

Identifying Meaningful Places: The Non-parametric Way

Petteri Nurmi and Sourav Bhattacharya

Helsinki Institute for Information Technology HIIT
Department of Computer Science, P.O. Box 68,
FI-00014 University of Helsinki, Finland
`petteri.nurmi@cs.helsinki.fi`,
`sourav.bhattacharya@cs.helsinki.fi`

Abstract. Gathering and analyzing location data is an important part of many ubiquitous computing applications. The most common way to represent location information is to use numerical coordinates, e.g., latitudes and longitudes. A problem with this approach is that numerical coordinates are usually meaningless to a user and they contrast with the way humans refer to locations in daily communication. Instead of using coordinates, humans tend to use descriptive statements about their location; for example, "I'm home" or "I'm at Starbucks." Locations, to which a user can attach meaningful and descriptive semantics, are often called places. In this paper we focus on the automatic extraction of places from discontinuous GPS measurements. We describe and evaluate a non-parametric Bayesian approach for identifying places from this kind of data. The main novelty of our approach is that the algorithm is fully automated and does not require any parameter tuning. Another novel aspect of our algorithm is that it can accurately identify places without temporal information. We evaluate our approach using data that has been gathered from different users and different geographic areas. The traces that we use exhibit different characteristics and contain data from daily life as well as from traveling abroad. We also compare our algorithm against the popular k-means algorithm. The results indicate that our method can accurately identify meaningful places from a variety of location traces and that the algorithm is robust against noise.

1 Introduction

The location of a user plays an important role in many ubiquitous computing applications. The most common way to represent location information is to use numerical coordinates such as latitudes and longitudes. The main problems with this approach are that the raw location measurements are difficult to use in location-aware applications [1] and that the measurements are seldom meaningful to a user [2]. For example, we do not refer to our home or workplace as a pair of GPS coordinates. A more appropriate way to utilize location information is to use the notion of *place*. A widely used definition for a place is given by Relph, who defines it as a combination of the physical setting, the activities

supported by the place, and the meanings attributed to the place [3]. In this paper, we consider a place as a location to which a user can attach meaningful and descriptive semantics. Thus we focus only on the physical setting and the meanings attributed to a place; see [4] for other definitions of a place.

In this paper we focus on the task of automatically extracting places from discontinuous GPS traces. We consider two kinds of discontinuous traces: traces that have been gathered by sampling the GPS periodically (once every minute) and traces that have been gathered by sampling the GPS whenever the GSM cell identifier changes. Our methodology can be applied also with other kinds of discontinuous GPS traces (e.g, periodically sampled localization traces). As our main contribution we introduce a statistical model for extracting places from this kind of data. The model is based on the non-parametric Bayesian framework¹, more precisely, Dirichlet process mixture models [5,6]. We model data points using multivariate Normal distributions and thus our model can also be understood as an infinite Gaussian mixture model [7]. The main novelty of our approach is that the algorithm is fully automated and does not require any parameter tuning. Another novel aspect of our algorithm is that it can accurately identify places without considering temporal information.

We evaluate our place extraction algorithm using two data sets. The data sets that we use exhibit rather different characteristics: the first data set contains location traces from daily life situations and the second data set contains location traces gathered during a business trip. We also compare the Dirichlet process clustering algorithm against the popular k-means algorithm. Our results indicate that the Dirichlet process model is a good candidate for extracting places as, in both cases, the algorithm produces accurate and compact results. In addition, the Dirichlet process model is robust against noise in the location measurements.

The rest of the paper is organized as follows: Sec. 2 gives background information on why certain locations are meaningful. Sec. 3 introduces related work on place extraction. Sec. 4 introduces our problem setting and presents the statistical model. Sec. 5 presents our experiments. Sec. 6 concludes the paper.

2 Background: Why Some Places Are Meaningful?

There are various explanations to why some places would be meaningful in the first place. For example, environmental psychology [8] examines how people structure their daily activities around common places such as GROCERY STORE, HOME and WORK. Accordingly, this view suggests that meaningful locations correspond to locations around which users relate specific activities. This view has been applied in pervasive computing by Zhou et al. [9].

A complementary view can be given using social identity theory [10]. Social identity theory studies the relationships between the individual and the society, and, more specifically, how an individual's self conceptions relate to the

¹ The term non-parametric Bayesian is somewhat confusing as the models actually contain an infinite number of parameters. In this paper we use the term 'non-parametric' to follow its common usage found in the literature; see, e.g., [5].

expectations and norms imposed by the society. Nurmi and Koolwaaij [11] have used social identity theory to argue that some places are meaningful because they act as boundaries between different roles and social categories. For example, **WORK** is related to being an employee whereas **HOME** is strongly related to social categories specific to private life. This view is complementary to the environmental psychology view in the sense that it attempts to explain why the structuring takes place.

The places that act as boundaries between different social categories are not the only meaningful places. Consider for example the sentence “let’s meet at the same place where we met yesterday”. This sentence refers to a location that is meaningful in a specific social context. The data in our setting does not carry sufficient information about the social situation of a user, and for this reason we ignore this kind of places in this paper.

Also various public places can be meaningful, simply because they serve as easily recognizable navigation cues. Zhou et al. [12] conducted a user study that identified five different place categories: generic, well-known public, specific public, personal and activity-based. Of these categories, we focus on the generic (airport, gas store), personal (home, workplace) and activity-based (hobby related) places, as they separate different social categories and activities.

3 Related Work

The main research directions in the analysis of location data are *localization* and *place extraction*. In localization, the goal is to determine the user’s location as accurately as possible using whatever location information is available. The most common technique for localization is fingerprinting; see, e.g., [13,14]. The second category, *place extraction*, attempts to find spatial areas that are somehow important to the user. Localization complements place extraction in the sense that it can be used to gather accurate location traces that place detection algorithms can use. In this paper we focus exclusively on place extraction.

Approaches for extracting places from location data can be categorized based on the source of location information. The most common approach has been to use continuously gathered GPS traces. The algorithms for GPS traces are typically based on distance and time-related heuristics. For example, Marmasse et al. [15] use signal loss and distance between successive measurements to identify buildings. Meaningful places are then obtained based on the frequency of visits to the specific buildings. Ashbrook and Starner [2] use a cut-off parameter to determine whether a user stays long enough within an area that has a predefined radius. If the duration of the stay exceeds the value of the cut-off parameter, the location is identified as a place. Toyama et al. [16] present a variation of this work that employs multiple radius parameters to detect meaningful locations at different granularities. Zhou et al. [17] use a modified DBScan algorithm and temporal preprocessing to extract places. The temporal preprocessing ensures that the places are really visited frequently enough and the modification to the DBScan algorithm is needed to cope with signal errors. Other approaches for GPS data are presented, e.g., in [18,19,20,21].

The main problem with the GPS-based approaches is that GPS signal is not available indoors. In addition, tall buildings can reflect signals and cause signal loss or weak measurements in metropolitan areas. Nevertheless, GPS information is easily available and it requires minimal infrastructure investments. The main problem with place detection algorithms that use GPS traces is that they usually have tunable parameters or cut-off values. The algorithms can easily become sensitive to fluctuations in GPS signals and may require parameter tuning for different GPS receivers and environments; see also [22].

In bounded areas, such as office buildings, campuses, research laboratories or even individual cities, information about the physical location of radio beacons can be used to derive estimated location traces (e.g., [13,14]), from which places can be extracted. For example, Kang et al. [23] first use a customized spatial clustering algorithm to detect clusters from the data. The clusters act as candidate places and they are labeled as a place when the user stays long enough within the cluster. The BeaconPrint algorithm of Hightower et al. uses similarities in fingerprints of the radio environment to detect places [24]. An advantage of these techniques is that they work both indoors and outdoors. However, a disadvantage of these approaches is that detailed fingerprint information is not available on many mobile phones² and custom hardware is often needed to obtain the required information. Nevertheless, our algorithm can be also used for location traces derived via fingerprinting techniques.

Aipperspach et al. [22] have used a commercial positioning system to obtain high precision indoor location traces. These traces were then used to extract places within a home. The algorithm that they use is based on Gaussian mixture models, which are a simplified version of the model we use. The main problem with this approach is that obtaining high precision location traces requires costly infrastructure investments, which limits the usefulness of this approach.

Also some work on extracting places from GSM identifiers has been suggested [25,26]. Since the size of GSM cells can be rather large, places extracted from GSM data are necessarily only crude estimates of the true meaningful places. Nevertheless, the major advantage of GSM cell based clustering is that it does not require any additional hardware and that the clustering can be performed on the device without need to ever connect to a server. This option is thus optimal from a privacy perspective.

It is also possible to combine the advantages of the GPS approach with GSM identifier based clustering. Nurmi and Koolwaaij [11] use data consisting of GSM transitions and GPS coordinates at the transition point. This approach is more accurate than the GSM identifier based approach, and requires fewer resources from the device as GPS data is read only when a cell transition occurs. The main disadvantage of this approach is that the system is not able to get accurate location information when the user is indoors. Furthermore, since data is collected only at the transition points, this can cause bias to the results.

² For example, Nokia phones provide only information about the GSM cell tower to which the phone is currently connected. Accurate fingerprinting, on the other hand, requires information about several GSM towers.

The work presented in this paper offers three advantages over existing work. First of all, our approach is fully automated in the sense that it does not depend on any tunable parameters; most earlier methods require at least specifying the number of clusters beforehand whereas our algorithm is able to infer also this from data. Secondly, our approach requires minimal hardware investments³. Finally, the discontinuous nature of the GPS traces makes it possible to gather data for longer periods before the user needs to recharge the device. In practice we have been able to gather data for more than one day without recharging the mobile device.

4 Setting, Statistical Model and Algorithm

This section describes the statistical model and the algorithm that is used to cluster data points. The notation that is used in the paper is summarized in Table 1.

Table 1. A summary of the notation used in the paper

Symbol	Description
y_i	Individual data point
\mathbf{y}	The vector (y_1, \dots, y_n) of data points
c_i	Cluster indicator for data point i
\mathbf{c}	Vector (c_1, \dots, c_k) of cluster indicators
\mathbf{c}_{-i}	The vector $(c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n)$
k	The number of clusters
n	Number of data points
\bar{y}	Sample mean
\bar{y}_j	Mean of data points associated with cluster j
Σ	Sample precision
μ_j	The mean vector of cluster j
S_j	The precision matrix of cluster j
n_j	Number of data points in cluster j
$n_{-i,j}$	Number of data points in cluster j excluding point i
λ	Mean vector for the prior on cluster means μ_j
R	Precision matrix for the prior on cluster means μ_j
β	Degrees of freedom for the prior on cluster precision matrices S_j
W	Inverse scaling matrix for the prior on cluster precision matrices S_j
α	Concentration parameter of the Dirichlet process prior
ϕ	Auxiliary variable that is used for sampling α
π_ϕ	Mixture weight for the distribution used to sample α

Model

The data in our setting consists of (`latitude`, `longitude`) pairs that mark the transition point between two GSM cells. We use $\mathbf{y} = (y_1, \dots, y_n)$ to denote the

³ Mobile phone with an integrated GPS or a phone and a Bluetooth GPS device.

data. Each data point y_i is assumed to belong to a single cluster. Intuitively, this assumption implies that the user cannot be simultaneously at HOME and at WORK. We assume that the number of clusters is finite but unknown beforehand. The variable k is used to denote the number of clusters, and c_i is used to denote the cluster indicator that specifies to which cluster data point y_i is currently assigned. We use $\mathbf{c} = (c_1, \dots, c_n)$ to denote the vector of cluster indicators over all data points.

The data in each cluster is assumed to follow a multivariate Normal distribution with mean μ_j and precision⁴ matrix S_j . We assume that both μ_j and S_j are unknown. The distribution of a single data point y_i is thus given by

$$y_i | c_i = j, \mu_j, S_j \sim \mathcal{N}(\mu_j, S_j^{-1}). \quad (1)$$

Since the cluster parameters μ_j and S_j are unknown, we need to assign priors for them. The selection of the priors is important as they influence how likely it is that the clustering algorithm creates a new cluster component. In our case we use conjugate priors because they offer a good balance between computational simplicity and clustering performance. The conjugate prior for the multivariate Normal distribution, when both the mean and the precision matrix are unknown, is to assign a Normal distribution on the mean vector and a Wishart distribution on the precision matrix (see, e.g., [27]). Accordingly, we have

$$\mu_j \sim \mathcal{N}(\lambda, R^{-1}) \quad (2)$$

$$S_j \sim \text{Wi}\left(2\beta, \frac{1}{2}W^{-1}\right), \quad (3)$$

where λ, β, R and W are hyperparameters. Following Rasmussen [7], we use a hierarchical model and assign priors to all hyperparameters. If we want clusters that are on average of specific size, we can fix the values of β and W beforehand. This can be done, for example, by assigning W to be the sample covariance and setting β so that the product βW^{-1} corresponds to the desired coverage⁵.

Let \bar{y} denote the sample mean and Σ the sample precision. We assign λ a Normal distribution whose mean equals the sample mean and whose covariance matrix corresponds to the sample covariance, i.e.,

$$\lambda \sim \mathcal{N}(\bar{y}, \Sigma^{-1}). \quad (4)$$

The distribution of λ has full support over the set of data points. This implies that samples from the prior on cluster means μ_j also have full support over the data points. The prior also implies that values of μ_j that are near the sample mean are most likely. A potential problem with this prior is that it puts more weight on the center (sample mean) of the data points, but this does not necessarily correspond to a place. For example, in large cities people often commute

⁴ I.e., Inverse covariance.

⁵ The product βW^{-1} corresponds to the expectation of the Wishart distribution on the cluster precision matrices.

for a long period of time to get to work. In this case, the sample mean corresponds to the midpoint of the travel route and samples from the prior on cluster means rarely fall near the actual clusters (home or work). Thus, the algorithm can take longer time to convergence. An alternative is to assign λ a uniform distribution over the set of data points.

The matrix R specifies the precision matrix for the cluster means. Intuitively, we would want the expectation of the distribution on R to correspond to the sample precision Σ as in this case the values for μ_j are on average drawn from a distribution that is specified by the sufficient statistics of the data. We also have to ensure that the resulting Wishart distribution is well defined⁶. This can be achieved by assigning the following distribution on R :

$$R \sim \mathcal{Wi} \left(2, \frac{1}{2}\Sigma \right). \quad (5)$$

The hyperparameters for the prior on precision matrices are more complicated. We start from the variable β , which defines the degrees of freedom for the Wishart distribution on S_j . We do not want to limit the size of clusters beforehand and hence we need to assign a vague prior on β . However, we also need to ensure that the Wishart distribution over S_j remains well defined. These two goals can be achieved by assigning β a flat, continuous distribution over the interval $[1, \infty)$. In order to achieve this, we consider the variable $(\beta - 1)^{-1}$ and assign a Gamma prior for it:

$$(\beta - 1)^{-1} \sim \mathcal{G} \left(\frac{1}{2}, 2 \right). \quad (6)$$

Samples for $\beta - 1$ follow a flat inverse-Gamma distribution and they are within the interval $(0, \infty)$. Thus the distribution of β is as desired.

For the hyperparameter W , i.e., the inverse scale matrix of the prior on S_j , we assign the following Wishart prior:

$$W \sim \mathcal{Wi} \left(2, \frac{1}{2}\Sigma^{-1} \right). \quad (7)$$

The expectation of W equals the sample covariance and, since the expectation of S_j equals βW^{-1} , samples from S_j are on average scaled variants of the sample precision matrix.

Our model specification is lacking a prior for the cluster indicators c_i . We can consider our model as a limiting case of a mixture model where the number of components goes to infinity, and the mixing proportions have been integrated out. Following Neal [28], the prior distribution of c_i can be written in the following form:

$$\begin{aligned} c_i = j | \mathbf{c}_{-i} &\sim \frac{n_{-i,j}}{n - 1 + \alpha} \\ c_i \neq q | \mathbf{c}_{-i} &\sim \frac{\alpha}{n - 1 + \alpha} \quad (\forall q \in \{1, \dots, k\}). \end{aligned} \quad (8)$$

⁶ A Wishart distribution $\mathcal{Wi}(b, W)$ is well defined whenever the $p \times p$ matrix W is positive definite and $b \geq p$ holds for the degrees of freedom parameter b .

Here $n_{-i,j}$ denotes the number of data points that belong to cluster j when the data point i is ignored. The variable α is the concentration parameter of the Dirichlet process prior that, together with the priors on μ_j and S_j , governs the rate at which new clusters are created and \mathbf{c}_{-i} is a vector that contains all other cluster indicators except c_i , i.e., $\mathbf{c}_{-i} = (c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_n)$.

The support of the prior on c_i is the countably infinite set $\{1, 2, \dots, k, \dots\}$ where k denotes the number of clusters that have currently data points associated with them. For each of the represented clusters $j \in \{1, \dots, k\}$, the prior assigns a probability mass of $n_{-i,j}/(n - 1 + \alpha)$. A probability mass of $\alpha/(n - 1 + \alpha)$ is assigned for all of the unrepresented clusters combined. Thus, although the number of clusters is potentially infinite, only some of them are represented at a given time and we do not need to make a distinction between the clusters that are unrepresented.

To finalize our model specification, we need to assign a prior on the concentration parameter α . Again, we assign a vague inverse-Gamma prior so that

$$\alpha^{-1} \sim \mathcal{G}\left(\frac{1}{2}, 2\right). \quad (9)$$

This prior results in a flat distribution that has support over $(0, \infty)$.

Algorithm

In order to utilize the model, we need to be able to compute summaries for the parameters from the posterior distribution of the parameters given the data. A standard way to achieve this in a Bayesian framework is to use Markov chain Monte Carlo (MCMC) techniques. In our case we use Gibbs sampling (see [7,28]), which is a MCMC algorithm that sequentially updates each parameter in turn. The updates are sampled from a probability distribution that is conditioned on the values of the other parameters. Thus, when sampling a new value for a specific parameter, we keep the values of all other parameters fixed. Detailed discussion about MCMC is out of scope of the paper and we refer to [29] for more information.

A high-level description of the algorithm is shown in Alg. 1, and the sampling distributions that are needed to perform Gibbs sampling are given in the Appendix. To improve the speed of convergence, we sampled the values of the cluster parameters and hyperparameters nine times as often as the cluster indicators c_i . In other words, we set the threshold value in Alg. 1 to 10.

Two parameters, α and β , cannot be sampled using traditional methods as their conditional distributions do not correspond to any density that is known in closed-form. In order to sample α , we used the scheme proposed by West [30]. In this scheme, the first step is to sample the value of an auxiliary variable ϕ that depends on the number of parameters and the current value of α . After this, the new value of α can be drawn from a distribution that corresponds to a mixture of two Gamma distributions. The sampling formulas are given in the Appendix and we refer to [30] for more details. Sampling β is more complicated.

Algorithm 1. Gibbs sampler for the model

```

1: Input: data  $\mathbf{y}$ 
2: Initialization:
3: Compute sufficient statistics  $\bar{\mathbf{y}}$  and  $\Sigma$ 
4: Create a single cluster and assign all data points to it (i.e.,  $\mathbf{c} = (1, \dots, 1)$ )
5: Draw initial values for the hyperparameters
6: Sample parameters for the first cluster using the priors
7: repeat
8:   if iterations since last cluster indicator update  $<$  threshold then
9:     for each active cluster component  $c$  do
10:      Sample new value for  $\mu_c$  and  $S_c$ 
11:     end for
12:     Sample hyperparameters  $\lambda, R, W$ 
13:     Sample  $\beta$  using Adaptive Rejection Sampling
14:     Sample auxiliary variable  $\phi$  from a Beta distribution
15:     Compute mixture weight  $\pi_\phi$ 
16:     Sample  $\alpha^{-1}$  from a mixture of two Gamma distributions using  $\phi, k$  and  $\pi_\phi$ 
17:   else
18:     for each data point  $y_i$  do
19:       if iterations  $>$  burnout period then
20:         Store current values
21:       end if
22:       Construct the sampling probabilities for the represented clusters given  $y_i$ 
23:       Create a Monte Carlo estimate for the probability of the unrepresented
         classes
24:       Sample new value for  $c_i$  using the constructed probabilities
25:     end for
26:   end if
27: until convergence

```

Rasmussen [7] observed that the distribution of $\log \beta$ is log-concave, which makes it possible to use adaptive rejection sampling [31] for sampling new values of β . The formulas required to perform adaptive rejection sampling are also given in the Appendix.

Performance

The performance of the Dirichlet process algorithm depends, among other things, on the number of points and on the spatial distribution of data. When the data is relatively evenly distributed, the cluster indicators mix properly and the algorithm converges rapidly. However, when the data is spread out, i.e., it has long and narrow commuting traces (for example, the Innsbruck dataset in Sec. 5), the mixing is much slower. In general, for a given number of clusters, the cluster parameters converge in few hundred (100 - 500) iterations, but the cluster indicators may require several thousands, or even hundreds of thousands, of iterations to converge. The development of inference algorithms for Dirichlet process models is currently an active research area and many improvements have been recently suggested in the literature [32,33].

5 Experiments

5.1 Datasets

We have evaluated our approach using two different datasets. The first dataset has been collected in the city of Enschede, (the Netherlands) and the second dataset has been collected in Innsbruck (Austria). In the following we briefly describe these two datasets. The datasets are shown in Fig. 1.

Enschede. The first dataset that we consider has been gathered by a single user in the city of Enschede. The test subject lives in the city and the measurements have been gathered over a period of one year. Hence, this dataset is a good representative of location traces collected from daily life situations. The data collection was based on voluntary participation. The data was collected using a Nokia 6680 mobile phone and an external Bluetooth GPS receiver (Emtac S3 BTGPS). The GPS measurements were collected whenever the GSM base station to which the device is connected changed. In total, the data set contained over 19000 location measurements. However, most of the measurements were duplicates and there were only 700 distinct GPS measurements. Most of the duplicates correspond to indoor measurements as commonly used GPS devices return the last known GPS measurement when they lose the signal. Moreover, as the data collection was voluntary, the user mainly collected data during working hours.

Innsbruck. The second dataset that we consider has been collected in Innsbruck during UbiComp 2007. The data has been collected by a single user

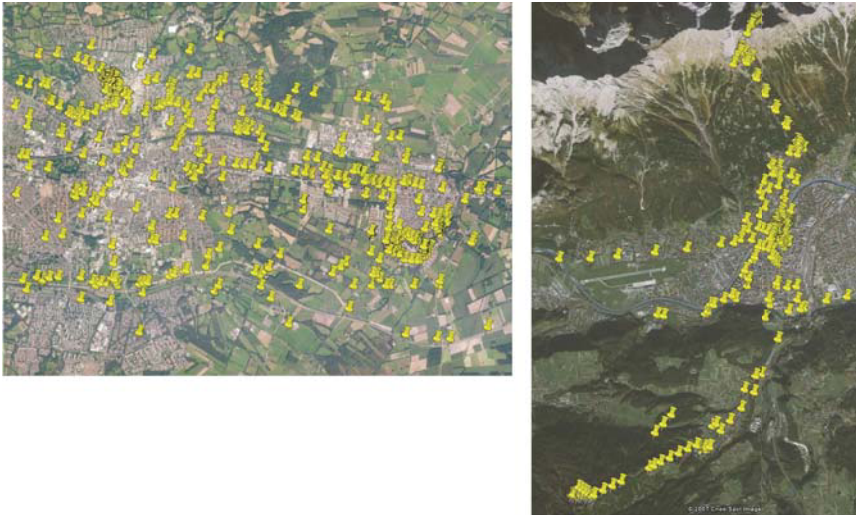


Fig. 1. A visualization of the datasets that we use in our experiments. The figure on the left-hand side shows the preprocessed Enschede dataset and figure on the right-hand side shows the preprocessed Innsbruck data.

using a Nokia N95 mobile phone and a Holux GPSlim 236B Bluetooth GPS receiver. The dataset contains 530 unique measurements. The measurements were collected by sampling the GPS receiver once every minute. The measurements contain both work and tourism related location traces. Hence the setting in the Innsbruck dataset nicely complements the Enschede dataset.

5.2 Experimental Setting and Evaluation Metrics

Before running our algorithm on the location traces, we performed a sanity check that removed unrealistic observations. GPS receivers occasionally give measurements that are suddenly off by several hundreds of kilometers; though, our experience suggests this is extremely rare and occurs mainly on cold starts. Nevertheless, the sanity check ensures that when this event occurs, the faulty data is ignored. As another preprocessing step, we removed all duplicate measurements. While it might seem that we lose information by dropping data, the removal of duplicates does not affect the clustering accuracy of the Dirichlet process algorithm. This is because, when the number of points is small, the spatial distribution of points, i.e., how close neighboring points are to each other, dominates the clustering. In frequently visited regions the spatial distribution is typically compact whereas when the user is commuting the distribution is more spread out.

After removing the duplicates, we ran the Dirichlet process algorithm on the data. For both data sets we ran around 225 000 iterations. From the results, we computed summaries of the mean and precision matrices for the clusters. We also performed post-processing on the results. In the post-processing phase we pruned out clusters that had large variance. From the results we observed a clear threshold as a fraction of the clusters had a relatively small variance whereas the remaining clusters tended to have a larger variance. To select the best cutoff threshold, we used agglomerative clustering on the cluster variances; see Fig. 2. This gave us thresholds that were around 1.0×10^{-5} . Note that

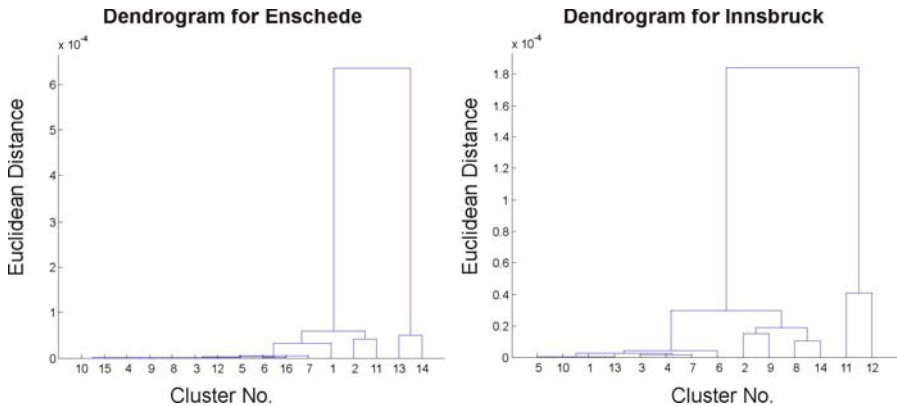


Fig. 2. Dendrograms for cluster variances in the Enschede and Innsbruck datasets

we are considering coordinate units, and this threshold value corresponds to approximately 100 meters. We also removed clusters whose relative frequency (i.e., n_j/n) was smaller than 3% as these clusters are unlikely to correspond to meaningful places.

After the post-processing, we visualized the clusters using Google Earth. For visualization we use the 95% error ellipses, which correspond to the 95% confidence region around the mean of a cluster. We showed the resulting clusters to the user whose data was used. We asked the user to label the clusters and to assess the quality of clustering. Although the evaluation procedure we use is non-standard for evaluating machine learning algorithms, it provides an intuitive way for evaluating the accuracy of the extracted places. A similar evaluation procedure has also been used in human computer interaction research [9]. To provide a comparison against other techniques, we also repeated the same experiments using the k-means algorithm. When we ran the k-means algorithm, we used the same number of clusters that our algorithm was able to identify from the data and we also performed the same pre- and post-processing steps. The results of our experiments are discussed in the next section.

5.3 Results

Enschede. The results for the Enschede dataset are shown in Fig. 3. The algorithm discovered 16 clusters, of which 4 were considered meaningful after post-processing. The size of the clusters varied from three data points to 282 data points. Although we considered only GSM transition points, the algorithm was able to identify the home and office clusters exactly. In addition to home and work, the algorithm was able to identify a park area. The fourth cluster corresponded to regions around work. Hence, the algorithm was able to detect two partially overlapping work clusters. The k-means algorithm, on the other hand, outputted 6 clusters, most of which were relatively large



Fig. 3. Results for the Enschede dataset. The figure on the left-hand side shows the results of the Dirichlet process clustering and the results on the right hand side show the results of the k-means algorithm.



Fig. 4. Results for the Innsbruck dataset

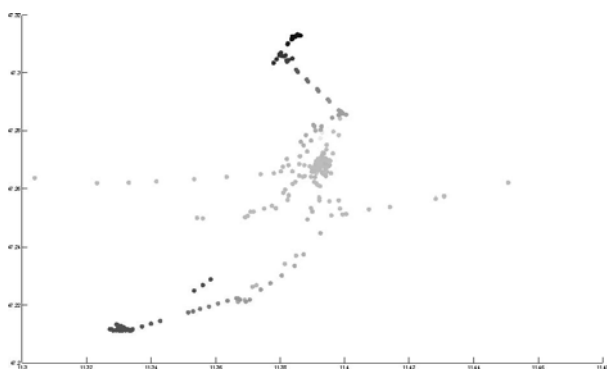


Fig. 5. Altitude plot for Innsbruck data. The darker the color, the higher the altitude.

and meaningless. One of the clusters corresponds to work and one corresponds to home, but both cover a much larger region than the Dirichlet clustering. Since the data collection was based on voluntary participation, the original data consisted mainly of commuting traces and measurements from home and office. However, the results suggest that the Dirichlet clustering is much better in handling noise caused by the irregular sampling of GPS measurements.

Innsbruck. The results for the Innsbruck dataset are shown in Fig. 4. Initially, the algorithm was able to detect 14 clusters. The size of the clusters ranged from 7 data points to 195 data points. After the post-processing step we were left with 5 clusters, all of which were meaningful. One of the clusters corresponded to the hotel where the user was staying during the conference. From the downtown area the algorithm was able to detect two other clusters that were centered around locations where the person had eaten. The remaining places corresponded to locations on top of a mountain (2 places,

one corresponds to the location of the UbiComp banquet). The data did not contain the conference venue, because the person did not gather data during the conference sessions. Again, the results of the Dirichlet process clustering are much more compact than the results of k-means, which identified 7 places from the data (after post-processing). Moreover, the results of the Dirichlet process clustering have a rather clear pruning threshold, whereas the results of the K-means do not.

From the results of Dirichlet process clustering we observe that there are two clusters (Banquet and Hafelkar) that are not as compact and accurate as the other clusters. Both places are on top of mountains (see the altitude plot in Fig. 5), which indicates that changes in altitude cause some problems for the clustering. Though, the Dirichlet clustering suffers less than k-means. Considering how to reliably take into account also altitude information in the clustering is part of our future work.

6 Conclusions and Future Work

In this paper we have introduced a statistical approach for extracting places from discontinuous location traces. Contrary to most of previous research, our algorithm does not have any tunable parameters. We demonstrated the accuracy and robustness of the algorithm using two real world datasets that exhibit rather different characteristics. Our results suggest that Dirichlet processes are a powerful tool for spatial analysis of location measurements and that they can be used to automatically detect locations that are meaningful to users.

In terms of future work, we are currently extending the model to take altitude information into account. In addition, we are constantly collecting more location measurements and we are also planning to compare the algorithm more extensively against other methods suggested in the literature. However, instead of focusing on multiple persons in the same city, we are focusing on comparing the algorithms in cities with different spatial characteristics. Finally, we plan to speed up the converge of the algorithm by considering an improved inference algorithm.

Acknowledgments

The authors are grateful to Wray Buntine for providing insights into Dirichlet processes and Bayesian modeling. The authors acknowledge Jussi Kollin for the implementation of the adaptive rejection sampling algorithm, Johan Koolwaaij for providing us with the Enschede data, and Patrik Floréen for commenting earlier versions of the paper.

This work was supported in part by the IST Programme of the European Community, under the PASCAL network of excellence, IST-2002-506778. The publication only reflects the authors' views.

References

1. Hariharan, R., Krumm, J., Horvitz, E.: Web-enhanced GPS. In: Strang, T., Linnhoff-Popien, C. (eds.) *LoCA 2005*. LNCS, vol. 3479, pp. 95–104. Springer, Heidelberg (2005)
2. Ashbrook, D., Starner, T.: Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing* 7(5), 275–286 (2003)
3. Relph, E.: *Place and Placelessness*. Pion Books, London (1976)
4. Turner, P., Turner, S.: Two phenomenological studies of place. In: *Proceedings of the 17th Conference on Human Computer Interaction (HCI): People and Computers*, pp. 21–35 (2003)
5. Antoniak, C.E.: Mixtures of Dirichlet processes with applications to Bayesian non-parametric problems. *The Annals of Statistics* 2(6), 1152–1174 (1974)
6. MacEachern, S., Müller, P.: Estimating mixture of Dirichlet process models. *Journal of Computational and Graphical Statistics* 7, 223–238 (1998)
7. Rasmussen, C.E.: The infinite Gaussian mixture model. In: Solla, S.A., Leen, T.K., Müller, K.R. (eds.) *Advances in Neural Information Processing Systems (NIPS)*, vol. 12, pp. 554–560. MIT Press, Cambridge (2000)
8. Saegert, S., Winkel, G.H.: Environmental psychology. *Annual Review on Psychology* 41, 441–477 (1990)
9. Zhou, C., Ludford, P., Frankowski, D., Terveen, L.: An experiment in discovering personally meaningful places from location data. In: *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*, pp. 2029–2032 (2005) *Late Breaking Results: Short Papers*
10. Delamater, J. (ed.): *Handbook of Social Psychology*. Handbooks of Sociology and Social Research. Springer, Heidelberg (2006)
11. Nurmi, P., Koolwaaij, J.: Identifying meaningful locations. In: *Proceedings of the 3rd Annual Conference on Mobile and Ubiquitous Computing (MobiQuitous 2006)*, IEEE Computer Society, Los Alamitos (2006)
12. Zhou, C., Ludford, P., Frankowski, D., Terveen, L.: Talking about place: An experiment in how people describe places. In: Ferscha, A., Mayrhofer, R., Strang, T., Linnhoff-Popien, C., Dey, A., Butz, A., Schmidt, A. (eds.) *Adjunct Proceedings of the Third International Conference on Pervasive Computing (PERVASIVE)* (2005)
13. Otsason, V., Varshavsky, A., LaMarca, A., de Lara, E.: Accurate GSM indoor localization. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) *UbiComp 2005*. LNCS, vol. 3660, pp. 141–158. Springer, Heidelberg (2005)
14. Chen, M.Y., Sohn, T., Chmelev, D., Hähnel, D., Hightower, J., Hughes, J., LaMarca, A., Potter, F., Smith, I.E., Varshavsky, A.: Practical metropolitan-scale positioning for GSM phones. In: Dourish, P., Friday, A. (eds.) *UbiComp 2006*. LNCS, vol. 4206, pp. 225–242. Springer, Heidelberg (2006)
15. Marmasse, N., Schmandt, C.: A user-centered location model. *Personal and Ubiquitous Computing* 6(5-6), 318–321 (2002)
16. Toyama, N., Ota, T., Kato, F., Toyota, Y., Hattori, T., Hagino, T.: Exploiting multiple radii to learn significant locations. In: Strang, T., Linnhoff-Popien, C. (eds.) *LoCA 2005*. LNCS, vol. 3479, pp. 157–168. Springer, Heidelberg (2005)
17. Zhou, C., Frankowski, D., Ludford, P., Shekhar, S., Terveen, L.: Discovering personal gazetteers: an interactive clustering approach. In: *Proceedings of the 12th annual ACM international workshop on Geographic information systems (GIS)*, pp. 266–273. ACM Press, New York (2004)

18. Patterson, D.J., Liao, L., Gajos, K., Collier, M., Livic, N., Olson, K., Wang, S., Fox, D., Kautz, H.A.: Opportunity knocks: A system to provide cognitive assistance with transportation services. In: Davies, N., Mynatt, E.D., Siio, I. (eds.) *UbiComp 2004*. LNCS, vol. 3205, pp. 433–450. Springer, Heidelberg (2004)
19. Hariharan, R., Toyama, K.: Project Lachesis: Parsing and modeling location histories. In: Egenhofer, M., Freksa, C., Miller, H. (eds.) *GIScience 2004*. LNCS, vol. 3234, Springer, Heidelberg (2004)
20. Adams, B., Phung, D., Venkatesh, S.: Extraction of social context and application to personal multimedia exploration. In: *Proceedings of the ACM Conference on Multimedia (MM)*, pp. 987–996. ACM, New York (2006)
21. Liu, J., Wolfson, O., Yin, H.: Extracting semantic location from outdoor positioning systems. In: *Proceedings of the 7th International Conference on Mobile Data Management (MDM)*, IEEE Computer Society, Los Alamitos (2006)
22. Aipperspach, R., Rattenbury, T., Woodruff, A., Canny, J.: A quantitative method for revealing and comparing places in the home. In: Dourish, P., Friday, A. (eds.) *UbiComp 2006*. LNCS, vol. 4206, pp. 1–18. Springer, Heidelberg (2006)
23. Kang, J., Welbourne, W., Stewart, B., Borriello, G.: Extracting places from traces of locations. In: *Proceedings of the 2nd ACM international workshop on Wireless mobile applications and services on WLAN hotspots (WMASH)*, pp. 110–118. ACM Press, New York (2004)
24. Hightower, J., Consolvo, S., LaMarca, A., Smith, I., Hughes, J.: Learning and recognizing the places we go. In: Beigl, M., Intille, S.S., Rekimoto, J., Tokuda, H. (eds.) *UbiComp 2005*. LNCS, vol. 3660, pp. 159–176. Springer, Heidelberg (2005)
25. Laasonen, K., Raento, M., Toivonen, H.: Adaptive on-device location recognition. In: Ferscha, A., Mattern, F. (eds.) *PERVASIVE 2004*. LNCS, vol. 3001, pp. 287–304. Springer, Heidelberg (2004)
26. Meneses, F., Moreira, A.: Using GSM CellID positioning for place discovering. In: *Proceedings of the 1st Workshop on Location Based Services for Health Care (Locare)*, pp. 34–42 (2006)
27. Gelman, A., Carlin, J., Stern, H., Rubin, D.: *Bayesian Data Analysis*. Chapman & Hall/CRC (2004)
28. Neal, R.: Markov chain methods for Dirichlet process mixture models. Technical Report 9815, University of Toronto, Department of Statistics (1998)
29. Gilks, W., Spiegelhalter, D., Richardson, S.: *Markov Chain Monte Carlo in Practice*. Chapman & Hall/CRC (1996)
30. West, M.: Hyperparameter estimation in Dirichlet process mixture models. ISDS Discussion Paper #92-A03, Duke University (1992)
31. Gilks, W., Wild, P.: Adaptive rejection sampling for Gibbs sampling. *Applied Statistics* 41, 337–348 (1992)
32. Jain, S., Neal, R.: Splitting and merging components of a nonconjugate dirichlet process mixture model. *Bayesian Analysis* 2(3), 445–472 (2007)
33. Daumé III, H.: Fast search for dirichlet process mixture models. In: Meila, M., Shen, X. (eds.) *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 83–90 (2007)

Appendix: Formulas for the Gibbs Sampler

$$\begin{aligned}
p(\mu_j | \mathbf{c}, \mathbf{y}, S_j, \lambda, R) &\sim \mathcal{N} \left(\left(n_j \bar{\mathbf{y}}_j^T S_j + \lambda^T R \right) (n_j S_j + R)^{-1}, (n_j S_j + R)^{-1} \right) \\
p(S_j | \mathbf{c}, \mathbf{y}, \mu_j, \beta, W) &\sim \mathcal{W}_i \left(\beta + n_j, \left(W\beta + \sum_{i:c_i=j} (y_i - \mu_j)(y_i - \mu_j)^T \right)^{-1} \right) \\
p(\lambda | \mu_1, \dots, \mu_k, R) &\sim \mathcal{N} \left(\left(\bar{\mathbf{y}}^T \Sigma + \left(\sum_{j=1}^k \mu_j^T \right) R \right) (\Sigma + kR)^{-1}, (\Sigma + kR)^{-1} \right) \\
p(R | \mu_1, \dots, \mu_k, \lambda) &\sim \mathcal{W}_i \left(k + 1, \left(\Sigma^{-1} + \sum_{j=1}^k (\mu_j - \lambda)(\mu_j - \lambda)^T \right)^{-1} \right) \\
p(W | S_1, \dots, S_k, \beta) &\sim \mathcal{W}_i \left(k\beta + 1, \left(\Sigma + \beta \sum_{j=1}^k S_j \right)^{-1} \right) \\
p(\phi | \alpha, k) &\sim \mathcal{B}e(\alpha + 1, n) \\
p(\alpha^{-1} | \phi, k) &\sim \pi_\phi \mathcal{G} \left(k + \frac{1}{2}, 2 - \log \phi \right) + (1 - \pi_\phi) \mathcal{G} \left(k - \frac{1}{2}, 2 - \log \phi \right) \\
p(c_i = j | \mathbf{c}_{-i}, y_i, \mu_j, S_j) &\sim Z^{-1} \frac{n_{-i,j}}{n - 1 + \alpha} p(y_i | c_i = j, \mu_j, S_j) \\
p(c_i = c^* | \mathbf{c}_{-i}, \alpha) &\sim Z^{-1} \frac{\alpha}{n - 1 + \alpha} \int p(y_i | \mu_j, S_j) p(\mu_j, S_j | \lambda, R, \beta, W) d\mu_j dS_j
\end{aligned}$$

Here Z^{-1} is a normalizing constant, $\pi_\phi = (k - 0.5) / (2n - n \log \phi + k - 0.5)$ and c^* represents a new cluster component. The variable ϕ is an auxiliary variable, which is used for sampling the value of α ; see [30] for details and derivation.

The conditional distribution $p(\beta | S_1, \dots, S_k, W)$ is not of standard form and we sample instead values for $\log \beta$ using adaptive rejection sampling. The formulas for adaptive rejection sampling are:

$$\begin{aligned}
\log p(\log \beta | S_1, \dots, S_k, W) &\propto \log \beta - \frac{3}{2} \log(\beta - 1) - \frac{1}{\beta - 1} - k\beta \log 2 \\
&\quad + \frac{\beta}{2} \sum_{j=1}^k \log |S_j| - k \left(\Gamma \left(\frac{\beta}{2} \right) + \Gamma \left(\frac{\beta - 1}{2} \right) \right) \\
\frac{\partial}{\partial \log \beta} \log p(\log \beta | S_1, \dots, S_k, W) &= 1 - \frac{3\beta}{2(\beta - 1)} + \frac{\beta}{(\beta - 1)^2} + \frac{k\beta}{2} \\
&\quad - k\beta \log 2 + \frac{\beta k}{2} \log |W| + \frac{\beta}{2} \sum_{j=1}^k \log |S_j| \\
&\quad - \frac{k\beta}{2} \left(\Psi \left(\frac{\beta}{2} \right) + \Psi \left(\frac{\beta - 1}{2} \right) \right).
\end{aligned}$$

Here Ψ is the digamma function, i.e., the logarithmic derivative of the Gamma function.