Annotating Points of Interest with Geo-tagged Tweets

Kaiqi Zhao Gao Cong Aixin Sun
Nanyang Technological University, Nanyang Avenue, Singapore 639798
kzhao002@e.ntu.edu.sg gaocong@ntu.edu.sg axsun@ntu.edu.sg

ABSTRACT

Microblogging services like Twitter contain abundant of user generated content covering a wide range of topics. Many of the tweets can be associated to real-world entities for providing additional information for the latter. In this paper, we aim to associate tweets that are semantically related to real-world locations or Points of Interest (POIs). Tweets contain dynamic and real-time information while POIs contain relatively static information. The tweets associated with POIs provide complementary information for many applications like opinion mining and POI recommendation; the associated POIs can also be used as POI tags in Twitter. We define the research problem of annotating POIs with tweets and propose a novel *supervised Bayesian Model* (sBM). The model takes into account the textual, spatial features and user behaviors together with the supervised information of whether a tweet is POI-related. It is able to capture user interests in latent regions for the prediction of whether a tweet is POI-related and the association between the tweet and its most semantically related POI. On tweets and POIs collected for two cities (New York City and Singapore), we demonstrate the effectiveness of our models against baseline methods.

1. INTRODUCTION

The prevalence of smartphones enables massive amount of data being generated at unprecedented scale on various social media platforms. On microblogging platforms like Twitter, users update their status, comments on news events, and express their opinions of products, services or locations, in an informal and casual manner. The information contributed by users often covers a wide range of topics. On the other hand, real-world entities have online presence on many social service platforms, such as Foursquare and Google Maps. For instance, Foursquare hosts millions of points of interest (POIs) and users' check-in to these POIs. Many data mining tasks have been conducted on the data from both platforms. On Foursquare, there are studies on the properties of POIs, user behaviours, and POI recommendations [12, 15, 17, 20, 24, 27]. On Twitter, example studies include sentiment analysis, user mobility pattern analysis, and even POI recommendations based on *geo-tagged tweets* (*i.e.,* tweets associated with latitude/longitude coordinates) [1, 11, 12, 28]. However, the two kinds of interesting data have been utilized separately in most studies, even for the same tasks like POI recommendation.

We argue that the two types of data complement each other: POIs contain static information, for example name, address, reviews and tips. Tweets are posted in a dynamic way and contain real-time information (*e.g.,* the restaurant is holding a discount event). In this research, we aim to *annotate POIs with their relevant tweets based on their semantic relatedness*. For instance, if a tweet comments on the service or hygiene standard of a restaurant, then we associate the tweet and the restaurant.

Many applications could be greatly benefitted from such kind of "data integration" from these two types of data. Tweets that are associated with POIs become a complementary data source for real-time event detection, opinion mining and sentiment analysis of POIs. Twitter can also enhance user experiences by supporting POI level geo-tags for tweets instead of coarse-grain location (*e.g.,* city level). When a user posts a tweet, we provide user an option to tag the tweet with the candidate POIs based on semantic relatedness. Map systems such as Google Maps also benefit from the associations between tweets and POIs to support user exploration over real-time information of a POI or a spatial region, for what is happening about the POI or the region [6].

However, determining whether a tweet is semantically related to a POI is challenging, given the volume and shortness of tweets, and the large number of POIs at fine-grained level. Some proposals [3, 19, 22] assume that each geo-tagged tweet is associated to a POI. However, not all geo-tagged tweets are necessarily POI-related in terms of its semantic meaning. In our data collected from Twitter, only 8%-10% of geo-tagged tweets are semantically related to POIs. To the best of our knowledge, no previous work has studied the problem by considering the *semantic relatedness* between geo-tagged tweets and POI. The challenges are two-fold. On the one hand, because the coordinates provided by GPS devices are often not precise, the nearest POI is not the true POI where the tweet was posted in many cases. Only 51% and 17% POI-related tweets were posted nearest to their POIs in our Singapore and New York datasets, respectively. On the other hand, even if a tweet was posted right at a particular POI, it remains hard to determine if the content of the tweet is relevant to the POI. Note that this problem is different from the problem of inferring tweet location, *e.g.,* the coordinates of tweets [1, 11, 23] or the city level location of tweets [9, 16]. These studies assume that the coordinates of tweets are unknown and aim at inferring the missing location information for tweets. They complement our research as we target on the geo-tagged tweets.

We develop two baseline solutions to the problem. The first solution is to first identify candidate POIs which are geographically

CIKM '16, October 24–28, 2016, Indianapolis, IN, USA.

DOI: http://dx.doi.org/10.1145/2983323.2983850

close to a given tweet and then associate the tweet to the top-ranked POIs based on their semantic relatedness. The semantic relatedness can be computed by using the language model of the tweet and the language model of a candidate POI. Alternatively, the tweets can be firstly classified to be POI-related or non-POI-related. Then the POI-related tweets are associated with POIs according to the distances between their language models. We name these two baseline solutions *RANK* and *CLASS*, respectively. Both baseline solutions fail in capturing user behaviour as well as the latent relations between textual content, coordinates, and users of the tweets.

In this paper, we propose a novel supervised Bayesian model (sBM for short) which explores three aspects of the tweets: textual content, coordinates, and user behaviours (user interests in latent regions) under the supervised information of POI relatedness. The proposed sBM model is able to capture user interests in latent regions. Intuitively, the coordinates act as spatial filters to exclude POIs that are far from a given tweet; the user behaviors provide the user's interests in some POIs or the types of the POIs to further narrow the search space, and the textual content helps identify the true POI in the small search space. Compared with the existing geospatial topic models [11, 12, 15, 20, 25, 26, 28, 29], the novelty of our model is two-fold. First, we add a relatedness response to each document to determine whether a tweet is POI-related by considering the context (*e.g.,* the number of words that are not related to any POI, the distance from the tweet to its most semantically related POI, and the region in which the tweet was posted). By associating words and regions to the relatedness response, regions and language models are better fit to solve the relatedness problem. Second, we introduce a set of dummy POIs to model the location of non-POI-related tweets. The dummy POIs benefit the model in that we can model the POI-related and non-POI-related tweets in a consistent manner. Moreover, dummy POIs themselves capture spatial and textual information because each dummy POI has its own language model and coordinates like real POI. In summary, this paper makes the following contributions:

- We define the research problem of annotating POIs with geo-tagged tweets.
- We develop two baseline solutions and propose a novel supervised Bayesian model, which explores three aspects of the tweets: textual content, coordinates, and user behaviours under the supervised information.
- We conduct experiments on real word datasets and demonstrate the effectiveness of the proposed model in annotating POIs with geo-tagged tweets.

The rest of the paper is organized as follows. We survey the related work in Section 2. In Section 3, we formulate the research problem and present two baseline models. The supervised Bayesian model is presented in Section 4. After reporting the experimental results in Section 5, we conclude this paper in Section 6.

2. RELATED WORK

Location Identification: Most related to this work is location identification. We categorize the studies of location identification into two types. The first type of studies aims to associate a POI to a GPS record or geo-tagged post [19]. Lian et al. [19] extract several features between a GPS record and a POI and apply them in a learning-to-rank model, to select the most appropriate POI for the GPS record. The extracted features include the popularity of the POI, distance between the GPS record and the POI, frequently check-in time slots of the POI, and the number of user check-ins at the POI. Our problem is different from this type of studies in that we need to identify the POI-related tweets from those that are not related. That is, we consider the content of the tweets.

The second type of studies associate a POI to tweets or other social media posts [9, 14, 16, 18]. Dalvi et al. [9] propose a probabilistic model to infer the user's location and match tweets to spatial objects such as restaurants. They assume that each user has a location which can be inferred from the user's visit history. They then use the inferred user location together with a language model to identify the spatial object for a tweet. However, a user may travel to many locations and have multiple activity regions in a city [8]. If a tweet is posted at a "Starbucks" that is far from the inferred user location, their model probably matches the tweet to a "Starbucks" near the inferred user location. Moreover, as reported in their paper, it is difficult to infer the accurate user location. A radius of 10 miles is therefore used to represent user's location, which is too far to help in identifying POIs in dense regions. Kinsella et al. [14] propose to identify spatial objects for each tweet at different granularity levels, from country to zip code. They model each spatial object using a language model and then compare with the language model of the tweet to locate the most probable spatial object. Li et al. [16] compute a coarse-grained user location at city level and aim to disambiguate POIs with the same name but in different cities. Li et al. [18] propose to combine the visual features and textual features to infer the POI for an Instagram photo. This type of studies assumes that tweets are not geo-tagged, and aim at inferring the coordinate locations of tweets. In contrast, we consider geo-tagged tweets and use them to annotate POIs. In fact, these proposals are complementary to our work for tweets without geo-tags.

User Behavior Modeling: User behavior is an important factor in associating POIs with geo-tagged tweets. If we know the preferred regions of a user and the popular POIs in those regions, we can probably guess on which POI the user posted a tweet. Many efforts have been done in modeling user behaviors in geographical data [11,12,15,20,25,26,28,29]. Hong et al. [11] propose a generative model to analyze the geographical topics in geo-tagged tweets. Kurashima et al. [15] and Hu et al. [12] study the user preference on latent regions. Yuan et al. [28] explore personalized regions for individuals and incorporate temporal information in modeling user behaviors. Yin et al. [25,26] consider user interests over time. Most of the existing proposals for user behavior modeling use Foursquare check-ins data, and they assume that each post is related to POIs. Our problem is different in that we have both POI related and unrelated tweets, and thus the existing user behaviour models are not applicable to our problem because they always associate a tweet with a POI. Moreover, we propose a supervised model while the existing methods are unsupervised. Our supervised model introduces a relatedness response and a set of dummy POIs to better fit the POI annotation problem, which have not been explored in existing models. The detailed difference between our supervised model to existing methods could be found in Section 4.1.

Other Related Work: Many pieces of research have been done in finding the location of a Twitter user [2, 5, 7]. Amitay et al. [2] use heuristic rules to infer user locations. Cheng et al. [7] and Chandra et al. [5] propose probabilistic models to estimate the city level location of a user. Identifying POIs for tweets is based on the limited information of a single tweet, while identifying the location for a user can be based on all her tweets.

3. ANNOTATING POIS WITH TWEETS

We formulate our problem in Section 3.1. Then, we discuss two baseline solutions in Section 3.2.

3.1 Problem Formulation

Suppose we have a collection of historical records of tweets associated with their POIs, denoted by $D = \{d_1, d_2, ..., d_{|D|}\}$. Each historical record d is represented as a 4-tuple $\langle u_d, \widetilde{\mathbf{cd}}_d, \mathbf{w}_d, l_d \rangle$, where u_d, $\widetilde{\mathbf{cd}}_d$, $\mathbf{w}_d$ are the user, coordinates, and the set of words of tweet d, respectively, and l_d represents the POI where the tweet was posted or associated with. Attribute l_d is *NULL* if the tweet is not related to any POI. We also have a collection of POIs $L = l_1, l_2, ..., l_{|L|}$ and each POI l is represented as a pair $\langle \widetilde{\mathbf{cd}}_l, \mathbf{t}_l \rangle$, where $\widetilde{\mathbf{cd}}_l$, $\mathbf{t}_l$ are the coordinates and the text context of the POI. The text context contains the name of the POI as well as the tips of the POI[1]. Given a new tweet $\langle u_d, \widetilde{\mathbf{cd}}_d, \mathbf{w}_d \rangle$, the POI annotation problem is to *EITHER* return the top-1 POI that is relevant to the tweet if it is POI-related, *OR* return no POI if it is not-POI-related.

3.2 Basic Solutions

This is a new research problem and there is no existing solutions. Since we have both POI related and unrelated tweets, directly annotating the top-ranked POI to each tweet only according to distance or text similarity performs bad in our experiments. Hence, we propose two basic solutions based on both spatial and textual information. The first one is a ranking based model which combines a spatial filter and a ranking component based on language models. The second one solves the problem in two steps: First, we classify the tweets into two classes, namely POI-related and non-POI-related; Second, a ranking model is used to map a POI-related tweet to a POI based on both the language model and spatial distance. The two models are named as *RANK* and *CLASS*, respectively.

RANK: Intuitively, if a tweet is posted at a POI, the coordinates of the tweet and the POI should be close to each other. Note that their coordinates are seldom the same because of the accuracy of GPS devices and the spatial region of a POI (which may not be a single coordinate point). The idea of RANK is to apply a spatial filter to restrict the search area for a tweet to be associated, and then to map the tweet to a POI in the restricted area. Suppose we restrict the search space within a small range of nearby POIs (*e.g.,* 100 meters) centered at the tweet's coordinate. The next problem is to find out whether any POI in the search space is semantically close to the tweet. More specifically, let $L_{d,m}$ be the set of POIs within m meters of tweet d, and $P(\mathbf{w}|l)$, $P(\mathbf{w}|d)$ be the language models of POI l and tweet d, respectively. The language model for a POI l is computed by counting the words in the text context of the POI (*i.e.,* $\mathbf{t}_l$), and the words in the historical tweets posted in the POI *i.e.,* $\cup\{\mathbf{w}_d | l_d = l\}$.

RANK associates tweet d to a POI $l \in L_{d,m}$ that achieves the highest likelihood of the tweet as follows:

$$\underset{l \in L_{d,m}}{\operatorname{argmax}} \prod_{w \in \mathbf{w}_d} P(w|l).$$

Note that, this ranking mechanism will associate a geo-tagged tweet to a POI even though it is non-POI-related. To solve this problem, we add a dummy POI which stands for "Non-POI", and build a location independent language model for it using all non-POI-related tweets in the training set. We rank all the POIs together with the dummy POI and pick the top-1 result. If the dummy POI is the top-1 result, then the tweet is considered to be non-POI-related.

CLASS: This method first classifies the tweets into POI-related or non-POI-related, and then maps POI-related tweets to their corresponding POIs based on the combination of a distance model and a language model. To classify tweets into the two classes, we use the words in the training tweets as features to learn a linear Support Vector Machine (SVM).[2]

If a tweet is POI-related, the next problem is to map the tweet to its POI. We develop a ranking model which comprises a distance model and a language model to identify the POI for POI-related tweet d. The distance model is used to constrain the POI to be close to the tweet. To this end, we use a zero-mean normal distribution to model the probability of observing POI l given tweet d in spatial within an error tolerance σ^2, *i.e.,* $dist(d, l) \sim \mathcal{N}(0, \sigma^2)$. The variance σ^2 can be learnt on the history tweets using the maximum likelihood principle. The language model is built in the same way as we do for RANK. The top-1 ranked POI for tweet d is based on the overall ranking score computed by:

$$Score(d, l) \propto exp\{-\frac{dist(d, l)^2}{2\sigma^2}\} \times \prod_{w \in \mathbf{w}_d} P(w|l).$$

Expanded CLASS: We now expand the CLASS model by incorporating region information in the classification. This model is named CLASS-R. Specifically, we first use k-means clustering to group the training tweets to R regions and assign the closest region to each tweet. Then, we compute the probability of posting POI related tweets $p(d \in D_+|r)$ in each region r, and the probability for a user u to posting tweets in each region $p(r|u)$, where D_+ is the set of POI-related tweets. We use the probability $p(d \in D_+|u) = \sum p(d \in D_+|r)p(r|d)$ as region features, and thus we have R region features for each tweet. For training tweets, we set $p(r|d) = 1$ if r is assigned to tweet d; otherwise we set it at 0. For test tweets, we set $p(r|d) = p(r|u_d)$ for each region to smooth the region features using the user's interests to regions.

For the ranker, we use two alternative ranking models. One of them is a learning-to-rank approach proposed by Lian et al. [19], and the other is a geographical topic model proposed by Yuan et al. [28]. In the learning-to-rank method, the following features are used: 1) number of check-ins at a POI; 2) check-in time of the POI; 3) the check-ins of a user at the POI; 4) the distance between the POI and the tweet; and 5) text similarity between the POI and the tweet. We name the learning-to-rank method as CLASS-LR and the geographical topic model as CLASS-W4, based on the classification results of CLASS. We also name the two ranking methods based on the results of CLASS-R as CLASS-R-LR and CLASS-R-W4, respectively.

4. SUPERVISED BAYESIAN MODEL

We now discuss the motivations of building a supervised Bayesian model for the proposed problem and present the generative story of our model. Then, we present the method for estimating model parameters. Our model is further enhanced by incorporating external textual content of the locations, *i.e.,* Foursquare tips.

4.1 Motivation & Novelty

Both baselines presented in Section 3.2 cannot capture the relationships among variables. Next, we illustrate two example relationships and more relationships are presented as Intuitions in Section 4.2. For instance, there should be relations between mapping a tweet to a POI and determining its POI-relatedness. If we know that a tweet was probably posted at a bakery, then by comparing the location and language models of the tweet and the bakery, it is easier to tell whether the tweet is POI-related. There also should be relations among user, region and POI-relatedness. Some regions (*e.g.,*

[1]The name and tips are collected from Foursquare in our work.

[2]We have also tried other classifiers and SVM performed the best in our experiments.

Times Square), may have more POI-related tweets than other regions. If a user often visits POIs at Times Square, the tweets posted by the user at Times Square are more likely to be POI-related.

To better capture the relations among variables (e.g., text, region, user, location, and POI-relatedness), we develop a novel supervised Bayesian model. Compared with the existing geographical topic models [11, 12, 15, 28, 29], the novelty of our model is two-fold. First, we add a *relatedness response* to each tweet in our supervised model to determine whether the tweet is POI-related by considering the current context. The context includes 1) the number of words that are not related to any POI; 2) the distance from the tweet to its most semantically related POI; and 3) the latent region in which the tweet was posted. By jointly modeling words and latent regions with the relatedness response, we are able to find better latent regions and language models for solving the relatedness problem. Second, we introduce a set of dummy POIs to model the location of tweets that are not related to any POI. The dummy POIs benefit the model in that we can model the two types of tweets in a consistent way because the non-POI-related tweets can now be assigned to a "dummy POI". Moreover, each dummy POI has its own language model and coordinates like real POI, capturing semantic and geographical information. Such information makes it possible to analyze the language models and popularity (in regions) of dummy POIs. To the best of our knowledge, the existing *geographical topic models* are all unsupervised, and none of them exploits the two novel aspects.

Our model works as follows. When judging whether a tweet is POI-related, the model first estimates the most probable POI (including dummy POIs), and then computes the relatedness response. If the tweet is POI-related, the candidate POIs are ranked based on their textual and spatial information, and the joint probability of the tweet being POI-related and posted at each POI.

Compared with supervised LDA [4], our model associates several features in different scopes to the response including proportion of POI-related words in the scope of word tokens, geographical distance and latent region in the scope of document. We design a feature vector that contains different scopes of features for each tweet and the feature vector affects the training of both textual and geographical aspects of the model. Supervised LDA uses average topic assignments of a document as features for regression and does not consider different types of features.

4.2 Generative Story for Tweets

The supervised Bayesian model for annotating POIs with tweets are built according to the following intuitions.

Intuition 1: A user u may post geo-tagged tweets in some preferred latent regions, *e.g.,* shopping streets, sightseeing areas. Each region contains a set of nearby POIs. User interests in regions can be described by a categorical distribution $p(r|u)$.

Intuition 2: To visit a POI in a region, user would consider both its popularity and distance. For a region r, we use a popularity distribution $p(l|r)$ and a bivariate Gaussian distribution over coordinates $p(\widetilde{\mathbf{cd}}_l|r)$ to depict the two factors, respectively.

Intuition 3: A tweet may or may not be related to a POI. For POI-related tweets, the location and text are determined by the POI where they were posted. For daily conversational tweets that are not related to any POI, we can imagine that their location and text are determined by some dummy POIs that do not exist in the real world. We introduce a set of dummy POIs L' to model such tweets.

Intuition 4: In a POI-related tweet, users tend to use words related to the POI other than general words. That is, if the tweet is POI-related, the user may select words from both the word distribution of the POI $p(w|\eta_l)$ and background word (*e.g.,* I, like) distribution $p(w|\eta_0)$. If the tweet is non-POI-related, the user may select words from the word distribution of a dummy POI and the background word distribution.

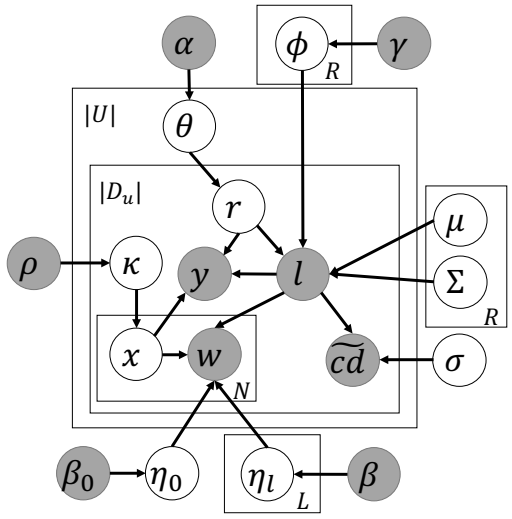


Figure 1: Supervised Bayesian Model for POI Annotation

Intuition 5: If a tweet is POI-related, it is likely that the tweet is close to the POI where it was posted, and the text of the tweet is related to its POI. Moreover, if a tweet is posted in a region that has many attractions, *e.g.,* sight seeing spots, restaurants, etc., it would be more likely to be POI-related other than tweets posted in regions that have fewer attractions. We use $\bar{x}_d$ to describe the portion of words that are related to the tweet's POI in tweet d, and $\texttt{dist}(d, l_d)$ to denote the surface distance between tweet d and its POI l_d. For each region, we use $\bar{c}_r$, which is the average count of POI-related tweets in region r against non-POI-related tweets to denote the probability of posting POI-related tweets in that region. Let R be the total number of latent regions, we have $R+2$ features to determine whether a tweet is POI-related or not.

With the above intuitions, we proceed to describe the generative process of the proposed sBM model. For convenience, we show the graphical representation of the model in Figure 1 and summarize the notations in Table 1.

To generate a geo-tagged tweet, a user first visits a region r according to a multinomial distribution over user's interests $Multi(\theta_u)$. Then, the user randomly chooses a POI (which can be a dummy POI) to visit according to 1) a multinomial distribution $Multi(\phi_r)$ over its popularity in region r; and 2) a bi-variate Gaussian distribution $\mathcal{N}(\mu_r, \Sigma_r)$. When the user compose the tweet, she would choose words either from the language model η_l of the POI (including dummy POIs) or from a background language model η_0. We introduce a latent variable x for each word token in a tweet to identify from which language model the word is selected from. More precisely, if $x = 1$, the user selects the word from the language model of the POI (including dummy POIs). If $x = 0$, the user selects the word from the background language model. The coordinates of the tweet are generated according to a Gaussian distribution $\mathcal{N}(\widetilde{\mathbf{cd}}_l, \sigma_l^2)$ with error σ_l^2.

Now we come to the supervised part of our model. The label y, indicating whether a tweet is POI-related, comes from a linear regression model. The covariates (features) of the regression model

Table 1: Summary of notations

Notation	Description
D_u	the set of tweets posted by user u
θ_u	the region interests of user u
r_d	latent region of tweet d
ϕ_r	the POI popularity in region r
$\widetilde{\mathbf{cd}}_l$, σ_l^2	the location and variance in coordinates system of POI l
μ_r, Σ_r	the mean and variance in coordinates system of region r
η_0	background word distribution
η_l	word distribution for POI l
$x_{(d,n)}$	the switch of selecting word (d,n) from either background or POI language models
$\widetilde{\mathbf{cd}}_d$	the coordinates of tweet d
κ_d	distribution of switch for document d
ω	the weights for the regression of POI-relatedness
σ^2	the variance of POI-relatedness

comprise the portion of POI-related words $\bar{x}_d$, the distance between the tweet and the POI $\texttt{dist}(d, l_d)$, and the popularity of real POIs in the region $\bar{c}_r$. These covariates comprise vector $\bar{z}_d$, which is a $R+3$ dimensional vector that comprises a constant 1 and the $R+2$ features, *i.e.,* $\bar{z}_d = \langle 1, \bar{x}_d, \texttt{dist}(d, l_d), \bar{c}_1, \ldots, \bar{c}_R \rangle$. The regression coefficients on these covariates comprises vector ω, which is a $R+3$ dimensional vector containing the weights to the $R+2$ features, *i.e.,* $\omega_1, ..., \omega_{R+2}$ and the bias ω_0. We generate a label y according to a normal distribution $\mathcal{N}(\omega^T \bar{z}_d, \sigma^2)$ with variance σ^2. The generative process are summarized as follows.

- For each user u,
  - Draw a region preference distribution $\theta_u \sim Dir(\alpha)$
- For each region r,
  - Draw a POI distribution $\phi_r \sim Dir(\gamma)$
- For each POI l including dummy POI,
  - Draw a word distribution $\eta_l \sim Dir(\beta)$
- Draw a background word distribution $\eta_0 \sim Dir(\beta_0)$
- For each tweet d posted by user u,
  - Draw a switch distribution $\kappa_d \sim Beta(\rho)$
  - Draw a region $r_d \sim Multi(\theta_u)$
  - Draw a POI $l_d \sim Multi(\phi_r) \times \mathcal{N}(\widetilde{\mathbf{cd}}_l | \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r)$
  - Draw coordinates $\widetilde{\mathbf{cd}}_d \sim \mathcal{N}(\widetilde{\mathbf{cd}}_{l_d}, \sigma_l^2)$
  - For each word position n in d,
    - Draw a switch $x_{(d,n)} \sim Binomial(\kappa_d)$
    - If $x = 1$, draw a word $w_{(d,n)} \sim Multi(\eta_{l_d})$
    - If $x = 0$, draw a word $w_{(d,n)} \sim Multi(\eta_0)$.
  - Draw $y \sim \mathcal{N}(\omega^T \bar{z}_d, \sigma^2)$

4.3 Parameter Inference

An open problem is how to determine the locations of dummy POIs in order to generate the coordinates of non-POI-related tweets from a Gaussian distribution ($\mathcal{N}(\widetilde{\mathbf{cd}}_{l_d}, \sigma_l^2)$). It is possible to learn the locations of dummy POIs together with the other parameters in the model. However, by considering that 1) it could be time-consuming when the number of dummy POIs becomes large and 2) the locations of dummy POIs do not affect the model much because they have no exact geographical meanings, we adopt an approximate strategy to compute the Gaussian distribution for dummy POIs. Suppose we need to generate a set of dummy POIs with size $|L'|$, our goal is to divide the non-POI-related tweets into $|L'|$ subsets, such that the tweets are equally distributed in these subsets, and the POIs in each subset are geographically close to each other. This is because we want to avoid the extreme case that a large number of tweets are assigned to very few dummy POIs. Specifically, we iteratively divide the coordinates space into four equal-sized cells and build a Quadtree [21]. In each iteration, we divide the cell with largest number of tweets in the Quadtree. The dividing process stops when we have more than $|L'|$ leaf cells each contains a reasonable number of tweets (*e.g.,* at least 100 tweets). Then, we pick the top $|L'|$ leaf cells with largest number of tweets as dummy POIs. The coordinates of a dummy POI is computed by averaging the coordinates of its tweets. Finally, the tweets that are not located in any of the top $|L'|$ cells are assigned their closet dummy POIs. The error σ_l of each POI including dummy POIs is then computed by regression:

$$\sigma_l^2 = \frac{1}{|D_l| - 1} \sum_{d \in D_l} ||\widetilde{\mathbf{cd}}_d - \widetilde{\mathbf{cd}}_l||^2, \tag{1}$$

where D_l is the set of tweets assigned to POI l.

With the known Gaussian distributions of POIs and dummy POIs, the inference problem becomes to compute the other parameters by maximizing the corpus level likelihood. The likelihood of generating the corpus D using our model with the set of parameters Φ is computed by:

$$\begin{aligned} P(D|\Phi) &= P(y_d|\bar{z}_d) \prod_d P(r_d|u) P(l_d|r_d) P(\widetilde{\mathbf{cd}}_l|r_d) P(\widetilde{\mathbf{cd}}_d|l_d) \\ &\times \prod_d \prod_n P(w_{d,n}|x_{d,n}, \eta_0, \eta_{l_d}) P(x_{d,n}|\kappa_d), \end{aligned} \tag{2}$$

$$P(w_{d,n}|x_{d,n}, \eta_0, \eta_{l_d}) = \begin{cases} P(w_{d,n}|\eta_0) & \text{if } x_{d,n} = 0 \\ P(w_{d,n}|\eta_{l_d}) & \text{if } x_{d,n} = 1. \end{cases} \tag{3}$$

Estimating the parameters by maximizing the likelihood is intractable. We therefore develop a two-step learning algorithm combining Gibbs sampling and Expectation-Maximization algorithm to estimate the parameters.

Expectation: We approximate the posterior distribution of latent variables x and r given other variables using collapsed Gibbs sampling. With each x and r known, the Gaussian parameters μ_r, Σ_r and the regression parameters ω and σ^2 are updated in the maximization step. The Gibbs samplers for latent variable x and r are derived as in Eq. (4) and Eq. (5) (Equations are listed on the next page).

The variable $c_{l,v}^{x,-(d,n)}$ in Eq. (4) is the count of assigning x to the word v in the tweets posted in POI l, excluding the assignment for the n-th word in document d. We use $*$ to denote *ANY*. For example, $c_{*,v}^{x,-(d,n)}$ stands for the count of assigning x to the word v in the tweets posted in *ANY* POI, excluding the n-th word in document d. Similarly, the variable $c_{u,l}^{r,-d}$ in Eq. (5) is the count of assigning r to the tweets that posted by user u at POI l, excluding the assignment in document d.

Maximization: We update the Gaussian parameters μ_r, Σ_r and the regression parameters ω, σ^2 by maximizing the likelihood given the samples of x and r. Let D_r be the set of tweets that are assigned to region r, and $\widetilde{\mathbf{cd}}_{l_d}$ be the coordinates of the POI l_d associated with tweet d. The update functions of the Gaussian parameters are

$$P(x_{(d,n)}=0|X_{-(d,n)},D) \propto (c^{0,-(d,n)}_{*,*,d}+\alpha)\frac{c^{0,-(d,n)}_{*,w_{d,n},*}+\beta_{w_{d,n}}}{\sum_v c^{0,-(d,n)}_{*,v,*}+\beta_v}\times\frac{1}{\sqrt{2\pi\sigma^2}}exp\{-\frac{(y_d-\omega^T\bar{z}_d)^2}{2\sigma^2}\}$$

$$P(x_{(d,n)}=1|X_{-(d,n)},D) \propto (c^{1,-(d,n)}_{*,*,d}+\alpha)\frac{c^{1,-(d,n)}_{l_d,w_{d,n},*}+\beta_{w_{d,n}}}{\sum_v c^{1,-(d,n)}_{l_d,v,*}+\beta_v}\times\frac{1}{\sqrt{2\pi\sigma^2}}exp\{-\frac{(y_d-\omega^T\bar{z}_d)^2}{2\sigma^2}\} \quad (4)$$

$$P(r_d=r|R_{-d},D) \propto (c^{r,-d}_{u_d,*}+\alpha)\frac{c^{r,-d}_{*,l_d}+\gamma_{l_d}}{\sum_l c^{r,-d}_{*,l}+\gamma_l}\times|\Sigma_r|^{-\frac{1}{2}}\exp\{(\widetilde{\mathbf{cd}}_d-\mu_r)^T\Sigma_r^{-1}(\widetilde{\mathbf{cd}}_d-\mu_r)\} \quad (5)$$

$$P(l_d=l|u_d,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d)\propto P(l,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d|u_d)=P(\widetilde{\mathbf{cd}}_d|l)\sum_r\phi_{r,l}\theta_{u_d,r}P(\widetilde{\mathbf{cd}}_l|r)\prod_n\kappa_d\eta_{0,w_{d,n}}+(1-\kappa_d)\eta_{l,w_{d,n}}. \quad (6)$$

computed as in Eq. (7) and Eq. (8).

$$\mu_r = \frac{1}{|D_r|}\sum_{d\in D_r}\widetilde{\mathbf{cd}}_{l_d} \quad (7)$$

$$\Sigma_r = \frac{1}{|D_r|}\sum_{d\in D_r}(\widetilde{\mathbf{cd}}_{l_d}-\mu_r)(\widetilde{\mathbf{cd}}_{l_d}-\mu_r)^T. \quad (8)$$

We update the regression parameters by computing $\bar{z}_d$ for each tweet d using the samples of POI-relatedness indicator x for each word, and region r. Specifically, we set $\bar{x}_d$ by computing the proportion of words that are POI-related. For the region features, since we know the region of a tweet is r_d by sampling, we set the value of the r_d-th region feature in $\bar{z}_d$ to $\bar{c}_{r_d}$, the proportion of POI-related tweets in region r_d, and set the values of other regions to 0. Let $\bar{Z}$ be the a $|D|\times(R+3)$ matrix in which each row is $\bar{z}_d$ for a tweet d, the regression parameters are updated in Eq. (9) and Eq. (10).

$$\omega \leftarrow (\bar{Z}^T\bar{Z})^{-1}\bar{Z}^T y \quad (9)$$

$$\sigma^2 \leftarrow \frac{1}{|D|}(y^Ty-y^T\bar{Z}(\bar{Z}^T\bar{Z})^{-1}\bar{Z}^Ty) \quad (10)$$

The expectation step and maximization step are repeated until the likelihood converges. The latent variables can be efficiently sampled. Sampling x and r only needs to scan the tweet set D once and thus the complexity of sampling is $O(|W|+R|D|)$, where W is all word tokens in D and R is the number of regions. The update of parameters μ_r and σ_r has a summation on D and thus the complexity is $O(R|D|)$. For the update of regression parameters ω and σ^2, the most time consumption part is the matrix multiplication of $\bar{Z}^T\bar{Z}$ and its inversion. The complexity of updating the regression parameters is $O(R^2|D|+R^3)$. Suppose we run I iterations for the estimation process, the overall complexity of the learning process is $O(I(|W|+R^2|D|+R^3))$.

4.4 Prediction for New Tweets

After learning the parameters, the POI of a tweet is predicted by the model in two steps. Given a new tweet $d=\langle u_d,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d\rangle$, the POI-relatedness y_d is first predicted and then the POIs are ranked according to the joint probability $P(y_d=1,l_d=l|u_d,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d)$.

POI-Relatedness: In this step, we first compute the probability $P(l_d=l|u_d,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d)$ as in Eq. (6) to guess the most probable POI (including dummy POIs) for the tweet. The corresponding $y_d=\omega^T\bar{z}_d$ is then computed to determine whether the tweet is POI-related. If $y_>0$, we classify the tweet as POI-related.

In Eq. (6), the estimation of variable κ_d is intractable. A collapsed Gibbs sampler similar to Eq. (4) is used here without involving the POI-relatedness label y. Specifically, we infer the following Gibbs sampler to sample the indicator x_n for each word $w_{d,n}$:

$$\begin{aligned}P(x_{(d,n)}=0|X^{new}_{-(d,n)},\eta_0)&\propto(c^{0,-(d,n)}_{*,*,d}+\alpha)\times\eta_{0,w_{d,n}}\\P(x_{(d,n)}=1|X^{new}_{-(d,n)},\eta_l)&\propto(c^{1,-(d,n)}_{*,*,d}+\alpha)\times\eta_{l,w_{d,n}},\end{aligned} \quad (11)$$

where $X^{new}_{-(d,n)}$ is the set of assignments of x for word tokens in the new document, and η are the language models learnt from the training process. Other variables are defined in the same way as in the training process. After sampling x for each word, we compute κ_d as $\frac{\sum_{n=0}^{N_d}I(x_n=1)+\rho}{N_d+2\rho}$, where $I(\cdot)$ is an indicator function, N_d is the total number of words in tweet d, and ρ is used for smoothing.

POI Ranking: When a tweet is POI-related or $y_d>0$, we compute the most probable real POI for the tweet by considering two factors. One is the posterior probability of the POI given in Eq. (6), the other is the probability of observing the POI-relatedness label $y_d=1$ given the POI l. In other words, we rank all the POIs in the descending order of the joint probability $P(y_d=1,l_d=l|u_d,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d)=P(y_d=1|\bar{z}_{d,l})P(l|u_d,\widetilde{\mathbf{cd}}_d,\mathbf{w}_d)$ and then return the top-1 ranked POI for the given tweet. The vector $\bar{z}_{d,l}$ is the feature vector when we assign POI l to tweet d.

4.5 Incorporating Tips

Because a tweet has a limit of 140 characters, and if a POI has very few posted tweets, the language model built for the POI would be extremely sparse. With the sparse language model, it is difficult to correctly judge whether a tweet is related to the POIs. However, we can make use of the external context for POIs. Example external context could be Foursquare tips, Yelp reviews, etc. In this paper, we focus on making use of Foursquare tips, but the model can accommodate other text resources.

Suppose we are given a set of tips D_t for all POIs. Each tip is represented as a pair of a POI l and a set of words $\mathbf{w}$, *i.e.,* $\langle l,\mathbf{w}\rangle$, where l is the POI of the tip, and $\mathbf{w}$ is the words of the tip. By observing that the proportion of background words used in tips and tweets are different, we use different distributions of drawing a switch variable (κ) for the two sources. We build a generative model with switch variable κ for the tips, and then use the learnt language model for each POI and background language model as prior for the language models in the supervised Bayesian model. Specifically, for each word in a tip, we first draw a switch variable x, then we draw the word from the language model of the POI if $x=1$, otherwise we draw the word from the background language model. We apply Gibbs sampling to learn the language models from tips. Suppose the count of assigning a word v to a POI l in tips is $c^{tips}_{l,v}$, we substitute the new prior $\beta'_{lv}=c^{tips}_{l,v}+\beta_v$ in Eq. (4) to learn the model presented in Section 4.2.

5. EXPERIMENTS

We show the effectiveness of the proposed supervised Bayesian model (sBM) by experimenting on two real world datasets. We first discuss the experimental setup, including the dataset preparation and performance measures. Then we set the parameters, *e.g.,* the number of dummy POIs and the number of latent regions, by empirically studying on a validation dataset. Finally, we compare the performance of sBM with baseline methods, and conduct an empirical study on the regions learnt by the model.

5.1 Experimental Setup

Dataset: We collect English tweets for experiments using Twitter API[3] in two cities: New York city (NYC) and Singapore (SG). Specifically, a random sample of geo-tagged tweets posted by 2,393 users in New York city were collected from September 2010 to January 2015. For Singapore, a random sample of geo-tagged tweets from 9,978 users were collected from March 2014 to August 2014. The POIs from both cities together with their tips were collected using Foursquare API. In the rest of the paper, we use NYC and SG for the abbreviations of the two datasets for simplicity.

From each of the two cities, we randomly selected some tweets and asked three annotators to annotate whether a tweet is POI-related. The groundtruth label is based on majority voting from the three annotators. Another annotator is then engaged to associate the POI-related tweets to their nearby POIs. Finally, we obtained 4,827 and 5,827 geo-tagged tweets with groundtruth labels for model validation and evaluation, for the two cities, respectively.

In order to acquire a much larger dataset for training the models, we approximate the groundtruth for training using the following rule-based strategy. When a user checks in a POI in Foursquare, there is an option to share the check-in on Twitter as a check-in tweet. Similarly, Instagram also allows users to share their posts on Twitter. We consider as positive samples the Foursquare check-in tweets, and the tweets shared through Instagram and are associated with POIs. The remaining tweets that do not satisfy these conditions are considered as negative samples. This treatment results in more than 210 thousands training tweets for the NYC dataset, and 380 thousands of training tweets for the SG dataset. The detailed statistics of the two datasets are reported in Table 2.

Evaluation Measures: To evaluate the performance of our models, we adopt the measures that are similar to precision and recall. The difference is that we have two levels of predictions here, *i.e.,* whether a tweet is POI-related, and whether a POI-related tweet is correctly associated with its groundtruth POI. Only when the groundtruth POI is correctly predicted for a POI-related tweet, the prediction is considered to be true positive.

Let TP be the number of true positive, TP_R be the number of times we classify a tweet as POI-related, FP be the number of times we incorrectly associate any POI to a non-POI-related tweet, FN be the number of times we predict $NULL$ for a POI-related tweet, and TN be the number of times we predict $NULL$ for a non-POI-related tweet.

The precision and recall are computed as:

$$\text{Precision} = \frac{TP}{TP_R + FP}, \quad \text{Recall} = \frac{TP}{TP_R + FN}.$$

We can also evaluate a model for its ability of judging whether a tweet is POI-related or not, using similar measures. To distinguish from the above precision and recall, we call them *relatedness precision* and *relatedness recall* (or R-Precision and R-Recall for short).

[3] https://dev.twitter.com/rest/public

These two measures are computed as:

$$\text{R-Precision} = \frac{TP_R}{TP_R + FP}, \quad \text{R-Recall} = \frac{TP_R}{TP_R + FN}.$$

Since there is a trade-off between precision and recall, we also compute the F1 score for the two sets of measures, *i.e.,*

$$\text{F1} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}.$$

We use F1 and R-F1 to denote the F1 scores computed from the two sets of Precision and Recall, respectively.

Methods to Compare: We compare our model (sBM) with the two baseline methods (*i.e.,* RANK and CLASS) discussed in Section 3.2. We also implemented the learning-to-rank model and the geographical topic model W4 for comparison. In total, 9 methods are evaluated in our experiments, summarized in Table 3.

The parameters of all methods are set at the best value in terms of F1 score on the validation set. For RANK, we tried the distance threshold from 50, 100, 200 and 500 meters, and then select 100 meters, which performed best in terms of F1 score on both datasets. Smaller threshold (e.g., 50 meters) results in low recall because the related POI may be filtered out, while larger value (e.g., 200 meters) introduces more POIs to rank and thus may decrease the precision. For W4, the number of topics are set at 100 and the number of personalized regions is set at 2. The classifiers in both CLASS and CLASS-R are implemented using Liblinear [10]. The learning-to-rank model was implemented using SVM^{rank} [13].

5.2 Parameter Selection

We have two parameters for tuning in the supervised Bayesian model. One is the number of dummy POIs, and the other is the number of latent regions. We first investigate the effect of dummy POIs by fixing the number of latent regions. Then we fix the number of dummy POIs and select the number of latent regions.

Setting the Number of Dummy POIs: We fix the number of regions to 50. Figure 2 plots the precision, recall and F1 by varying the number of dummy POIs from 10 to 10,000 on the validation sets of the two cities, NYC and SG. Note that, the x-axis is in log-scale. Observe from the figures, the varying of the number of dummy POIs does not affect the F1 score much. On the NYC dataset, the performance slightly increases as the number of dummy POIs increases. On the SG dataset, when the number of dummy POIs increases to 10,000, precision increases significantly, with degradation in recall values. The increase in the number of dummy POIs leads to decrease of the distance from an arbitrary tweet to a dummy POI. As the result, the probability for assigning a tweet to a dummy POI increases, thus leading to the increase in the probability of classifying a tweet as non-POI-related.

Considering the changes in precision, recall, and F1, we set the number of dummy POIs to 10,000 on both datasets.

Setting the Number of Regions: With the number of dummy POIs fixed in both datasets, we investigate the effect of the number of latent regions. The precision, recall and F1 by setting different numbers of regions are reported in Figure 3 on both datasets.

On the SG dataset, the precision increases and reaches its best value when the number of regions increases to 30. The precision becomes stable for larger number of regions. On the other hand, the recall decreases slightly as the number of regions increases. The precision increases because each region captures more detailed information, *i.e.,* user interests and language models. The recall decreases because it is more likely to classify a tweet to non-POI-related because of the increasing number of specific regions.

Table 2: Statistics of the two datasets

	New York city			Singapore		
# users			2,393			9,978
# POIs			482,480			321,985
# tweets (train validation test)	212,954	2,429	2,398	385,270	2,926	2,901
# POI-related tweets (train validation test)	43,010	474	559	29,290	221	289
# tips per POI			2.38			1.40
Vocabulary size			332,369			111,415

Table 3: Summary of the 9 methods evaluated in our experiments

Method	Description	Features used	
		User Preference	Latent Regions
RANK	RANK model with distance threshold = 100 meters	-	-
CLASS	CLASS model	-	-
CLASS-R	CLASS model enhanced by regions	-	✓
CLASS-W4	CLASS model + geographical topic model W4	✓	✓
CLASS-R-W4	CLASS-R model + geographical topic model W4	✓	✓
CLASS-LR	CLASS model + learning-to-rank	✓	-
CLASS-R-LR	CLASS-R model + learning-to-rank	✓	✓
sBM-T	supervised Bayesian model without using tips	✓	✓
sBM	supervised Bayesian model	✓	✓

On the NYC dataset, both precision and recall are poor when the number of regions is fewer than 30. One possible reason is that the POI-related tweets are distributed in many regions. When the number of regions is small, the regions learnt are too large to capture local information. When the number of regions is larger than 30, the model gets stable.

Based on the observations made from Figure 3, we set the number of regions at 30 on both datasets.

5.3 Overall Performance

We now compare the overall performance of the 9 methods listed in Table 3. For sBM, all the parameters are set as described in Section 5.1. The number of regions in CLASS-R is set to the same number as the supervised Bayesian model. We train sBM on a single machine with Intel Xeon E5-1620 CPU and 16 GB RAM and it takes 21 minutes and 42 minutes for training on NYC and SG datasets, respectively. We run the 9 methods on the test set and evaluate the results using precision, recall, and F1 score. The performance of all methods are reported in Figure 4.

As shown in Figure 4(c), sBM-T and sBM significantly outperforms the other models. Specifically, sBM improves the F1 score of the second best solution CLASS-LR by 28.3% on the NYC data and outperforms the second best solution RANK by 20.4% on the SG data. Moreover, sBM consistently performs better than sBM-T on both datasets because it considers tips as external information.

RANK achieves considerable precision on the NYC dataset because the distance threshold reduces the search space. However, the recall is low because the threshold also excludes the correct POI in many cases. RANK achieves considerable recall but low precision on the SG dataset as the POIs are much denser in Singapore. This result suggests that a simple language model cannot distinguish the similar POIs within the search space.

CLASS performs worse than RANK in most cases because the distance model in CLASS is less effective in eliminating the POIs that are not related to the tweet. CLASS-R performs better on the SG dataset but worse on the NYC data. It has low recall because the regions in New York city are more diverse than those in Singapore, and the regions learnt by simple clustering algorithm do not well fit the POI-relatedness problem.

CLASS-W4 and CLASS-R-W4 perform worst because 1) the W4 model does not model the relation between word and POI; and 2) the number of POI-related tweets takes a small portion of all tweets in our problem. The W4 model designed for POI-related tweets cannot leverage all the tweets to learn the user mobilities.

CLASS-LR and CLASS-R-LR both outperform their counterparts *i.e.,* CLASS and CLASS-R, respectively. CLASS-R-LR achieves even higher precision than sBM. One possible reason is that it uses more features to identify the correct POI for a tweet. The region features in CLASS-R and CLASS-R-LR are helpful to increase the precision, but result in lower recall compared to CLASS and CLASS-LR. This is because the region features are used to filter more POIs that may not be related to the tweet.

Our proposed supervised Bayesian model captures the relations between relatedness response and other variables, and user interests on regions. Thus, the model is more likely to infer the correct POI for POI-related tweets, and thus can achieve higher accuracy.

5.4 Performance of POI-Relatedness Problem

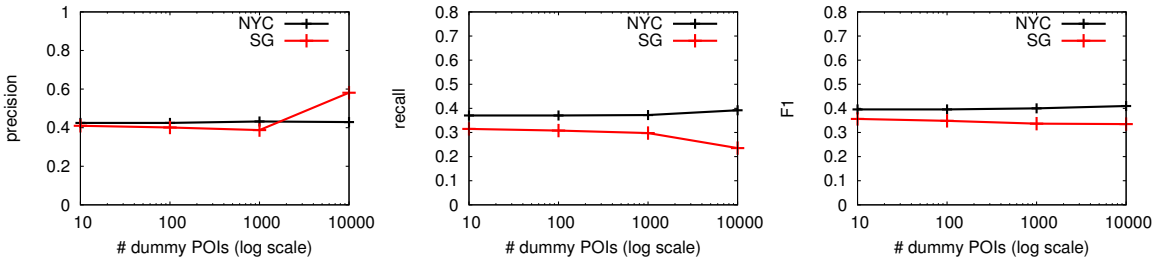
Because the learning-to-rank baselines use the same classifier as CLASS and CLASS-R for the POI-relatedness problem, we compare our model sBM with the other three methods RANK, CLASS, and CLASS-R. The results of the POI-relatedness problem are evaluated by R-Precision, R-Recall and R-F1, see Figure 5.

Observe from Figure 5, sBM outperforms CLASS by 29.1% on the NYC dataset and outperforms CLASS-R by 6% on the SG dataset. As discussed in the previous subsection, the CLASS-R performs worse on the NYC dataset because of the simple clustering algorithm used. Compared to CLASS, the regions even bring adverse impact on its performance. However, the method works well on the SG data because POI-related tweets in Singapore is relatively densely distributed.

Benefiting from the relatedness response and dummy POIs, sBM better fits latent regions for the relatedness problem. Because the relatedness response drives the distribution of POI-related tweets to be identical in different regions, *i.e.,* some regions could be more likely to have POI-related tweets while others tend to include tweets that are not related to POIs.

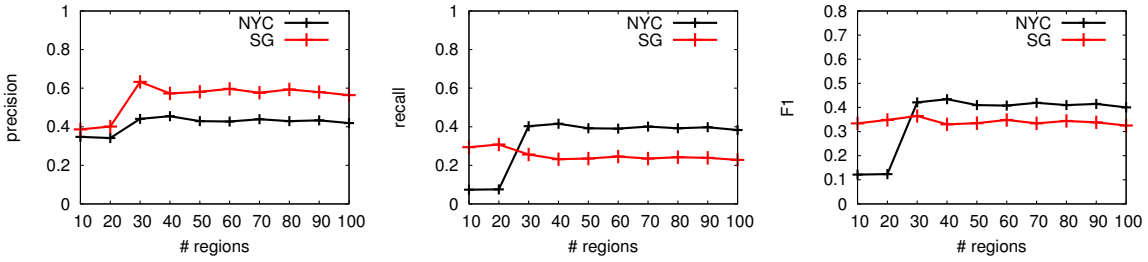
5.5 Empirical Study on Latent Regions

In this subsection, we compare the regions learnt by sBM and



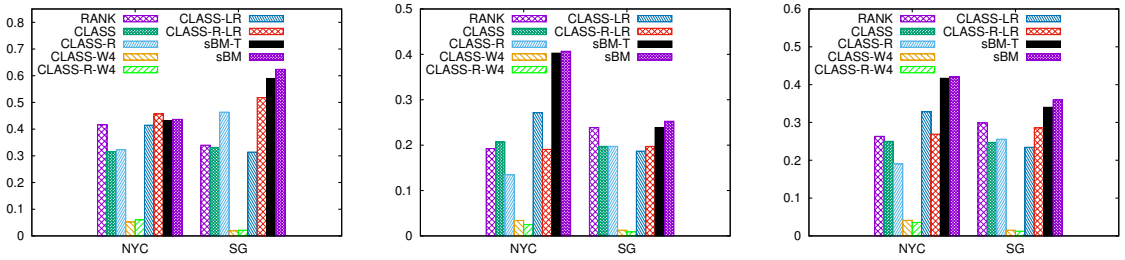
(a) Precision (b) Recall (c) F1

Figure 2: Effects of the number of dummy POIs (fixing the number of latent regions at 50)



(a) Precision (b) Recall (c) F1

Figure 3: Effects of the number of Latent Regions (fixing the number of dummy POIs to be 10,000)



(a) Precision (b) Recall (c) F1

Figure 4: Precision, Recall, and F1 of all the 9 methods, on the two datasets NYC and SG

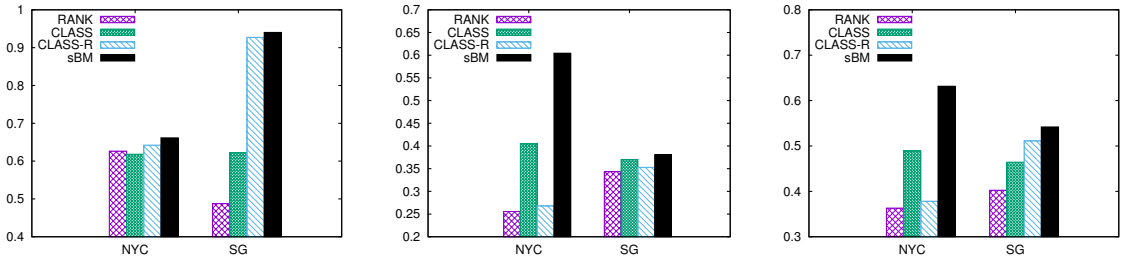
those learnt by k-means algorithm to demonstrate the effect of the relatedness response on regions. Specifically, we randomly pick two of the 30 regions learnt by our sBM model in each dataset. One of them has positive weight ω_{2+r} in the regression, while the other one has negative weight. Note that, positive weight indicates that the tweets posted in the region are more likely to be POI-related. For convenience, we call the former as *positive region* and the latter as *negative region*. To draw the boundary of a region on a map, we compute the contour line of its bivariate Gaussian distribution at confidence level 0.95. For k-means algorithm, we first assign each training tweet a cluster. Then, for each cluster, we use its containing tweets to estimate the bivariate Gaussian distribution for it. We select the two regions that are closest to the two regions learnt by sBM, respectively. The regions learnt by the two methods and the tweets in the two datasets are plotted in Figure 6.

In Figure 6, blue points indicate the non-POI-related tweets, and red points indicate the POI-related tweets. Positive regions are plotted in green color, while negative regions are plotted in yellow. In the example of New York city, sBM learns tighter positive region and the positive region has more POI-related tweets. On the other hand, sBM learns a negative region with fewer POI-related tweets. Similar observations are made on Singapore data. This is because the relatedness response affects the learning of latent regions. It learns regions that are fit to the POI relatedness problem.

6. CONCLUSION

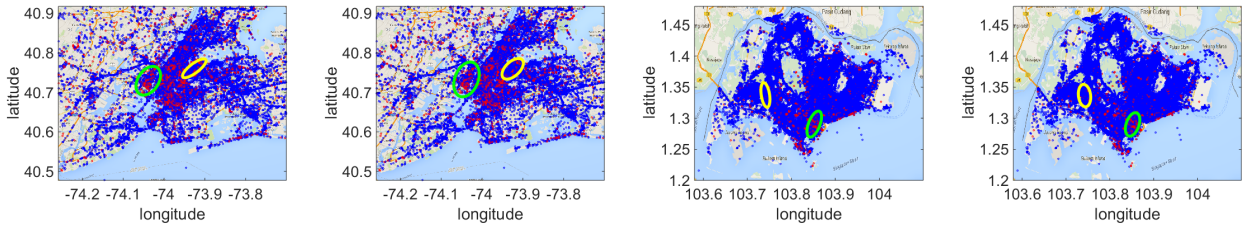
We study the problem of annotating POIs with geo-tagged tweets. By exploring the text, coordinates and the user behaviors, we start with two basic solutions and then propose a supervised Bayesian model. We further extend the model by integrating external text sources of POIs, *i.e.,* Foursquare tips, and show by experiment that our proposed supervised Bayesian model is effective for associating POI with geo-tagged tweets. The problem and solutions proposed in this paper benefit many applications, including user behaviors analysis, POI recommendations, geo-textual data stream publish/subscription, to name a few. This work opens a few interesting directions for future work. It would be interesting to investigate how to annotate POIs with tweets without geo-tagged coordinates. We also plan to explore how to efficiently annotate a large number of POIs over tweet stream.

Acknowledgment This work is supported in part by a Tier-1 grant awarded by Ministry of Education Singapore (RG22/15), and a grant awarded by Microsoft.



(a) R-Precision (b) R-Recall (c) R-F1

Figure 5: R-Precision, R-Recall, and R-F1 of the four methods for the POI-Relatedness problem, on the two datasets



(a) sBM - New York City (b) k-means - New York City (c) sBM - Singapore (d) k-means - Singapore

Figure 6: Regions in both Datasets (sBM v.s. k-means)

7. REFERENCES

[1] A. Ahmed, L. Hong, and A. J. Smola. Hierarchical geographical modeling of user locations from social media posts. In *WWW*, pages 25–36, 2013.

[2] E. Amitay, N. Har'El, R. Sivan, and A. Soffer. Web-a-where: Geotagging web content. In *SIGIR*, pages 273–280, 2004.

[3] T. Bhattacharya, L. Kulik, and J. Bailey. Automatically recognizing places of interest from unreliable gps data using spatio-temporal density estimation and line intersections. *Pervasive Mob. Comput.*, 19(C):86–107, 2015.

[4] D. M. Blei and J. D. McAuliffe. Supervised topic models. In *NIPS*, pages 121–128, 2007.

[5] S. Chandra, L. Khan, and F. Muhaya. Estimating twitter user location using social interactions–a content based approach. In *SocialCom*, pages 838–843, 2011.

[6] L. Chen, G. Cong, X. Cao, and K. L. Tan. Temporal spatial-keyword top-k publish/subscribe. In *ICDE*, pages 255–266, 2015.

[7] Z. Cheng, J. Caverlee, and K. Lee. You are where you tweet: A content-based approach to geo-locating twitter users. In *CIKM*, pages 759–768, 2010.

[8] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: User movement in location-based social networks. In *SIGKDD*, pages 1082–1090, 2011.

[9] N. Dalvi, R. Kumar, and B. Pang. Object matching in tweets with spatial models. In *WSDM*, pages 43–52, 2012.

[10] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, June 2008.

[11] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *WWW*, pages 769–778, 2012.

[12] B. Hu and M. Ester. Spatial topic modeling in online social media for location recommendation. In *RecSys*, pages 25–32, 2013.

[13] T. Joachims. Training linear svms in linear time. In *SIGKDD*, pages 217–226, 2006.

[14] S. Kinsella, V. Murdock, and N. O'Hare. "i'm eating a sandwich in glasgow": Modeling locations with tweets. In *SMUC*, pages 61–68, 2011.

[15] T. Kurashima, T. Iwata, T. Hoshide, N. Takaya, and K. Fujimura. Geo topic model: Joint modeling of user's activity area and interests for location recommendation. In *WSDM*, pages 375–384, 2013.

[16] G. Li, J. Hu, J. Feng, and K. Tan. Effective location identification from microblogs. In *ICDE*, pages 880–891, 2014.

[17] X. Li, G. Cong, X.-L. Li, T.-A. N. Pham, and S. Krishnaswamy. Rank-geofm: A ranking based geographical factorization method for point of interest recommendation. In *SIGIR*, pages 433–442, 2015.

[18] X. Li, T.-A. N. Pham, G. Cong, Q. Yuan, X.-L. Li, and S. Krishnaswamy. Where you instagram?: Associating your instagram photos with points of interest. In *CIKM*, pages 1231–1240, 2015.

[19] D. Lian and X. Xie. Learning location naming from user check-in histories. In *GIS*, pages 112–121, 2011.

[20] B. Liu, Y. Fu, Z. Yao, and H. Xiong. Learning geographical preferences for point-of-interest recommendation. In *SIGKDD*, pages 1043–1051, 2013.

[21] H. Samet and R. E. Webber. Storing a collection of polygons using quadtrees. *ACM Trans. Graph.*, 4(3):182–222, 1985.

[22] B. Shaw, J. Shea, S. Sinha, and A. Hogue. Learning to rank for spatiotemporal search. In *WSDM*, pages 717–726, 2013.

[23] B. P. Wing and J. Baldridge. Simple supervised document geolocation with geodesic grids. In *HLT*, pages 955–964, 2011.

[24] M. Ye, P. Yin, W.-C. Lee, and D.-L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *SIGIR*, pages 325–334, 2011.

[25] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang. A temporal context-aware model for user behavior modeling in social media systems. In *SIGMOD*, pages 1543–1554, 2014.

[26] H. Yin, B. Cui, L. Chen, Z. Hu, and C. Zhang. Modeling location-based user rating profiles for personalized recommendation. *TKDD*, Jan. 2014.

[27] Q. Yuan, G. Cong, Z. Ma, A. Sun, and N. M. Thalmann. Time-aware point-of-interest recommendation. In *SIGIR*, pages 363–372, 2013.

[28] 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 *SIGKDD*, pages 605–613, 2013.

[29] K. Zhao, G. Cong, Q. Yuan, and K. Q. Zhu. Sar: A sentiment-aspect-region model for user preference analysis in geo-tagged reviews. In *ICDE*, pages 675–686, 2015.