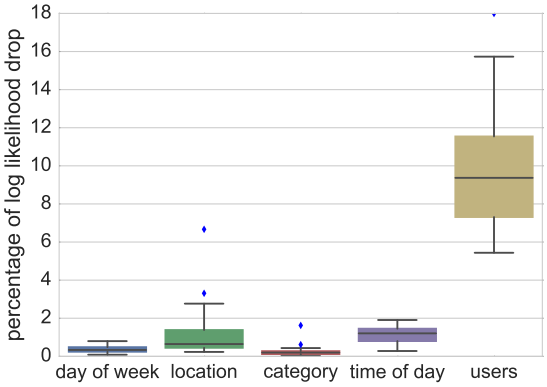


Figure 5: Contribution of each feature to the data likelihood. The boxplots summarize how much the log likelihood drops once we fix the distribution of a single feature across topics. We observe a consistent behavior across cities, in that the variance of `users` across topics is most important for the predictive performance of the model.

check-in at different cities (see Figure 6). Among the remaining features, `location` and `time of the day`, are consistently more important across cities than `day of the week` and `category`.

5 Similar regions across cities

In this section, we address the task of discovering similar regions across different cities. Addressing this task would be useful, for example, to generate touristic recommendations for people who visit a foreign city or generally to develop a better understanding of a foreign city based on knowledge from one's own city. Specifically, we are given the trained models of two (2) cities as input and aim to identify one region from each city so that a similarity measure for the two regions is maximized. Following the conventions of this work, each region is spatially defined in terms of a bivariate Gaussian distribution. Moreover, in the interest of simplicity, we consider only cases where the similarity measure concerns a single categorical feature. In what follows, we devise a measure to quantify the similarity of two regions and present an algorithm to find similar regions according to that measure. Then, we describe some aggregate observations from employing this measure on our dataset.

5.1 Similarity Measure

We start by defining a similarity measure `jointsim` that has a natural interpretation in our setting. It quantifies (i) how similar the venues of two regions are on average, according to the corresponding models, but also (ii) how many venues they contain, according to the model. The rationale for the second point is that one wishes to identify regions that have large probability under the model and avoid identifying pairs of tiny regions that are 'spuriously' similar.

Let us now define formally the `jointsim` measure, which operates under the following settings. We are given two model instances, I_1 and I_2, each corresponding to a city in our data. As explained in earlier sections of the paper, each model instance describes the distribution of venues in its respective city, along with distributions over every features. Moreover, we are given two bivariate

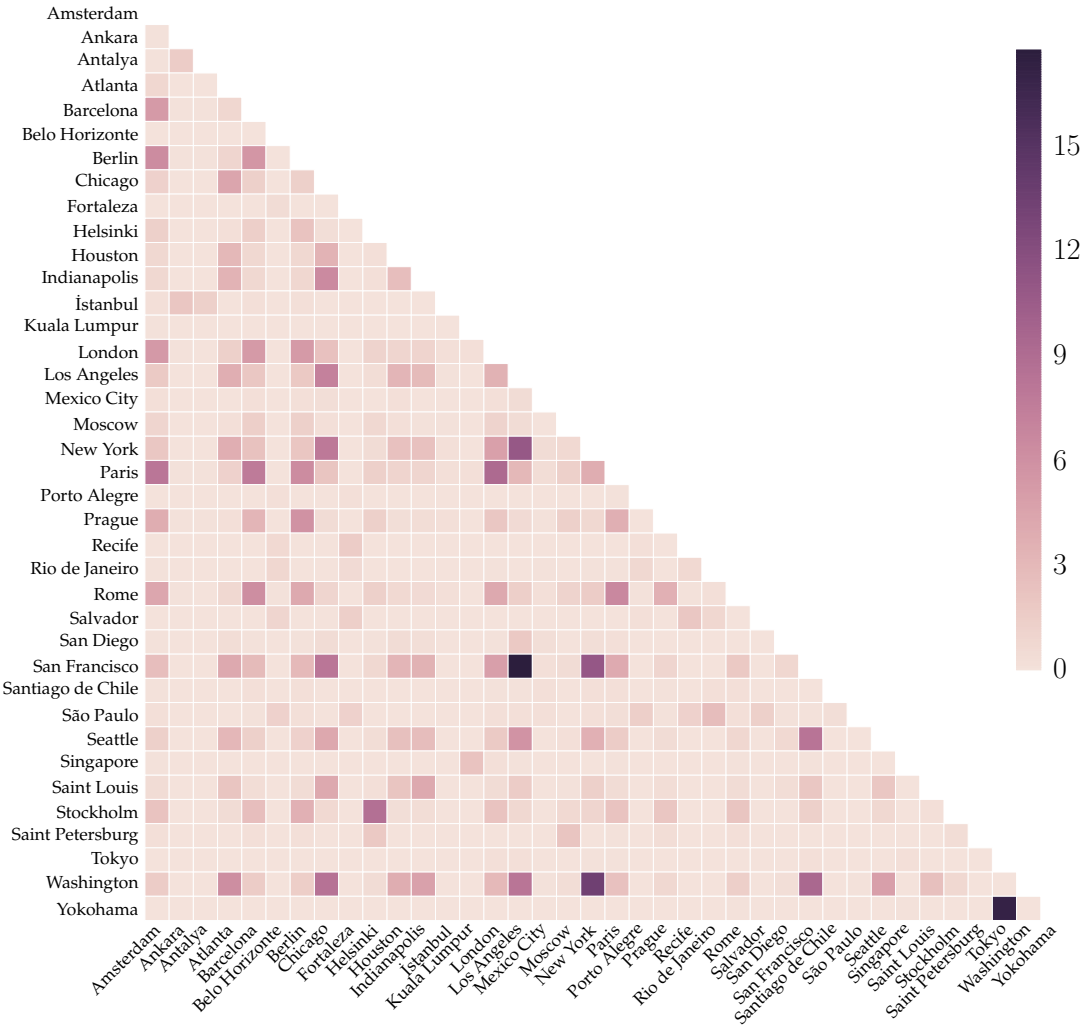


Figure 6: Jaccard coefficient between sets of users that appear in different pairs of cities.

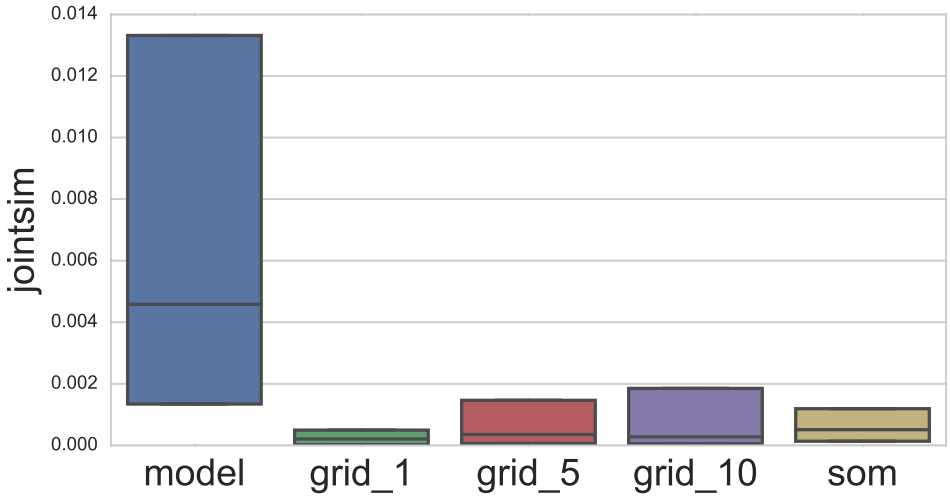


Figure 7: The `jointsim` values for the best-matching regions returned by `GeoExplore` across all pairs of cities, for different sets of base-regions (**model**, **grid-α** and **SOM**).

Gaussian distributions, G_1 and G_2, each defining one geographic region in the respective model I_1 and I_2. To define the similarity between the two regions with respect to feature X (e.g., $X =$ `category`), we first define a random procedure $\mathcal{R}(I, G, X)$. Given a model instance I and a region G, random procedure $\mathcal{R}$ generates a pair of values (x, s), with $x \in Dom(X)$ and $s \in \mathbb{R}^+$. It is defined as follows.

Definition 1 ($\mathcal{R}(I, G, X)$) *Perform the following steps.*

- *Generate a data point $d \sim I$, with location $loc = l$ and feature X value $X = x$.*
- *Let $s = \mathcal{N}(l)$ be the probability density of G at location l.*
- *Return the pair (x, s).*

In plain terms, random procedure $\mathcal{R}(I, G, X)$ picks a random data point from model instance I, and associates it with (1) its value $X = x$ for feature X and (2) the density of region-defining Gaussian G. Similarity measure `jointsim` then answers the question: 'if random procedure $\mathcal{R}$ is applied on model I_1 and region G_1, on one hand, and model I_2 and region G_2, on the other, with respective output pairs (x_1, s_1) and (x_2, s_2), what is the expected value of the expression

$$s_1 \cdot s_2 \cdot \delta_{x_1 x_2},$$

over possible invocations of procedure $\mathcal{R}$'? In the expression above, $\delta_{x_1 x_2}$ is the Kronecker delta – equal to one (1) only when $x_1 = x_2$ and zero (0) otherwise. In other words, `jointsim` combines the answer of the following two questions: if we consider two random venues, one from each model I_1 and I_2, then (1) what is the probability that they have the same value for feature X, and (2) if they do, how much are their locations covered by regions G_1 and G_2? The measure is formally defined below:

Definition 2 (`jointsim`) *Let (x_1, s_1) and (x_2, s_2) be the output of a single invocation of random procedure $\mathcal{R}(I_1, G_1, X)$ and $\mathcal{R}(I_2, G_2, X)$, respectively. Similarity `jointsim` is defined as the expected value*

$$\texttt{jointsim}(G_1, G_2; I_1, I_2, X) = E_{\mathcal{R}}[s_1 \cdot s_2 \cdot \delta_{x_1 x_2}] \tag{6}$$

We now proceed to provide an analytical expression for `jointsim`. First, in the interest of simplicity, we fix model instances I_1, I_2 and feature X and write $\texttt{jointsim}(G_1, G_2) = \texttt{jointsim}(G_1, G_2; I_1, I_2, X)$. Let us also write $\psi_i(x, l)$ to denote the **joint** probability that model I_i ($i \in \{1, 2\}$), generates a data point with feature value $X = x$ and location l,

$$\psi_i(x|l) = p(X = x, loc = l|I_i),$$

which can be expanded to

$$\begin{aligned}
\psi_i(x|l) = & \quad p(X = x, loc = l; I_i) \\
= & \quad p(X = x, loc = l|I_i) \\
= & \quad \textstyle\sum_{z=1,\ldots,k} p(X = x, loc = l, topic = z|I_i) \\
= & \quad \textstyle\sum_{z=1,\ldots,k} \mathcal{N}_z(l)\beta_z(v)\theta_z
\end{aligned}$$

where $\mathcal{N}_z$ denotes the Gaussian probability density function for the Gaussian distribution associated with the z-th topic, $z = 1, \ldots, k$, of model I_i. Finally, for locations l_1 and l_2, let us write $h(l_1, l_2)$ for the inner-product function

$$h(l_1, l_2) = \sum_{x \in Dom(X)} \psi_1(x, l_1)\psi_2(x, l_2). \tag{7}$$

With the notational conventions above, we now provide an analytical expression for `jointsim`.

$$\texttt{jointsim}(G_1, G_2) = \int_{l_1, l_2} \mathcal{N}_1(l_1)\mathcal{N}_2(l_2)h(l_1, l_2)dl_1 dl_2 \tag{8}$$

Note that, in equation (8), $\mathcal{N}_1$ and $\mathcal{N}_2$ denote the probability density functions of Gaussians G_1 and G_2, respectively. Moreover, we approximate the integral of equation (8) with a discrete sum approximation over a 100×100 grid on each model.

Having defined our similarity measures, let us consider the corresponding maximization problem.

Problem 2 *Consider two models I_1, I_2, and feature X. Identify two bivariate Gaussians G_1, G_2, such that `jointsim`(G_1, G_2) is maximized.*

Problem 2 has two intuitive properties:

- all other things being equal, it favors regions G_1 and G_2 that cover areas with high probability mass according to models I_1, I_2,
- all other things being equal, it favors regions G_1 and G_2 that cover areas where data points are associated with similar feature distributions.

To put in plain terms, Problem 2 favors regions that correspond to areas of **many** *and* **similar** data points. This is seen also in equations (7) and (8). Indeed, they take into account the similarity of data points (venues) in terms of feature X at locations within the two regions, but they also take into account the probability mass assigned to those regions by models I_1 and I_2. This makes Problem 2 appropriate to consider in cases when one does not have prior restrictions or preferences for the candidate regions that would comprise an optimal solution and at the same time would like to avoid spurious solutions, i.e., pairs of regions that are very similar, but that cover too small probability mass of the respective models.

5.2 Best-First Search for `jointsim`

To the best of our knowledge, the problem of identifying similar regions across cities has not been defined formally before within a probabilistic framework. In this section, we also propose the first algorithm, GeoExplore, to approach Problem 2. Algorithm GeoExplore follows a typical *best-first* exploration scheme and comprises of the following two phases. Its first phase consists of one step: it begins with a candidate collection of regions $\mathbb{G}_1$ and $\mathbb{G}_2$ for each side (let us call them '**base regions**') and evaluates all pairwise similarities $\texttt{jointsim}(G_1, G_2)$, for $G_1 \in \mathbb{G}_1$, $G_2 \in \mathbb{G}_2$. Its second phase consists of the remaining steps: it explores the possibility to improve the currently best `jointsim` measure by combining previously considered regions. This is motivated by the fact that Problem 2 favors regions of larger probability mass, and therefore combined regions might yield better `jointsim` values.

Pseudocode for GeoExplore is shown in Algorithm 1. It repeats a three-steps *Retrieve* - *Update* - *Expand* procedure for each step. During *Retrieve*, the algorithm retrieves the next candidate solution for Problem 2. Each candidate solution comes in the form of a triplet; two Gaussians and their `jointsim` score. During *Update*, the algorithm updates the score of the best-matching pair, if a better pair has just been retrieved. Finally, during *Expand*, the algorithm expands the latest retrieved Gaussians to form Gaussians from each side, and thus new candidate solutions for Problem 2. Subroutine **expand(G, I)** operates as follows:

- When G is not specified (i.e., G = NULL in Algorithm 1), then **expand** simply returns the set of base regions $\mathbb{G}$. This case occurs during the first expansions only. Moreover, each base region $G_i \in \mathbb{G}$ is associated with positive weight w_i, either specified as input, or set to $1/\|\mathbb{G}\|$ by default.
- When $G = G_i$ for some $G_i \in \mathbb{G}$, then **expand** returns the set of Gaussians $\{G_i\} \bigcup \{G_i \cup G_j; G_j \in \mathbb{G}, j \neq i\}$, where $G_i \cup G_j$ is defined as the best Gaussian-fit to the mixture model determined by $[G_i, G_j]$, with respective proportions (w_i, w_j). The intuition for this step is that we expand the best-performing pair of Gaussians by combining them with other base Gaussians.
- In a recursive fashion, when $G = G_i \cup G_{i'} \ldots G_{i''}$, then **expand** returns the set of Gaussians $\{G\} \bigcup \{G \cup G_j; G_j \in \mathbb{G}, j \neq i, i', \ldots, i''\}$ each defined as the best Gaussian-fit to the mixture model determined by $[G, G_j]$, with respective proportions $(w_i + w_{i'} + \ldots + w_{i''}, w_j)$.

Note that in practice, to prevent the algorithm from exploring the combinatorially large space, we terminate GeoExplore after a number R of **while** loops. If k_1, k_2 are the number of topics in the two model instances, we perform $O(k_1 + k_2)$ expansions as well as $O(k_1 \cdot k_2)$ `jointsim` evaluations in each loop. If the respective running-time costs of the operations are $c_1 = c_{expansion}$ and $c_2 = c_{\texttt{jointsim}}$, the total running time of the algorithm is $O(R \cdot ((k_1 + k_2)c_1) + k_1 k_2 c_2)$.

5.3 Empirical performance

We employed GeoExplore with $R = 5$ expansions on all pairs of cities in our dataset (Section 2.1) and report the `jointsim` values returned for different base region collections. Specifically, we experimented with the following collections of base regions:

Model We simply used as collections $\mathbb{G}_1$, $\mathbb{G}_2$ the Gaussians associated with the respective topics in the input model I_1, I_2, and assigned to each Gaussian a weight equal to the respective θ parameter value found in the model.

Grid-a We used as collections $\mathbb{G}_1$, $\mathbb{G}_2$ Gaussians that covered in a grid-like fashion the respective cities, each with size equal to $1/a$ the size of the median size of Gaussians found in model I_1, I_2.

Algorithm 1 GeoExplore

Input: models I_1 and I_2, base regions $\mathbb{G}_1$, $\mathbb{G}_2$
Output: Best Pair G_1, G_2

INITIALIZE
BestG$_1$ = NULL, BestG$_2$ = NULL, BestScore = 0
H = MaxHeap()
# Initialize max-heap with empty solution, zero score
Push(BestG$_1$, BestG$_2$, BestScore) to H

while H is Not Empty **do**
  # RETRIEVE top solution in max-heap
  Pop (G$_1$, G$_2$, Score) from H
  # UPDATE best solution
  if Score > BestScore **then**
    BestG$_1$ = G$_1$, BestG$_2$ = G$_2$, BestScore = Score
  end if
  # EXPAND retrieved solution
  for G$_a$, G$_b$ in **expand(G$_1$)**, **expand(G$_2$)** $\neq$ G$_1$, G$_2$ **do**
    Score = `jointsim`$(G_a, G_b | I_1, I_2)$
    Push (G$_a$, G$_b$, Score) to H
  end for
end while
return best_pair

Table 3: The improvement rate for each value of R, i.e., the fraction of times that the algorithm found an improved solution out of all the times it reached that round.

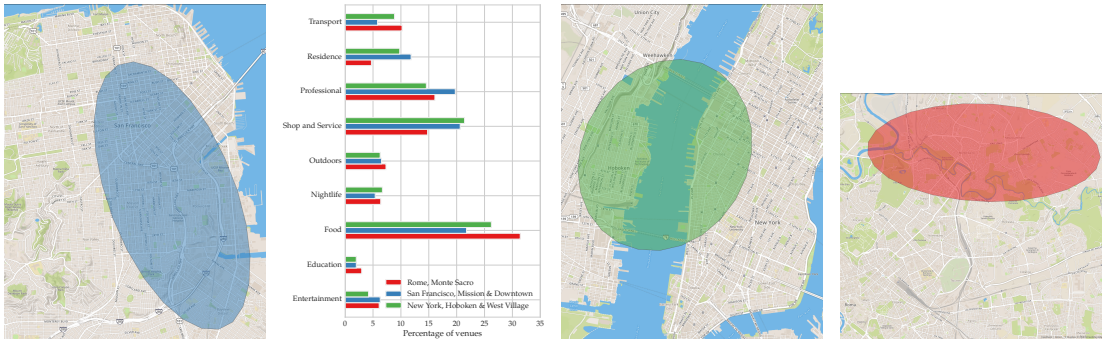
R	1	2	3	4	5
improvement rate	53.0%	2.7%	2.3%	1.8%	1.6%

SOM We employed the method of Self Organized Maps [6] to generate a set of closed regions that cover each city. Subsequently, for each such region, we find the smallest bivariate Gaussian so that the entire region is enclosed within two standard deviations of the Gaussian in each direction. We use these Gaussians as collections $\mathbb{G}_1$, $\mathbb{G}_2$ and employ GeoExplore.

The results are shown in Figure 7. We observe that using the model Gaussians as our base regions leads to better performance compared with the grid baselines.

The reason we stopped expansions at $R = 5$ was that beyond that point we found it was very unlikely to obtain an expansion that led to an improvement – see Table 3.

Finally, in Figure 8, we show two examples of pairs of regions with high `jointsim` score, which we discovered by running GeoExplore. In both cases, the first region is in San Francisco. As we can see in blue on 8(a), it covers the Mission district and a part of the downtown area. Among all the other cities in our data, GeoExplore found that the region 34 of New York (shown in green on 8(c) and covering the city of Hoboken and West Village was most similar to it in the sense of `jointsim`, as well as a region in Rome that covers the district of Monte Sacro (shown in red on 8(d)). As explained earlier, these regions are not spuriously small and since we measure similarity with respect to the *category* feature, it is not surprising that they have similar distributions of



(a) San Francisco: Mission and Downtown

(b) Distributions of Venue Categories.

(c) New York: Hoboken and West Village

(d) Rome: Monte Sacro

Figure 8: Examples of regions identified as similar by GeoExplore.

checkins among Foursquare top categories – i.e., predominantly at *Food*, but also *Shops* and other *Professional* venues, as one can see on 8(b).

6 Comparison with previous approaches

In this section, we compare empirically our approach with previous works. Ideally, our comparison would be with works that address the same task as this paper, i.e., model the distribution of venues across a city. In such a case, we would have a natural and direct measure of comparison, namely the predictive performance of each model in terms of log-likelihood. However, to the best of our knowledge, no such work is readily available. Therefore, our comparison is with previous works that (*i*) address slightly different tasks and (*ii*) do not make explicit use of a probabilistic model. Nevertheless, the comparison serves as a 'sanity check' for our approach and helps demonstrate better the proposed technique.

We compare with two methods that provide publicly available results, namely Livehoods[5] and Hoodsquare[6]. Their results cover three large US cities that also appear in our dataset. Both methods use Foursquare data and output a geographical clustering of venues within a city, with each such cluster defining one region on the map of the city. In the case of Livehoods [4], the clusters are obtained through spectral clustering on a nearest-neighbor graph over venues, where the edge weights quantify the volume of visitors who check-in at both adjacent venues. In the case of Hoodsquare [5], clusters are obtained by employing the OPTICS clustering algorithm on venues, using a number of venue features (location, category, peak time of activity during the day, and a binary touristic indicator). Furthermore, we also implement the method of [6]. Although it uses Twitter data to classify land usage, it is also a simple non-parametric method based on Self Organized Maps (referred to as *SOM* hereafter) to segment the city by clustering geolocated tweets.

To compare, we perform the following procedure. First, we obtain the clusters returned by each method. We interpret each of those clusters as a region that belongs to one *topic*, in the sense that we have been using the term in the context of our model. To map them to our setting, we approximate the shape of each region with the smallest bivariate Gaussian so that the entire region is enclosed within two standard deviations of the Gaussian in each direction (see Figure 9 for the visual results of this approximation in San Francisco). In this way, we obtain a number k of

[5] `http://livehoods.org/maps`
[6] `http://pizza.cl.cam.ac.uk/hoodsquare`

Gaussians from each method. We then train an instance of our model on our data, using the same number k of topics, and keeping the Gaussians associated with each topic fixed to the Gaussians extracted with the aforementioned steps. As in previous sections of the paper, we hold out 20% of the data as test set, on which we evaluate the log-likelihood of each learned model instance.

The log-likelihood achieved by the different models is shown in the first row of Table 4. As one can see there, the results based on our model perform better in predicting the test set. This is not surprising, since our approach optimizes predictive accuracy directly. Nevertheless, the results provide evidence that our approach works reasonably well for the task it was designed to address.

To further quantify the differences between the four approaches, we report additional quantities from the learned models, described below. Essentially, those quantities capture how distinct the identified regions of each model are in terms of the associated features.

Mean Feature Entropy We consider each categorical feature separately and, for each topic region in the respective model of the four approaches, we measure the entropy of the respective multinomial distribution β. Intuitively, we would like the regions that constitute our model instances to capture the variance of the various features across the city. Therefore, we would like the β distributions of the model instances to have lower entropy (i.e., be farther from uniform). Table 4 reports the mean entropy of β across regions for each categorical feature and for each of the methods in the three US cities. The relevant lines in the table are the ones labelled 'mean [feature] entropy'. As one can confirm, in the majority of cases the model instance based on our method returns β distributions of lower entropy.

Jensen–Shannon Divergence from City Average Another way to quantify the distinctiveness of the various regions is to measure the distance of the β distribution of each feature and topic in the model from the average distribution μ for the same feature across the city. One principled approach to quantify this difference is to use Jensen–Shannon divergence $\mathrm{JSD}(\beta, \mu)$, a symmetrized version of the Kullback–Leibler divergence $\mathrm{KL}(P \parallel Q)$ defined as:

$$\mathrm{JSD}(\beta, \mu) = 0.5 \cdot \mathrm{KL}(\beta \parallel (\beta + \mu)/2) + 0.5 \cdot \mathrm{KL}(\mu \parallel (\beta + \mu)/2))$$

Intuitively, it is desirable for β distributions of different topics to differ from average city behavior as captured by μ distributions. Table 4 reports the average Jensen–Shannon divergence across the topics for each categorical feature, city, and method. The relevant lines in the table are the ones labelled '[feature] JSD from city'. Again, in the majority of cases, the model instance based on our method returns β distributions that differ more from city average distribution than the methods we compare with.

To summarize our findings from Table 4, our model has better predictive performance, while generally identifying topic regions that are more distinct with each other and further from average, despite their high overlap. The results thus provide evidence that our approach discovers regions with desirable properties.

7 Related Work

Urban Computing is an active area of research, partially due to the increasing volume of digital data related to human activity and the potential to use such data to improve life in cities. Below, we discuss related works that either address a similar task (finding geographical structure in city activity) or use a similar approach to ours to address different tasks. To the best of our knowledge, we are the first to employ a fully probabilistic approach to this task – and thus discuss further the concept of sparse modeling. Finally, we discuss other Urban Computing tasks more loosely related to our work.

Table 4: Comparison in San Francisco, New York and Seattle between our model (with @k topics), SOM, Livehoods (LH) and Hoodsquare (HS, which has no neighborhoods available in Seattle). The last abbreviation, JSD, stands for the average Jensen–Shannon divergence between regions and city-wide distributions of the four features we consider: category, users, dayOfWeek and timeOfDay. Each group of two adjacent columns is a comparison between a competing method and our model. The arrow after the name of each measure indicates whether higher or lower values are better.

		San Francisco						New York						Seattle			
		SOM	Us@22	HS	Us@13	LH	Us@39	SOM	Us@18	HS	Us@12	LH	Us@68	SOM	Us@43	LH	Us@64
likelihood per venue	↗	-198.2	-197.4	-202.9	-199.2	-197.0	-195.7	-270.4	-268.4	-335.3	-271.2	-264.2	-262.4	-175.6	-174.5	-174.7	-172.9
mean category entropy	↘	4.786	4.740	4.959	4.754	4.808	4.808	4.847	4.737	4.905	4.784	4.864	4.815	5.178	5.196	5.141	5.216
category JSD from city	↗	0.070	0.079	0.053	0.066	0.066	0.059	0.061	0.079	0.051	0.070	0.056	0.056	0.004	0.004	0.009	0.001
mean dayOfWeek entropy	↘	1.896	1.900	1.908	1.912	1.892	1.900	1.928	1.924	1.924	1.932	1.916	1.916	1.910	1.905	1.899	1.865
dayOfWeek JSD from city	↗	0.011	0.011	0.007	0.008	0.011	0.011	0.004	0.005	0.004	0.004	0.007	0.007	0.007	0.007	0.009	0.016
mean timeOfDay entropy	↘	1.437	1.424	1.488	1.483	1.428	1.416	1.589	1.550	1.536	1.561	1.540	1.540	1.476	1.442	1.475	1.377
timeOfDay JSD from city	↗	0.030	0.033	0.018	0.024	0.031	0.035	0.014	0.024	0.014	0.023	0.022	0.024	0.025	0.033	0.025	0.048
mean user entropy	↘	5.550	5.544	5.610	5.328	5.521	5.407	5.338	5.386	5.931	5.370	5.221	4.947	5.069	5.178	5.128	5.088
user JSD from city	↗	0.176	0.176	0.160	0.204	0.175	0.186	0.212	0.206	0.101	0.203	0.209	0.234	0.198	0.178	0.183	0.176

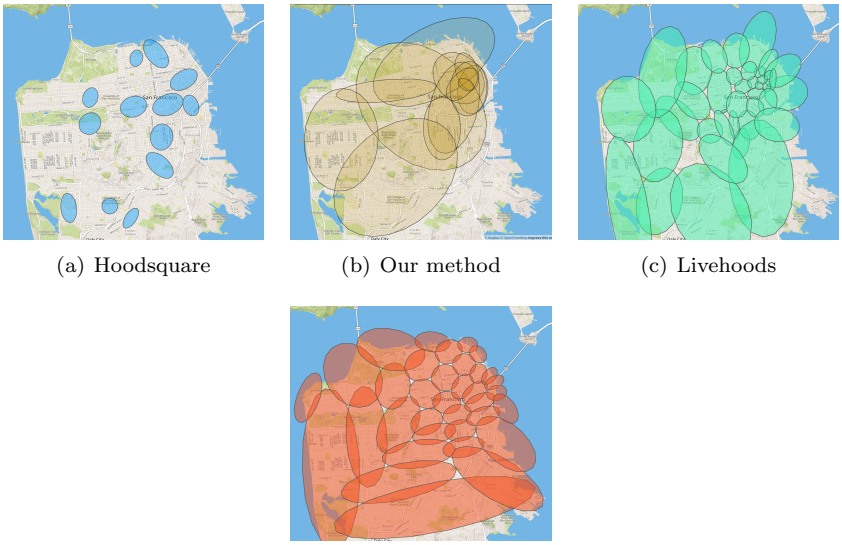
7.1 Finding Structure in Urban Activity

Finding cohesive geographical regions within cities has been attempted using a variety of data sources: public transport and taxi trajectories [10, 11], cellphone activity [12], geotagged tweets [6], social interactions [13] or types of buildings [14].

In that context, Location Based Social Networks (*LBSNs*) have also proven a rich source of data and were used by recent works. For instance, [4] collects checkins and build a m-nearest spatial neighbors graph of venues, with edges weighted by the cosine similarity of both venues' user distribution. The regions are the spectral clusters of this graph. Using similar data, [5] describes venues by category, peak time activity and a binary touristic indicator. Venues are clustered in hotspots along all these dimensions by the OPTICS algorithm. The city is divided into a grid, with cells described by their hotspot density for each feature. Finally, similar cells are iteratively clustered into regions. Like us, [15] considers venues to be essential in defining regions. The city is divided into a grid of cells with the goal of assigning each cell a category label in a way that is as specific as possible while being locally homogeneous. This is done through a bottom-up clustering which greedily merge neighboring cells to improve a cost function formalizing this trade off.

Whereas these results are evaluated with user interviews and build upon well known algorithmic techniques, they rely on ad-hoc modeling decisions (such as graph construction and grid granularity) that do not derive directly from the data, thus questioning the statistical significance of the obtained results. Furthermore, because the clustering is not guided simultaneously by all the available data features, such as time and aspects other than venue category, important information might be going amiss in those approaches.

On the other hand, there are works that take a probabilistic approach, although their aim is different than ours. For instance, [16] assigns venues to a grid and runs Latent Dirichlet Allocation (LDA) on their categories. However it does not output explicit regions, and the grid is a coarse approximation for using spatial information. Instead, [17] fixes a number K of localized topics to be discovered, as well as a set of N Gaussian spatial regions. Each region has a topic distribution and each topic is a multinomial distribution over all possible Flickr photos tags. Relaxing several assumptions, notably the Gaussian shape, [18] extends Hierarchical Dirichlet Process to spatial



(a) Hoodsquare (b) Our method (c) Livehoods

(d) SOM

Figure 9: In San Francisco, the second panel shows the top 12 regions (sorted by decreasing weight) we obtained with our probabilistic model. On the left, Hoodsquare regions are extracted from their website and transformed into Gaussians. On the right are the same picture for Livehoods and SOM. All methods show that the city activity occurs mainly downtown but it also highlights differences between approaches. For instance, although Livehoods exhibits some overlap, it is only due to the Gaussian conversion whereas we do not restrain venues to serve a single function in a single region by construction, but only let that happens if the data support it.

data, giving rise to an almost fully non-parametric model (as the number of regions still needs to be set manually). While such methods bear similarity with ours, the domain of application presented by their authors forbids direct comparisons.

Closer to the task of finding regions, [19] performs LDA on checkins in New York. The five resulting topics are called urban activities, and venues are clustered by their topic proportion across time. Contrary to us, this clustering does not produce clearly defined regions, since it is done as a post processing step. Indeed, their LDA model does not incorporate a spatial dimension. Moving from checkins to a dataset of 8 millions Flickr geotagged photos, [20] probabilistically assigns tags to one of the three levels of a spatial hierarchy, where each node is associated with a multinomial distribution over tags. Regions can then be characterized by finding their most descriptive tags.

7.2 Sparse Topic Model

We now present related applications of the Sparse additive generative model on spatial data. The original SAGE paper [3] evaluates its model effectiveness on the task of predicting the localization of twitter users by learning not only topics about words but also about metadata (i.e., in which region was the tweet written) and shows good accuracy. Later, a simplified version of it was used to find regions which exhibit geolocated idioms [21]. It is possible to better model user location by building a hierarchy of regions [22]. Even though the sparsity of model is well suited to the sparsity of textual data, note that methods which do not use topic modelling give competitive results in terms of accuracy [23, 24].

Another task hindered by data sparsity and which benefits from modeling user preferences is spatial item recommendation. The interested reader will find many examples exploiting LBSN in a recent survey [25], but here we give a taste of two approaches inspired by SAGE. In both cases, topics are distributions over words and venues. Each user is endowed with her own topic, and so to are each region. [26] uses SAGE to model user topics as a variation from the overall global distribution. To improve out of town recommendation, [27] assigns to regions both local and tourist topics. Learning such high number of parameters is made possible by combining SAGE and a hierarchical model called spatial pyramid.

7.3 Other Urban Computing Problems

After finding regions in a city, a natural task is to compare them, within or across cities. One might also look at different granularity to perform such urban comparisons, whether points of interest or cities as a whole. Finally, one might focus on users and model their preferences through mobility data.

Comparing regions We saw that one way to compare regions is to assign them descriptive tags [20]. Others have looked at a more information retrieval approach [28], while in [7], authors collect data from Foursquare and Flickr to associate venues with a features vector summarizing their time activity, their surroundings, their category and their popularity. They then define the similarity between two regions as the Earth Mover Distance between the two sets of venues feature vectors they contain. Finally they devise a pruning and clustering strategy to find similar regions between cities.

Finding points of interest Regions are only one subdivision of the city, another one are points of interest, which are locations where the user activity show some specificities. Geotagged photos can be mined to extract the semantics of such locations [29, 30, 31, 32]. As a representative example, [33] compares various spatial methods to discover small areas in San Francisco where one photo tag appears in burst. GPS trajectories provide useful information as well [34, 35, 36]. For instance, [37] extracts stay points from car GPS data and assess their significance by how many visitors go there, how far they traveled to reach them and how long they stay.

Comparing cities This has been done by comparing the spatial distribution of human hotspots using call data [38], the call data profile themselves [39], the distribution of venues category at various scales [40], or by building a network of city from urban residents mobility flow and computing centrality measures [41].

Clustering users With venues, users are the other side of the coin of what defines a region in a city, and some works have mine their activities to extract meaningful groups. For instance, [42] clusters users by the similarity of their venues category transition probability matrix. Another approach is to consider users as document, checkins as word and apply LDA, thus revealing cohesive communities [43] and describing people lifestyle [44].

One can find other applications of Urban Computing in related literature surveys [1, 45].

8 Conclusions

In this work, we made use of a probabilistic model to reveal how venues are distributed in cities in terms of several features. As most habitants of a city do not visit most of the available venues, we cope with the induced sparsity by adapting the sparse modeling approach of [3] to data at hand. Fitting our model to a large dataset of more than 11 million checkins in 40 cities around the world, we show the insights provided by such an unsupervised approach.

First, using the extracted model instances, we calculated the probability distribution of a single feature conditioned on the location in the city. This enabled us to construct a heatmap of that

feature to highlight what feature values are most likely and distinctive at different locations within a city.

Secondly, we described a principled approach to quantify the importance of different features within the trained models. Whereas all features contribute, we discovered that the most defining feature for the components uncovered by the model is the visitors of venues. This finding suggests that further analysis of user behavior is a promising direction for extracting additional insights.

Third, after focusing on the various regions of a single city, we used the extracted model instances to find the most two similar regions between two cities, a task which was previously attempted with a more heuristic approach [7]. This time we also benefit of the solid theoretical grounds of probabilistic models to define a principled measure of similarity and we describe a procedure to greedily find two regions maximizing measure. We illustrate this matching process with anecdotal evidence in several cities.

Finally, we compare our approach with previous approaches that provide similar output and show that our regions are both more consistent with the data (in terms of predictive performance) and have more sharply defined characteristics, meaning they are easier to distinguish from one another.

A review of recent related works in the Urban Computing field suggests that whereas the area is active and that understanding urban activities is a worthy endeavour which benefits from geotagged data, it can be pushed further by the use of probabilistic models, as such models come with great interpretative power.

Looking beyond this paper, there are further directions in which we can improve our discovery process, providing additional interpretations along the way:

- The first direction is to use a complementary evaluation process, one that would involve more closely users, since we show they are the main actor of the regions we discover. For instance, this could take the form of interviews. The purpose of this would be to identify and correct, if any, significant biases that are embodied in particular datasets (`Foursquare` activity, in our case). One major difficulty, however, lies in finding local experts who have good knowledge of the structure and activity of each city.
- The model itself could also be extended, for instance by incorporating hierarchies of regions. This would both provide more structure to our results and allow us to apply our method to larger geographical area while keeping sparsity and runtime under control. Hierarchy is a well studied concept in both spatial data mining and topic modelling [46] and thus we are confident this would be a feasible improvement. Another direction would be to incorporate additional features into the model (e.g., continuous features)
- Just as natural landscapes change with time, whether because of the day/night cycle or the passing of seasons, so do cities. It is not far fetched to imagine than coastal areas of Barcelona or San Francisco witness different patterns of activities in the winter than in the summer. Again, following the time evolution of topics has been addressed in different settings [47].

Acknowledgements

This work is supported by the European Community's H2020 Program under the scheme 'INFRAIA-1-2014-2015: Research Infrastructures', grant agreement #654024 'SoBigData: Social Mining & Big Data Ecosystem'.

References

[1] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban Computing: Concepts, Methodologies, and Applications," *ACM Transaction on Intelligent Systems and Technology*, vol. 5, no. 3, pp. 38:1–38:55, 2014.

[2] "World Urbanization Prospects, the 2014 Revision: Highlights," United Nations, Department of Economic and Social Affairs, Population Division, New-York, Tech. Rep., 2014.

[3] J. Eisenstein, A. Ahmed, and E. P. Xing, "Sparse Additive Generative Models of Text," in *ICML*, Seattle, WA, 2011, pp. 1041–1048.

[4] J. Cranshaw, J. I. Hong, and N. Sadeh, "The livehoods project: Utilizing social media to understand the dynamics of a city," in *ICWSM*, 2012, pp. 58–65.

[5] A. X. Zhang, A. Noulas, S. Scellato, and C. Mascolo, "Hoodsquare: Modeling and recommending neighborhoods in location-based social networks," in *ASE/IEEE SocialCom*, 2013, pp. 69–74.

[6] V. Frias-Martinez and E. Frias-Martinez, "Spectral clustering for sensing urban land use using Twitter activity," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 237–245, 2014.

[7] G. Le Falher, Gionis Aristides, and M. Mathioudakis, "Where Is the Soho of Rome? Measures and Algorithms for Finding Similar Neighborhoods in Cities," in *ICWSM*, Oxford, 2015.

[8] J. O. Berger and D. Sun, "Objective priors for the bivariate normal model," *The Annals of Statistics*, pp. 963–982, 2008.

[9] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman, "Indexing by latent semantic analysis," *JAsIs*, vol. 41, no. 6, pp. 391–407, 1990.

[10] N. J. Yuan, Y. Zheng, X. Xie, Y. Wang, K. Zheng, and H. Xiong, "Discovering Urban Functional Zones Using Latent Activity Trajectories," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 27, no. 3, pp. 712–725, 2015.

[11] Jichang Zhao, Ruiwen Li, X. Liang, and K. Xu, "Segmentation and evolution of urban areas in beijing: A view from mobility data of massive individuals," in *Proceedings of the 12th International Conference on Service Systems and Service Management (ICSSSM)*, 2015.

[12] J. L. Toole, M. Ulm, M. C. González, and D. Bauer, "Inferring Land Use from Mobile Phone Activity," in *UrbComp*, New York, NY, USA, 2012, pp. 1–8.

[13] J. R. Hipp, R. W. Faris, and A. Boessen, "Measuring 'neighborhood': Constructing network neighborhoods," *Social Networks*, vol. 34, no. 1, pp. 128–140, 2012.

[14] Z. Cao, S. Wang, G. Forestier, A. Puissant, and C. F. Eick, "Analyzing the Composition of Cities Using Spatial Clustering," in *UrbComp*, New York, NY, USA, 2013, pp. 14:1–14:8.

[15] C. Vaca, D. Quercia, F. Bonchi, and P. Fraternali, "Taxonomy-Based Discovery and Annotation of Functional Areas in the City," in *ICWSM*, 2015, pp. 445–453.

[16] J. Cranshaw and T. Yano, "Seeing a home away from the home: Distilling proto-neighborhoods from incidental data with Latent Topic Modeling," in *NIPS'10 Workshop of Computational Social Science and the Wisdom of the Crowds*, 2010.

[17] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang, "Geographical Topic Discovery and Comparison," in *WWW*, 2011, pp. 247–256.

[18] C. C. Kling, J. Kunegis, S. Sizov, and S. Staab, "Detecting non-gaussian geographical topics in tagged photo collections," in *WSDM*, 2014, pp. 603–612.

[19] F. Kling and A. Pozdnoukhov, "When a city tells a story: urban topic analysis," in *SIGSPATIAL*, 2012, pp. 482–485.

[20] M. Kafsi, H. Cramer, B. Thomee, and D. A. Shamma, "Describing and Understanding Neighborhood Characteristics through Online Social Media," in *WWW*, Florence, 2015, pp. 549–559.

[21] J. Eisenstein, "Written dialect variation in online social media," in *Handbook of Dialectology*, 2015.

[22] A. Ahmed, L. Hong, and A. J. Smola, "Hierarchical Geographical Modeling of User Locations from Social Media Posts," in *WWW*, Republic and Canton of Geneva, Switzerland, 2013, pp. 25–36.

[23] R. Priedhorsky, A. Culotta, and S. Y. Del Valle, "Inferring the Origin Locations of Tweets with Quantitative Confidence," in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing*, New York, NY, USA, 2014, pp. 1523–1536.

[24] D. Flatow, M. Naaman, K. E. Xie, Y. Volkovich, and Y. Kanza, "On the Accuracy of Hyper-local Geotagging of Social Media Content," in *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2015, pp. 127–136.

[25] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel, "Recommendations in location-based social networks: a survey," *GeoInformatica*, vol. 19, no. 3, pp. 525–565, 2015.

[26] B. Hu and M. Ester, "Spatial Topic Modeling in Online Social Media for Location Recommendation," in *Proceedings of the 7th ACM Conference on Recommender Systems*, New York, NY, USA, 2013, pp. 25–32.

[27] W. Wang, H. Yin, L. Chen, Y. Sun, S. Sadiq, and X. Zhou, "Geo-SAGE: A Geographical Sparse Additive Generative Model for Spatial Item Recommendation," in *KDD*, New York, New York, USA, 2015, pp. 1255–1264.

[28] C. Sheng, Y. Zheng, W. Hsu, M. Lee, and X. Xie, "Answering Top-k Similar Region Queries," in *Database Systems for Advanced Applications*, 2010, vol. 5981, pp. 186–201.

[29] D.-P. Deng, T.-R. Chuang, and R. Lemmens, "Conceptualization of place via spatial clustering and co-occurrence analysis," in *Proceedings of the 2009 International Workshop on Location Based Social Networks*, New York, New York, USA, 2009, pp. 49–56.

[30] M. Shirai, M. Hirota, S. Yokoyama, N. Fukuta, and H. Ishikawa, "Discovering Multiple HotSpots Using Geo-tagged Photographs," in *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, 2012, pp. 490–493.

[31] R. Feick and C. Robertson, "A multi-scale approach to exploring urban places in geotagged photographs," *Computers, Environment and Urban Systems*, 2014.

[32] Y. Hu, S. Gao, K. Janowicz, B. Yu, W. Li, and S. Prasad, "Extracting and understanding urban areas of interest using geotagged photos," *Computers, Environment and Urban Systems*, vol. 54, pp. 240–254, 2015.

[33] T. Rattenbury and M. Naaman, "Methods for extracting place semantics from Flickr tags," *ACM Transactions on the Web*, vol. 3, no. 1, pp. 1–30, 2009.

[34] M. R. Uddin, C. Ravishankar, and V. J. Tsotras, "Finding Regions of Interest from Trajectory Data," in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, 2011, pp. 39–48.

[35] C. Zhang, J. Han, L. Shou, J. Lu, and T. L. Porta, "Splitter: Mining Fine-Grained Sequential Patterns in Semantic Trajectories," *Proceedings of the VLDB Endowment*, vol. 7, no. 9, pp. 769–780, 2014.

[36] C. Parent, S. Spaccapietra, C. Renso, G. Andrienko, N. Andrienko, V. Bogorny, M. L. Damiani, A. Gkoulalas-Divanis, J. Macedo, N. Pelekis, Y. Theodoridis, and Z. Yan, "Semantic Trajectories Modeling and Analysis," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 42:1–42:32, 2013.

[37] X. Cao, G. Cong, and C. S. Jensen, "Mining significant semantic locations from GPS data," *Proceedings of the VLDB Endowment*, vol. 3, no. 1-2, pp. 1009–1020, 2010.

[38] T. Louail, M. Lenormand, O. G. Cantu Ros, M. Picornell, R. Herranz, E. Frias-Martinez, J. J. Ramasco, and M. Barthelemy, "From mobile phone data to the spatial structure of cities," *Scientific Reports*, vol. 4, 2014.

[39] S. Grauwin, S. Sobolevsky, S. Moritz, I. Gódor, and C. Ratti, "Towards a comparative science of cities: Using mobile traffic records in new york, london, and hong kong," in *Computational Approaches for Urban Environments*, 2015, vol. 13, pp. 363–387.

[40] D. Preoţctiuc-Pietro, J. Cranshaw, and T. Yano, "Exploring Venue-based City-to-city Similarity Measures," in *UrbComp*, New York, NY, USA, 2013, pp. 16:1–16:4.

[41] M. Lenormand, B. Gonçalves, A. Tugores, and J. J. Ramasco, "Human diffusion and city influence," *Journal of The Royal Society Interface*, vol. 12, no. 109, 2015.

[42] D. Preoţiuc-Pietro and T. Cohn, "Mining user behaviours: A study of check-in patterns in location based social networks," in *Proceedings of the 5th Annual ACM Web Science Conference*, New York, NY, USA, 2013, pp. 306–315.

[43] 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*, New York, NY, USA, 2012, pp. 919–926.

[44] N. J. Yuan, F. Zhang, D. Lian, K. Zheng, S. Yu, and X. Xie, "We Know How You Live: Exploring the Spectrum of Urban Lifestyles," in *Proceedings of the First ACM Conference on Online Social Networks*, New York, NY, USA, 2013, pp. 3–14.

[45] D. Tasse and J. Hong, "Using Social Media Data to Understand Cities," in *Proceedings of NSF Workshop on Big Data and Urban Informatics*, 2014.

[46] D. M. Blei, T. L. Griffiths, and M. I. Jordan, "The nested chinese restaurant process and bayesian nonparametric inference of topic hierarchies," *J. ACM*, vol. 57, no. 2, pp. 7:1–7:30, 2010.

[47] D. M. Blei and J. D. Lafferty, "Dynamic topic models," in *ICML*, 2006, pp. 113–120.