Geographic Segmentation via Latent Poisson Factor Model

Rose Yu[1,*] qiyu@usc.edu
Andrew Gelfand [2] agelfand@yahoo-inc.com
Suju Rajan [2] suju@yahoo-inc.com
Cyrus Shahabi [1] shahabi@usc.edu
Yan Liu [1] yanliu.cs@usc.edu
[1]Department of Computer Science University of Southern California
[2] Personalization Sciences Yahoo Labs

ABSTRACT

Discovering latent structures in spatial data is of critical importance to understanding the user behavior of location-based services. In this paper, we study the problem of geographic segmentation of spatial data, which involves dividing a collection of observations into distinct geo-spatial regions and uncovering abstract correlation structures in the data. We introduce a novel, Latent Poisson Factor (LPF) model to describe spatial count data. The model describes the spatial counts as a Poisson distribution with a mean that factors over a joint item-location latent space. The latent factors are constrained with weak labels to help uncover interesting spatial dependencies. We study the LPF model on a mobile app usage data set and a news article readership data set. We empirically demonstrate its effectiveness on a variety of prediction tasks on these two data sets.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

geographic segmentation, spatial data, mobile app usage

1. INTRODUCTION

The proliferation of in-vehicle navigation systems and GPS-equipped mobile devices over the last decade has resulted in a dramatic increase in the collection of spatial data. Understanding latent structures in such data is crucial to user behavior modeling and other prediction tasks including targeted advertising, personalization and location-based recommender system. Due to privacy concerns or limitations in the spatial resolution of the collection devices, spatial data are often available in an aggregated form, where spatial observations from distinct users are combined into counts occurring at coarse spatial locations.

This paper deals with the geographic segmentation of spatial count data. *Geographic segmentation* refers to the task of dividing observations into distinct geographical regions and identifying abstract structures in these observations [9]. Spatial clustering, e.g. [3, 16] is a common approach to tackle the problem. However, it usually relies on the spatial auto-correlation of actual geography of locations. Our interest in geographic segmentation is motivated primarily by our desire to understand how mobile app usage behavior for different categories of mobile users varies spatially. Our goal in this case is to describe a joint distribution of the number of uses of an app at a particular spatial location using data aggregated from many devices. By doing so we hope to answer some important questions of interest to mobile app developers, marketing researchers and many others. For instance, do people tend to use one app or one type of app when at a particular location? Which locations are likely to result in the use of a particular type of app?

In our setting, geographic segmentation is challenging because our observations have no correlation with the actual geography of the location, but the type of the location. For example, the fact that someone is interested in running a Yelp app, is not because she is at a specific geo-coordinate but because she is nearby a restaurant. Therefore, we need to capture the implicit relationship between a specific geographical area and the type of interest in that area, which is data dependent. Second, the density of the user population varies drastically across different locations. For instance, the number of observed uses of an app in San Francisco will be much larger than the number of uses of that app in Sunnyvale. As a result, aggregate spatial counts data would easily violate the spatial stationarity assumptions of many standard statistical methods. Finally, while aggregate statistics can mitigate the variability in individual usage data, it can also result in the accumulation of additive noise. This requires the model to take into account the intrinsic uncertainties in the data and automatically learn the underlying correlation.

To collectively address all these challenges, we develop a novel hierarchical Bayesian model, called the Labeled Poisson Model (LPF), to describe spatial count data. As the name suggests, the model describes the counts at a spatial location as a Poisson distribution with a mean that factors over a joint item-location latent space. The LPF model is

*Part of the work done while the author was interning at Yahoo Labs

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.

ISBN 978-1-4503-3716-8/16/02...$15.00

DOI: http://dx.doi.org/10.1145/2835776.2835806

also semi-supervised and uses weak labels on the geographic observations to help uncover underlying spatial dependencies. We study this model on two different data sets. One is a data set of mobile app usage collected by Yahoo Aviate, an intelligent launcher for Android devices; the other is a data set of the Yahoo news articles that were clicked on in different regions of the US. We use the LPF model to predict the number of uses of an app or clicks of a news article at a particular location and evaluate its efficacy by holding out the observed item counts at a set of locations (the test set) and observing how well the model can recover these held out values.

We observe that the semantics of a location, rather than its spatial coordinates (e.g. longitude, latitude), play a key role in the task of geographic segmentation. For mobile app usage, the types of business venues near a location influence the distribution of app activation at that location. For news data, demographic information such as the population, median age and income are important factors in determining the number of clicks in a location. Our proposed LPF model is able to utilize such information to infer the underlying spatial correlations and more accurately estimate the counts at a location. Moreover, our model is privacy preserving as it exploits the aggregate statistics rather than individual data. Our work provides some insight into the design of a general-purpose geographic segmentation strategy.

2. RELATED WORK

The use of aggregate statistics for behavior profiling is a fairly common practice in the data mining community. The early work of [18, 15] discovered web usage patterns by clustering page views and navigation paths across all users to a particular site. More recently, the authors in [10] use an aggregated set of page views and user transactions to generate a set of overlapping usage profiles. These usage profiles are then employed in a "personalization engine" to tailor the set of pages shown to a user when they first visit a site. The authors in [12] studied the reliability of crowd-sourced information, where the information (e.g. an entry in *Wikipedia*) from one source can be influenced by other sources. They propose a probabilistic model to tease out the dependency between sources and uncover the true information.

Geographic segmentation is of particular interest in marketing research, where the goal is to divide a market into distinct geographical regions (e.g. countries, cities or neighborhoods), so that a company or organization can use the market segmentation to, for example, decide how to market or promote a product in a region (see e.g., [9] for an overview of the geographic segmentation problem). Geographic segmentation has also seen successful applications in detecting crime hot spots [9] and geographical annotations[23]. Spatial clustering is a common approach in geographic segmentation. Many methods for finding clusters in spatial data exist (see e.g., [11]), and density-based methods that divide the spatial observations into distinct groups are particularly popular. DBSCAN [3] is one such density-based algorithm that works by first clustering points that lie in close spatial proximity to each other, and then removing outliers whose nearest neighbor is farther than some pre-specified threshold.

To our knowledge, geographic segmentation of mobile app usage has not yet been studied. Existing work [1, 8] has shown that time of day, location (e.g., whether at home, work, or out at a restaurant), activity (e.g., whether walking, running, or driving), and even social setting drive app usage behavior. However, this work did not consider location in a systematic way, instead mapping location to a small set of categories (e.g., at home, at work, or other). Location semantics has been studied in geographic topic modeling literature [21, 5, 22]. However, location semantics for mobile app usage stays under-explored. Yet other work [20, 6] has focused on location-based recommendation, which is somewhat different from our problem. In there, the goal is to find a set of spatially relevant items (e.g. nearby restaurants, local events or even mobile apps) for a particular user.

(a) Dialer (b) Yelp

(c) FlappyBird (d) Snapchat

Figure 1: Heatmap of the number of open events for the Dialer, Yelp, Flappy Bird and Snapchat apps. Opens are aggregated for 100 active devices over a 7 month period in the San Francisco Bay area. Red denotes a higher frequency of open events.

3. EXPLORATORY ANALYSIS

We conduct an exploratory analysis of spatial counts for two different application areas - namely, mobile app usage and news article click data. After introducing each of these data sets, we study the distribution of their spatial counts and check for the existence of spatial correlation. We also identify important characteristics of the geographic distribution in these settings.

3.1 App Usage Data

We collected app usage data from 14,836 mobile devices on which Yahoo Aviate [1] was installed. As mentioned, Yahoo Aviate is a mobile launcher for Android devices. The Android operating systems reports every time a user opens an app or brings an app to the foreground. For each such app open event, Yahoo Aviate records the name of the app that was opened, the time at which it was opened and the approximate location (latitude, longitude) of the device when

[1]https://play.google.com/store/apps/details?id=com.tul.aviate

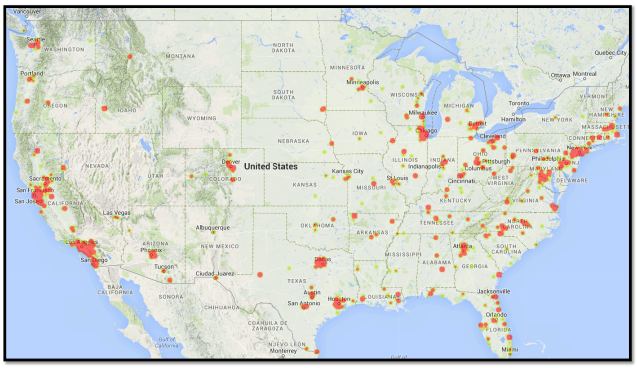


Figure 2: Heatmap of the number of articles read in different locations in the United States.

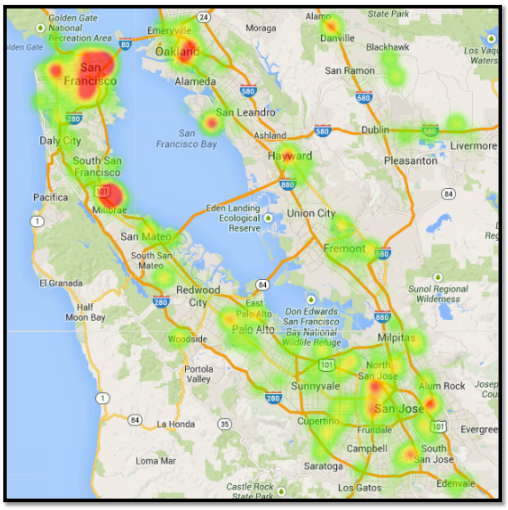


Figure 3: Heatmap of the number of open events for the 100 most popular apps. Opens are aggregated for 100 active devices over a 7 month period and red denotes a higher frequency of opens.

the app was opened [2]. App open events were recorded over a 7-month period for devices in the San Francisco Bay area.

Figure 1 is a heat map visualizing the number of opens of four popular apps - namely, Dialer, Yelp, Flappy Bird and Snapchat. The numbers of opens of each app are aggregated across the 100 devices with the most total open events. Areas where an app was frequently opened are shown in red and areas of infrequent app opens are uncolored. Interestingly, from this figure it appears that how frequently an app is opened at a location can be explained (at least in part) by looking at both the type of venues near that location and the intended functionality of the app. For example, Dialer, which is a communication app, is frequently used by the San Francisco Pier 39, a popular tourist destination. Yelp, which is a travel & local recommendations app is popular in Santana Row, an area with many shops, restaurants and bars. Flappy Bird, which is an arcade game, has a similar distribution to Yelp, which suggests that games may be popular in areas of commerce as well. Finally, Snapchat, a communication and social networking app, is highly concentrated near Stanford, a university area.

The results in Figure 1 suggest that mobile app usage at a location may be influenced by the types of business nearby. In order to understand how the frequency of all app opens varies spatially, we next aggregate the open events of the active users across the 100 most popular apps. Figure 3 displays the counts after this additional level of aggregation. Note that while the app opens are spread over the entire bay area, apps are opened with the highest frequency near the bay area's three major airports - SFO in Millbrae, SJC in San Jose and OAK in Oakland. One possible explanation for the phenomena is that people tend to open apps with increased frequency while waiting for flights.

3.2 Yahoo News Data

We also collected data from the interaction of 182,355 users with the news stream on the Yahoo homepage. The Yahoo stream shows a personalized list of news articles to each user visiting the page. Every article in the stream that a user clicks on is recorded. For each such article, we also record a set of categorical labels describing what that article is about, as well as the zip code from which the user's click originated. The categorical labels are topical in nature, such as 'Sports', 'Art', and 'Education', and an article can be labeled with multiple categories. The categories associated with an article are obtained by classifying the news article via a multi-labeled topical classifier.

We collect the click records per article and map each postal code to a longitude and latitude point [3]. Figure 2 is a heat map showing the counts of clicks in each postal area. For ease of plotting, we uniformly sampled 1,000 locations. Unlike the app usage data, the article click data are spread across the entire US. Big cities like San Francisco, Los Angeles and New York, which have large populations experience the largest number of clicks, while clicks are far more sparse in less densely populated areas. In addition, the spatial resolution for the article click records is at the postal code level. Since a zip code is likely to contain many types of businesses, using business categories to provide semantics for a location (as we did for the app usage data) will be of little help. For this reason, we choose postal code level population and demographic information to provide semantics for a location.

3.3 Distribution of Spatial Events

We analyze the distribution of app open and article click counts across all locations. Figure 4 is a log-log plot that tabulates the number of location cells at which a given event occurred. For Aviate app usage data, we use a $10^3 \times 10^3$ grid. For article click data, we use a $10^2 \times 10^2$ grid. The results of Aviate app usage data are shown in the top row and the news article results are shown on the bottom row. The left

[2]Note that an open event does not necessarily correspond to actual usage. Some apps, such as music players, run in the background. Nonetheless, we use the number of opens as a proxy for app usage.

[3]This was done using the Geocoding library: https://developers.google.com/maps-engine/documentation/geocode

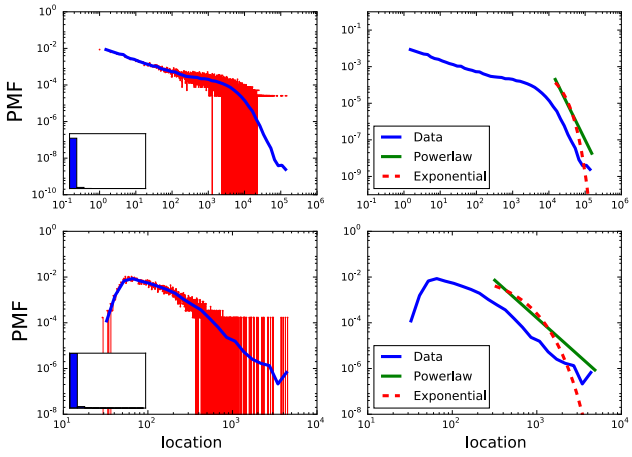


Figure 4: Plot of the number of locations at which a given event count occurred for Aviate app open events (upper row) and Yahoo! news article click events (bottom row). Left column is the raw probabilistic mass function. The right column shows the fit of the power-law and exponential distributions to this count data.

column contains plots of the empirical distributions of each data set. Interestingly, we see that while the count data are generally sparse, they closely follow the power law distributions. The right column shows the fit of the power law distribution to these empirical distributions. For comparison, we also fit the exponential distribution. For both data sets, we observe a good fit for the power law distribution, which is aligned with the findings in [19] on spatial event frequency over distances. The fact that aggregated spatial count data is power-law distributed suggests that we need a heavy-tailed distribution to model frequency counts. This finding motivates our choice of the Poisson distribution in the LPF model.

3.4 Study of Spatial Variation

The analysis in the previous two subsections show the existence of spatial auto-correlations in the aggregate counts of mobile app usage and news article click records. However, the counts do not seem to vary smoothly over space, meaning that the counts in one location do not seem similar to the counts in nearby locations. Local smoothness can be quantified by computing the correlation between the differences in observed counts and geographical distance. We evaluate the spatial smoothness of the app usage data by drawing a grid over the bay area and binning the app open events at different locations into each of the grid cells. We compute the Pearson correlation coefficient between differences in app usage and the distance between cell centers. Table 1 displays the correlation between app usage and inter cell distance for several differently sized grids. Note that spatial correlation can also be evaluated using variogram or auto-variogram with different covariance functions, here we use Euclidean distance as a simple example. We observe that almost no-correlation exists between app usage and the distance between location centers. As a result, we cannot assume that people living in a neighborhood tend to use similar mobile applications.

Table 1: Pearson correlation coefficient between the difference in the distribution over opens of popular apps in two cells i and j and the euclidean distance between cells i and j.

Grid Size	100	400	900	1600	2500
Pearson	-0.0406	-0.0327	-0.0158	-0.0221	-0.0167

The key insight we draw from the exploratory analysis in this section is that a location's semantics, either venue categories or demography, affect the distribution of counts in that location. In the app usage data set, the types of venues near a location can provide a hint as to the users' intent when opening an app. For example, when a user is in an area with many bars and restaurants and they open a restaurant review app, such as Yelp, their intent is probably to eat. Or, if they are in a university or high school area and they open SnapChat, their intent is probably to communicate with their friends. For the news article click data, we have noticed that zip code level demographic information, such as the median household income, median age and even total population can affect the types of articles clicked. These observations suggest that the semantics provided by categorical labels of a location, rather than simply the spatial coordinates, are needed when describing correlations in the observed spatial count data.

In the next section, we propose a model that uses the correlation between different location semantics and item types to model the intensity of aggregate spatial events. Often times, an app may be opened (or an article may be clicked on) for different reasons and the semantics associated with a location may also be uncertain. For instance, the number of opens of the Yelp app at a location can be explained by the fact that Yelp is an app used to recommend local restaurants, shops and services and also by the presence of restaurants, shops and services near that location. Therefore, we adopt a hierarchical Bayesian approach that models the uncertainty in the categorical labels assigned to each location and each item. Our approach borrows from the Labeled LDA model [13] and marries the multi-label supervision common in document classification with co-occurrence preference indicators commonly used in collaborative filtering models.

4. LABELED POISSON FACTOR MODEL

In this section, we formally introduce the Labeled Poisson Factor (LPF) model. The LPF model is a probabilistic graphical model that describes a process for generating aggregate spatial count data. Much like the probabilistic factor model of [14], the frequency of events at a location are due to the interaction of latent factors - in this case an item latent factor and a location latent factor. However, we associate labels to the dimensions of the latent space and use supervision to constrain the set of latent factors that can be used to explain the count at a particular location. In the remainder of this section, we elaborate on the generative process of our model and describe a collapsed Gibbs sampler used for learning and inference. The tedious derivations of the sampler are omitted for clarity of presentation.

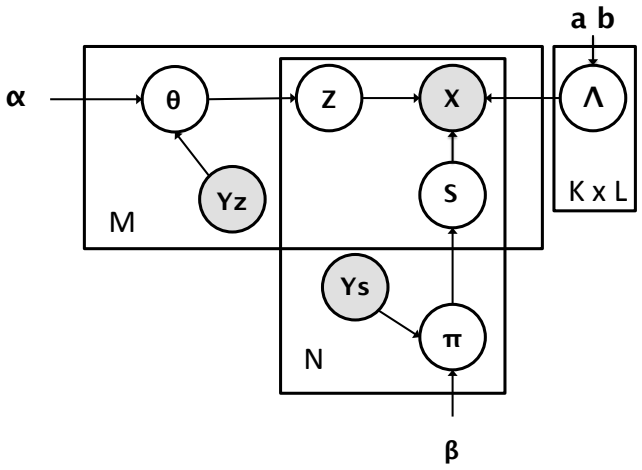


Figure 5: Graphical model of Labeled Poisson Factor model. Shaded circles are observations, blank circles are latent variables and the variables without a circle are model parameters.

Table 2: Latent Poisson Factor Model Notation

Symbols	Semantics
α	Dirichlet prior for item category
β	Dirichlet prior for location category
θ	Item category distribution parameter
π	Location category distribution parameter
Z	Item category indicator
S	Location category indicator
Yz	Item category labels
Ys	Location category labels
X	Spatial activity frequencies
Λ	Intensity distribution parameter
a, b	Shape, scale for Gamma prior distribution

The LPF model can be represented by the graphical model in Figure 5. Let X be an M by N data matrix describing the aggregated counts for the M items at N different locations, where X_{ij} is the aggregated count of item i at location j. In our setting, the item counts are either the number of opens of an app or the number of clicks on a news article. Yz is an M by K binary label matrix for each item, where $Yz_{ik} \in \{0, 1\}$ is 1 if item i has label k and there are a total of K possible labels. Ys is an N by L binary label matrix for the N different locations, where $Ys_{jl} \in \{0, 1\}$ is 1 if location j has label l and there are a total of L possible labels. In the app usage setting, the L different labels for a location come from Foursquare, which describe the types of business venues near that location. For example, a location may be considered a 'Food' and 'Nightlife' area, but not a 'Residential' area. In the news article setting, labels are derived from demographic data, such as median household income and age.

Since our count data exhibits power-law properties, we choose to model every count X_{ij} in X using a Poisson distribution. Since we also observed a co-occurrence of certain types of apps or news articles with specific types of locations, we further assume that the Poisson distribution for item i at location j has an intensity (mean) that factorizes over an items latent factor and a locations latent factor. Each item i is modeled as a multinomial mixture of item categories θ_i. However, rather than being a multinomial with K outcomes, the distribution of categories for item i, θ_i, is restricted so that it is only defined over the categorical labels assigned to item i in the matrix Yz. Each location j is also modeled as multinomial mixture distribution of categories π_j, that is similarly restricted to the categorical labels for location j given in matrix Ys. When no positive labels are observed for item i or location j, we assume that θ_i and π_j have all K and L possible outcomes, respectively.

Most existing Poisson factorization approaches model the Poisson intensity parameter for element X_{ij} as the inner product of two latent vectors: $X_{ij} \sim \text{Poisson}(\theta_i^T \pi_j)$. In our setting, this would imply an independence of the location factors and the item factors. We relax this assumption by imposing a joint latent space, Λ, and allow each item to pick a category based on location and vice versa. This modification describes the spatial data more accurately and introduces more flexibility into the model. The binary labels in Yz and Ys can only provide weak guidance to the actual categories contributing to an observed count because of the mutual dependence of item and location categories in the joint latent space. The binary labels nonetheless constrain the priors on latent factors: the category of an item or a location is constrained to belong to the categories indexed by the non-zero entries of the labels in Yz and Ys, respectively.

Our model is a variation of the Bayesian Poisson factorization model [24], which replaces the usual Gaussian likelihood

Algorithm 1 Generative Process of the LPF model

Input: hyper-parameter α, β, a, b
for $i = 1$ **to** M **do**
  $\alpha_i = \text{SHRINK}(\alpha, Yz_i)$
  Draw $\theta_i \sim \text{Dir}(\alpha_i)$
  for $j = 1$ **to** N **do**
    Draw $Z_{ij} \sim \text{Categorical}(\theta_i)$
  end for
end for
for $j = 1$ **to** N **do**
  $\beta_j = \text{SHRINK}(\beta, Ys_j)$
  Draw $\pi_i \sim \text{Dir}(\beta_j)$
  for $i = 1$ **to** M **do**
    Draw $S_{ij} \sim \text{Categorical}(\pi_i)$
  end for
end for
for $k = 1$ **to** K **do**
  for $l = 1$ **to** L **do**
    Draw $\Lambda_{kl} \sim \text{Gamma}(a, b)$
  end for
end for
for $j = 1$ **to** N **do**
  for $i = 1$ **to** M **do**
    Draw $X_{ij} \sim \text{Poisson}(\Lambda_{z_{ij}, s_{ij}})$
  end for
end for
function SHRINK(u, v)
  $u' = u$
  Squeeze out entries in u' where $v = 0$
return u'
end function

Algorithm 2 Collapsed Gibbs Sampler for the LPF model.

Input: data X, Yz, Ys, training mask I, Dirichlet hyper-parameters α, β, Gamma shape, scale hyper-parameters a, b
repeat
  Randomly initialize Z, S, Λ.
  Initialize category counts nZ, nS as zero matrices
  for $i = 1$ **to** M **do**
    $\alpha_i = \text{SHRINK}(\alpha, Yz_i)$
  end for
  for $j = 1$ **to** N **do**
    $\beta_j = \text{SHRINK}(\beta, Ys_j)$
  end for
  for $i = 1$ **to** M **do**
    for $j = 1$ **to** N **do**
      $nZ_{Z_{ij},i} = nZ_{Z_{ij},i} + I_{ij}$, $nS_{S_{ij},j} = nS_{S_{ij},j} + I_{ij}$
    end for
  end for
  for $i = 1$ **to** M **do**
    for $j = 1$ **to** N **do**
      if I_{ij} **then**
        Sample $Z_{ij} \sim \text{Multi}(1, [\alpha_i + nZ_{:,i}^{\neg ij}] \circ Yz_i)$
        Sample $S_{ij} \sim \text{Multi}(1, [\beta_j + nS_{:,j}^{\neg ij}] \circ Ys_j)$
        Update nZ, nS
      end if
    end for
  end for
  for $k = 1$ **to** K **do**
    for $l = 1$ **to** L **do**
      Count activity $nI_{kl} = \sum_{ij:Z_{ij}=k,S_{ij}=l} I_{ij}$
      $a' = a + \sum_{i,j} X_{ij} \circ I_{ij}$, $b' = b/(1 + nI_{kl} * b)$
      Sample $\Lambda_{kl} \sim \text{Gamma}(a', b')$
    end for
  end for
until Converge

and real-valued representations in probabilistic matrix factorization with a Poisson likelihood and non-negative representations. As pointed out by [4], Poisson factorization is better at handling sparse data and enjoys more efficient inference.

Algorithm 1 describes the generative process of the LPF model in detail. For each item i, we sample a multinomial distribution θ_i over item categories from a truncated Dirichlet prior α_i. The truncated Dirichlet prior α_i is found by using Yz to retain only the parameters in α corresponding to positively labeled categories. In particular, α_i is formed from α by setting the concentration parameters for the unlabeled categories to zero. For each location j, we then sample an independent item factor indicator Z_{ij} from the categorical distribution parameterized by θ_i. In a similar fashion, we generate location factor indicators S_{ij} by first sampling π_j from a Dirchlet prior β_j constrained using the location labels in the matrix Ys and then sampling S_{ij} from π_j for each item i. Finally, we generate the aggregate spatial count X_{ij} according to Poisson distribution with intensity $\Lambda_{Z_{ij},S_{ij}}$. For convenience, we put a conjugate Gamma prior on the joint latent space Λ. We use a Gibbs sampler for inference and learning. For efficiency, we collapse the latent variables of type $\{\theta_i\}$ and space $\{\pi_i\}$. The update steps of the collapsed Gibbs sampler are described in Algorithm 2.

5. EXPERIMENTS

We conduct several experiments to evaluate the capability of the LPF model to accurately capture spatial correlations and predict the number of spatial counts at a particular location, or for a particular item.

5.1 Setup

Our overall experimental setup is shown in Figure 6. In the first stage, we extract labels for each of the M items. In the second stage, we identify the N locations and assign labels to each. In the case of the app usage data, we discretize the geographical area into N equally sized cells; for the news article data, the number of locations N is equal to the number of distinct postal codes. Details on how the labels are extracted follow below. In the final stage, we evaluate the LPF model by holding out the observed counts at some of the locations and assess how well the model can recover those values.

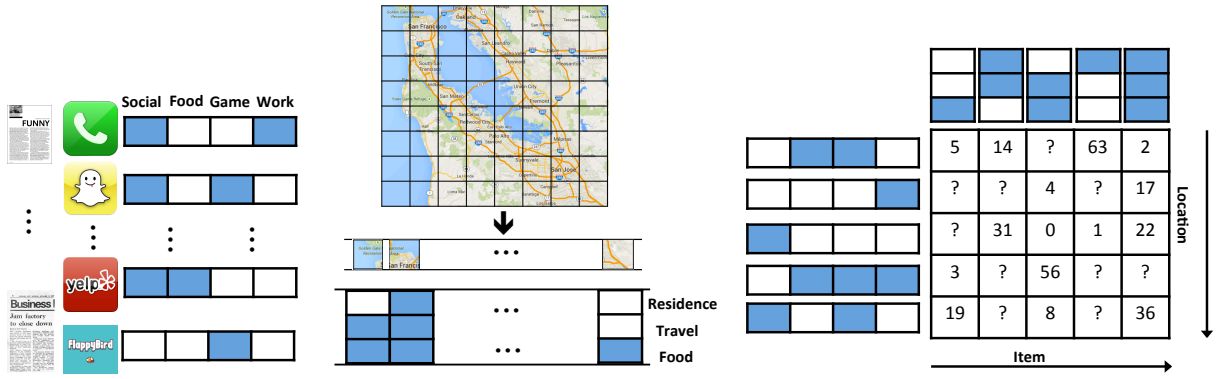
Table 3: Characteristics of the Aviate app usage and Yahoo news article click data sets.

Aviate App Usage		
Time Period	# of Devices	# of Records
7 month	14,836	51,765,517
# of AppOpen	# of LocUpdate	# of Apps
33,896,214	17,869,303	46,248

News Article Click	
# number of users	# number of records
182,355	1,034,615
# of articles	# of locations
30,544	5,907
# of article category	# of location category
21	40

We now describe how the categorical labels were identified for app usage data. The Yahoo Aviate launcher provides a mapping of each app to 24 named collection types, such as 'Productivity', 'News', 'Restaurants', etc. These collection types provide a natural labeling for each app. We discretize the San Francisco Bay Area into equally sized cells and use the Foursquare API to identify the set of business venues in each cell. Each business is associated with one of 10 categorical tags, such as 'Restaurant', 'Nightlife Spot', or 'College or University'. We use these categorical tags to label each cell.

For the Yahoo news article data, we gather demographic information for each zip code area, including the median income, age, number of households and the population density. We then compute the national percentile of each demographic feature and map them to values 0-9. For example, '0' in income indicates the median income at that locale is in the bottom 10% nationally and 1 indicates the median income is in the 10-20% percentile nationally. The key statistics of these two data sets are shown in Table 3.



(a) Item Labeling (b) Location Labeling (c) Matrix Completion

Figure 6: Set up of the LPF model experiments. (a): categorical labeling for items (b): categorical labeling for locations (c): hold out assessment of the aggregated counts.

5.2 Baseline

We compare the LPF model with the following baselines.

- LPF-Loc: the LPF model using only location labels
- LPF-Item: the LPF model using only item labels
- BPF : Bayesian Poisson Factor Analysis [24]
- MFSI: Matrix Factorization with Side Information [7]
- NMF: Non-negative Matrix Factorization [2]
- PM: Poisson Mixture model
- Global Mean: Mean of all training data
- Loc Mean: Mean of each location
- Item Mean: Mean of each item
- All Zero: Always predict zero

We mainly compare the LPF model with other latent factor models. We include LPF-Loc, which is the LPF model with supervision only on the labels of the locations, LPF-Item, which is the LPF model with supervision only on the labels of the items, and BPF to evaluate the benefits of incorporating both item and location semantics for the geographic segmentation task. We also include MFSI and NMF as non-Bayesian factorization methods. The comparison between latent factor models and other density-based spatial clustering techniques such as DBSCAN is not within the scope of the study in this paper. For NMF, we use the source code in http://www.csie.ntu.edu.tw/~cjlin/nmf/others/nmf.py. For MFSI, we use the GraphLab Create.

5.3 Mobile App Usage

We evaluate the predictive performance of the LPF model in three different settings: 1) A static setting; 2) A dynamic setting; and 3) While varying grid size.

Static Setting.

In this section, we evaluate the LPF model in a static setting by aggregating the app usage across all 7 months. We select the 100 most popular apps and bin the San Francisco Bay area into 100 equally sized grid cells. We aggregate the number of opens of each app in each cell across all 14,836 devices. We normalize the counts by the total number of devices and randomly remove 20% of the entries in the 100 by 100 data matrix X. These are the missing values that we try to recover. We repeat the process 10 times and report the average prediction RMSE with respect to the ground truth frequencies.

Table 4: RMSE of the LPF model and several baseline approaches on the Aviate app usage data. App opens are aggregated across 14,836 devices at 100 locations. The results are averaged over 10 random 80% - 20% training-test splits.

Algorithm	Mean	Variance
Labeled Poisson Factor	**1.1293**	0.1569
LPF-Item	1.2200	0.1780
LPF-Loc	1.1484	0.1989
Bayesian Poisson Factor Analysis	1.1927	0.2508
MFSI	1.4111	0.0085
Poisson Mixture	1.3159	0.2270
Non-negative Matrix Factorization	1.2096	0.2513
Global Mean	1.2841	0.2238
Loc Mean	1.2670	0.1870
Item Mean	1.2347	0.2024
All Zero	1.4488	0.2210

Table 4 shows the RMSE comparison of the LPF model and the baselines. Notice that the LPF model achieves the lowest RMSE in this setting. By comparing LPF with LPF-Item and LPF-Loc we can begin to understand the importance of the supervision provided by the item and location labels. When aggregating across the devices, we see that location semantics have a bigger impact on performance (RMSE of 1.22 for LPF-Item vs RMSE of 1.14 for LPF-Loc). On the other hand, LPF has relatively low variance in terms of prediction. This is mainly due to the inherent sparsity in the data, which is alleviated by Bayesian priors and the supervision from labels.

Figure 7 shows the geographic segmentation for a set of cells centered in Palo Alto with grid size 100. This plot is generated by first estimating the location category distribution, π_j, at each location j and then assigning the most probable category $l_j = \arg\max \pi_j$ to each location. We use different colors to denote different categories. Three ma-

Figure 7: Color-coded segmentation of Palo Alto area learned by LPF. The segment type are decided by assigning the most probable category with respect to the location category distribution π.

jor segments are identified in the plot. The brown color highlights the 'school' areas, including Stanford university on the bottom left and Menlo-Atherton High School on the top left. The yellow color represents shopping areas, including Escondido Village, Stanford Shopping Center and Ikea in the top middle. Finally, the orange color denotes hotels that appear near the Palo Alto Golf Course and The Westin Palo Alto. We note that these segments (learned by LPF) are discriminated by their categories rather than their actual geographic coordinates. The results suggest that LPF model can capture latent semantics in spatial count data.

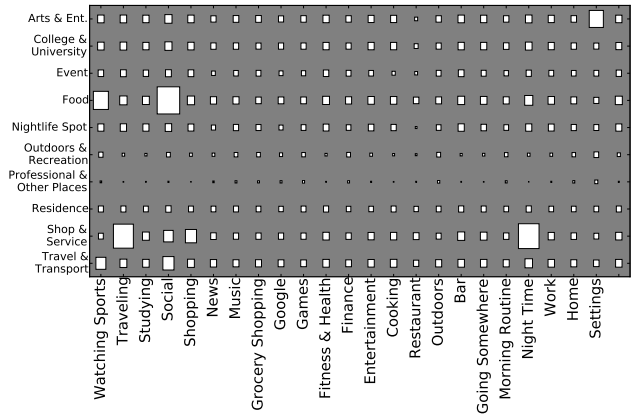


Figure 8: Hinton diagram of the joint latent space intensity, Λ of the LPF model. The area occupied by a square in this matrix is proportional to the magnitude of the intensity of that item-location category pair. So for example, 'Social' apps and 'Food' locations have a very strong intensity, while 'Music' apps and 'Food' locaitons have a much weaker intensity.

Figure 8 shows the Hinton diagram of the intensity of co-occurrence of app and location labels from the matrix Λ learned by LPF. We observe several interesting phenomenon in this learned joint latent space that are supported by our intutition. For example, night time apps are popular in food locations. Shopping apps are popular in shopping and services locations. Such results show that our model can correctly capture the underlying correlation between an app's category and a location's semantics.

Dynamic Setting.

In the dynamic setting, we make a sequence of predictions. Since the Aviate data set spans roughly 7 months, we split the data into 7, one month chunks. We train all methods using the data from month t and predict the number of opens for all apps at all locations in month $t+1$. We normalize the counts with respect to the total number of devices and the number of days in each month. We repeat the training and testing procedure 6 times and report the prediction RMSE for all methods in Table 5 for all time stamps.

In the dynamic setting, the counts in a single time stamp become quite sparse. As a result, the LPF model significantly outperforms the other baseline methods in all 6 of the evaluation periods. Notice that in the static setting, BPF and PM perform pretty well, while in the dynamic setting, the accuracy of both methods degrades. This demonstrates how the joint latent space and the supervision of the LPF can help alleviate the sparsity issue.

Varying Grid Size.

In our experimental setup, we discretize the San Francisco Bay area into evenly sized cells and binned the opens of each app into a cell. A major difficulty with binning in this fashion is choosing the right size for each cell. The size of a grid cell plays an important role as it not only affects the number of observed opens in a cell, but also the categorical labels associated with a cell. Small sized cells would lead to sparse observations. However, when a cell is too large, it will include business from a wide range of categories, which will consequently reduce the effectiveness of the location supervision.

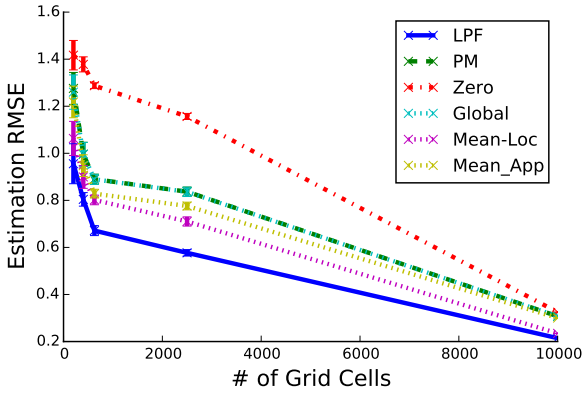


Figure 9: RMSE error bar for the LPF and baselines with respect to grid size over 10 random runs. The number of grid cells increases from 50 to 10,000.

In this subsection, we investigate the impact of differently sized grids on prediction accuracy. Each method is run on a grid ranging in size from 50 to 10,000 cells. For each grid size setting, we make API calls to Foursquare to obtain a finer-grained set of location categories. Figure 9 shows the prediction RMSE error bar with respect to the number of

Table 5: RMSE comparison of the LPF model and several baseline approaches in the dynamic setting, where the app opens in month $t+1$ are predicted using the app open observations in month t. App opens were aggregated over 14,856 devices and 100 locations.

Time stamp	LPF	BPF	PM	NMF	Zero	Global	Loc-Mean	Item -Mean	MFSI
TS 1	**0.2986**	0.6128	0.5271	5.4878	0.6812	0.5265	0.3240	0.5120	1.2331
TS 2	**0.2838**	0.6103	0.5292	5.4912	0.6847	0.5292	0.3167	0.5118	1.2142
TS 3	**0.3150**	0.6461	0.5560	5.4767	0.7159	0.5555	0.3509	0.5363	1.2885
TS 4	**0.3347**	0.6359	0.5663	5.4643	0.7321	0.5661	0.3695	0.5462	1.2473
TS 5	**0.3306**	0.6297	0.5667	5.4715	0.7276	0.5667	0.3700	0.5474	1.2364
TS 6	**0.3112**	0.5139	0.4916	5.5575	0.6019	0.4911	0.3379	0.4730	1.3740

number of grid cells. LPF outperforms the other baselines[4]. As expected, the mean RMSE goes down for all methods as we increase the number of grid cells. This decrease is due to two things: first, as we increase the number of cells in a fixed area, we obtain finer-grained location information. Therefore, the supervision from location tags becomes increasingly sharp. Second, finer-grid binning results in more sparse observed counts. Thus, the absolute values of the frequency counts decreases.

5.4 News Article Clicks

To demonstrate the applicability of the LPF model to other types of spatial count data, we also evaluate on the Yahoo news articles data.

Table 6: RMSE comparison of the LPF model and baseline approaches on the Yahoo news article data. The results are averaged over 10 random split of the dataset using 80% - 20% training-test split.

Algorithm	Mean	Variance
Labeled Poisson Factor	**1.7956**	0.04476
LPF-Item	1.8004	0.04607
LPF-Loc	1.7998	0.04014
Bayesian Poisson Factor Analysis	2.9659	0.04857
Poisson Mixture	2.4134	0.04829
Non-negative Matrix Factorization	3.7669	**0.02345**
Global Mean	2.3753	0.04453
Loc Mean	1.8049	0.04510
Item Mean	2.3001	0.04364
All Zero	3.3315	0.04639

We aggregate the article counts for 182,355 users and 30,544 articles. We randomly select 80% of the entries as training data and repeat this procedure 10 times. Table 6 shows the mean and variance of the prediction RMSE of LPF and the baselines. The LPF model achieves the lowest prediction RMSE in this setting too. Note that the gain from adding the supervision via location category labels is more significant than article category labels. This suggests that the density of news article clicks is more strongly related to location demographics than the topic of an article.

We also visualize the joint latent space intensity, Λ, in Figure 10. The size of a square represents the relative magnitude of the intensity for an item category - location category pair. Four different colors are used to differentiate the four types of demographic information. Though the model assigns almost equal weights for all location categories, there exists strong correlation of articles in Art & Entertainment, Technology & Electronics with the top 10% populations. The consumption of the news article category shows very weak dependencies with income. However, Society & Culture and Art & Entertainment articles tend to appeal more to zip codes with middle aged people.

[4]We omit the results for NMF and MFSI as their performance was significantly worse than the other methods

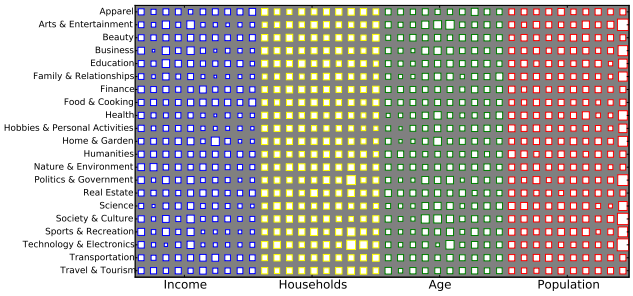


Figure 10: Hinton diagram of the joint latent space intensity on Yahoo news article data learned by LPF model. Colors correspond to different type of category tags: income, households, age and population.

6. CONCLUSIONS

In this paper, we studied the task of geographic segmentation on spatial count data, with applications in mobile application usage and news article readership. Through an exploratory analysis we found that the categories of locations, rather than the geographic coordinates, are more important in those applications. We developed a Bayesian hierarchical model, the Latent Poisson Factor (LPF) model, to capitalize on this observation and demonstrated the efficacy of LPF model at capturing rich spatial correlation structures and predicting aggregate counts on two data sets: Aviate app usage and Yahoo! news article clicks. Experiment results showed that our method can learn sensible spatial latent semantics as well as produce more accurate predictions.

7. ACKNOWLEDGMENTS

This work is supported in part by the U. S. Army Research Office under grant number W911NF-15-1-0491, NSF IIS-1254206 and USC Integrated Media System Center (IMSC). The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agency, or the U.S. Government.

8. REFERENCES

[1] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling asleep with angry birds, facebook and kindle: a large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, pages 47–56. ACM, 2011.

[2] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009, 2009.

[3] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996.

[4] P. K. Gopalan, L. Charlin, and D. Blei. Content-based recommendations with poisson factorization. In *Advances in Neural Information Processing Systems*, pages 3176–3184, 2014.

[5] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM, 2012.

[6] A. Karatzoglou, L. Baltrunas, K. Church, and M. Böhmer. Climbing the app wall: enabling mobile app discovery through context-aware recommendations. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, CIKM '12, New York, NY, USA, 2012. ACM.

[7] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[8] J. J. Levandoski, M. Sarwat, A. Eldawy, and M. F. Mokbel. Lars: A location-aware recommender system. In *Proceedings of the 2012 IEEE 28th International Conference on Data Engineering*, ICDE '12, pages 450–461, Washington, DC, USA, 2012. IEEE Computer Society.

[9] H. J. Miller and J. Han. *Geographic data mining and knowledge discovery*. CRC Press, 2009.

[10] B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data mining and knowledge discovery*, 6(1):61–82, 2002.

[11] R. T. Ng and J. Han. E cient and e ective clustering methods for spatial data mining. In *Proc. of*, pages 144–155, 1994.

[12] G.-J. Qi, C. C. Aggarwal, J. Han, and T. Huang. Mining collective intelligence in diverse groups. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1041–1052. International World Wide Web Conferences Steering Committee, 2013.

[13] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning. Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[14] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20, 2008.

[15] C. Shahabi, A. M. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users Web-page navigation. In *Research Issues in Data Engineering, 1997. Proceedings. Seventh International Workshop on*, 1997.

[16] J. Shi, N. Mamoulis, D. Wu, and D. W. Cheung. Density-based place clustering in geo-social networks. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 99–110. ACM, 2014.

[17] S. Xie, G. Wang, S. Lin, and P. S. Yu. Review spam detection via time series pattern discovery. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 635–636. ACM, 2012.

[18] T. W. Yan, M. Jacobsen, H. Garcia-Molina, and U. Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the Fifth International World Wide Web Conference on Computer Networks and ISDN Systems*, pages 1007–1014, 1996.

[19] M. Ye, D. Shou, W.-C. Lee, P. Yin, and K. Janowicz. On the semantic annotation of places in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 520–528. ACM, 2011.

[20] H. Yin, B. Cui, L. Chen, Z. Hu, and C. Zhang. Modeling location-based user rating profiles for personalized recommendation. *ACM Trans. Knowl. Discov. Data*, 38, 2014.

[21] Z. Yin, L. Cao, J. Han, C. Zhai, and T. Huang. Geographical topic discovery and comparison. In *Proceedings of the 20th international conference on World wide web*, pages 247–256. ACM, 2011.

[22] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Who, where, when and what: discover spatio-temporal topics for twitter users. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 605–613. ACM, 2013.

[23] H. Zhang, M. Korayem, E. You, and D. J. Crandall. Beyond co-occurrence: discovering and visualizing tag relationships from geo-spatial and temporal similarities. In *Proceedings of the fifth ACM international conference on Web search and data mining*, pages 33–42. ACM, 2012.

[24] M. Zhou, L. Hannah, D. Dunson, and L. Carin. Beta-negative binomial process and poisson factor analysis. *arXiv preprint arXiv:1112.3605*, 2011.