3-2015

Prediction of venues in foursquare using flipped topic models

Wen Haw CHONG
Singapore Management University, whchong.2013@phdis.smu.edu.sg

Bing Tian DAI
Singapore Management University, btdai@smu.edu.sg

Ee Peng LIM
Singapore Management University, eplim@smu.edu.sg

Prediction of Venues in Foursquare Using Flipped Topic Models

Wen-Haw Chong, Bing-Tian Dai, and Ee-Peng Lim

Singapore Management University 80 Stamford Road, Singapore 178902
whchong.2013@phdis.smu.edu.sg, {btdai,eplim}@smu.edu.sg

Abstract. Foursquare is a highly popular location-based social platform, where users indicate their presence at venues via check-ins and/or provide venue-related tips. On Foursquare, we explore Latent Dirichlet Allocation (LDA) topic models for venue prediction: predict venues that a user is likely to visit, given his history of *other* visited venues. However we depart from prior works which regard the users as documents and their visited venues as terms. Instead we 'flip' LDA models such that we regard venues as documents that attract users, which are now the terms. Flipping is simple and requires no changes to the LDA mechanism. Yet it improves prediction accuracy significantly as shown in our experiments. Furthermore, flipped models are superior when we model tips and check-ins as separate modes. This enables us to use tips to improve prediction accuracy, which is previously unexplored. Lastly, we observed the largest accuracy improvement for venues with fewer visitors, implying that the flipped models cope with sparse venue data more effectively.

Keywords: Foursquare, venue prediction, topic models.

1 Introduction

The prevalence and growing popularity of social media in recent years have led to an explosive grow in observable user behavior data. In particular, location-based platforms such as Foursquare and Gowalla provide rich context and user-visitation data. For example, Foursquare users can indicate their presence at venues via check-ins. They can optionally write reviews about visited venues, referred to as tips. These data are fast growing, fine-grained and vast in volume. Currently Foursquare[1] reports a user base of over 50 million, with more than 6 billion check-ins generated. Thus it is not surprising that check-ins has been especially well studied for user profiling and modeling [2,4,5,7].

In this work, we focus on Foursquare due to its market dominance and the ease of accessing related data. Our problem of interest is to predict venues that a user will visit. This translates easily to applications of commercial value, such as user profiling, venue analysis and targeted advertising. For example, venue owners may want to direct their advertisements or promotions at selected new users based on their propensity of visitations.

[1] https://foursquare.com/about

A. Hanbury et al. (Eds.): ECIR 2015, LNCS 9022, pp. 623–634, 2015.

We explore several topic models. Although our work is carried out on Foursquare, the models are easily applicable on venue visitation logs from other platforms. In addition, we also proposed models to handle user generated reviews/tips that are tied to venues. We discussed the targeted problem next.

1.1 Problem Definition

Our prediction task is straightforward: predict venues that a user will likely visit, given historical information of his *other* visited venues. We cast this as a ranking problem. Given a list of candidate venues for each user, we seek to rank venues such that high ranking venues are more likely to be visited by the user.

Our defined problem serves a different purpose and differs from *next* venue prediction [6,8,13] and *time-aware* venue prediction [9,7]. Next venue prediction aims to predict the next venue a user will visit, given additional factors such as a user's current location, time of the day, location of friends etc. Time-aware venue prediction is highly similar, but prediction is for a certain time slot and the user's current location may not be known. In contrast, for our venue prediction task, we do not assume that additional information or contextual constraints such as time are available. The task can also be understood as inferring the overall propensity of a user to visit a venue.

In many cases, the lack of additional information makes venue prediction task harder than next or time-aware venue prediction. For example, consider next venue prediction. With spatial constraints, a user's next venue is likely to be geographically near his current venue [13,6]. Time constraints help as well, e.g. food venues are obviously more likely to be visited during meal times [7]. In addition, for both next and time-aware venue prediction, a venue may be repeatedly visited [13,8] in a user's visitation history, e.g. his home or workplace. All these help to rank or narrow the list of candidate venues. In contrast for our problem, we consider candidate venues that are not visited by the user according to the observed visitation data. Hence, many methods for next venue and time-aware venue prediction tasks are less appropriate to solve the proposed problem.

1.2 Proposed Research Idea

Approach. Our approach is based on Latent Dirichlet Allocation (LDA) [1]. LDA was first introduced for modeling topics in text corpus. Since then, topic models have been widely applied in various domains, including social media platforms. Recent works [4,5] had applied LDA on Foursquare check-ins. Both works model the users as high level documents containing venues as terms. For discussion, we denote this as the base model: **LDA-Udoc**.

Our research idea originates from the key observation that in Foursquare [2], there are many more users than venues. There are many users with little visitation data. On the other hand, venues are often visited by many users who leave traces of check-in's and tips. Hence if we regard venues as documents containing users as terms, we obtain fewer, but longer documents over a larger term dictionary. The question is how these changes affect venue prediction. Based on this

insight, we define the LDA-Vdoc model which is essentially a flipped version of LDA-Udoc, while retaining all the underlying LDA mechanisms. Remarkably, LDA-Vdoc easily outperforms LDA-Udoc in venue prediction.

We consider further LDA extensions, whereby we model check-ins and tips as two separate modes of user behavior. Again, we compare the two design choices. **Vdoc** uses venues as high level documents while **Udoc** does so with users. Our experiments indicate that Vdoc performs better. In fact, the Vdoc model enables us to exploit tips to improve prediction accuracy. To the best of our knowledge, the venue as document idea and multi-modal extension were unexplored in prior works [4,5,6,7] which focused on check-ins (or location logs) only. Our research findings further reveal that accuracy improvement is largest for unpopular venues where there are fewer users, and hence sparser data. Obviously, venues may also have fewer users if they are newly added, thus there are parallels with the *cold-start* problem for new items in recommendation tasks. In such cases, Vdoc outperforms other models significantly.

Contributions. Flipping and the inclusion of tips constitute the novel aspects of our work. In summary, we present two flipped models, Vdoc-LDA and Vdoc for venue prediction in Foursquare. Vdoc-LDA models a single mode. If tips are available as well, we propose to apply Vdoc. Vdoc also copes with sparse venue data more effectively for prediction. This is important since new venues are continuously being added to Foursquare.

2 Models

We shall describe explored models, starting with the vanilla LDA models. Let the number of users, venues and topics be U, V and K respectively. Also let tip words be from a vocabulary of size W. We represent symmetric Dirichlet distributions with hyperparameters α as $\text{Dir}(\alpha)$; and multinomials with parameter vector θ as $\text{Mult}(\theta)$. Other notations are introduced in an inline manner for ease of reading.

2.1 LDA Models

We begin with the base model: **LDA-Udoc**. Traditionally, LDA assumes a text document is generated by sampling a topic for each word, followed by sampling the word conditional on the topic. Let us now regard a document as a user and a word as a check-in/tip venue. Each user u has a latent vector θ_u with a Dirichlet prior $\text{Dir}(\alpha)$. θ_u specifies his distribution over topics z which in turn specifies distributions over venues. The model assumes a single venue mode without differentiating whether users have chosen to check-in and/or write tips at venues. Note that prior work [4,5] had simply used check-ins. However we include venues from tips[2] such that prediction accuracies of all uni-modal and multi-modal models can be fairly compared on a common venue set. Tip words are ignored in LDA-Udoc. Formally, LDA-Udoc has the generative process:

[2] Some users write tips about a venue without generating check-ins.

1. For each user u, sample $\theta_u \sim \text{Dir}(\alpha)$
2. For each topic k, sample $\phi_k \sim \text{Dir}(\beta)$
3. For venue v_i in check-in/tip i of user u, sample:
   (a) Topic $z_i \sim \text{Mult}(\theta_u)$, Venue $v_i \sim \text{Mult}(\phi_{z_i})$

Now we flip the model and propose the **LDA-Vdoc** model, whereby venues *attract* users to check-in and/or write tips. Hence venues play a more active generative role and generate the users. Note that topics are now defined over users instead and denoted by y. LDA-Vdoc also does not differentiate between users from check-ins or tips. Tip words are ignored. The generative process is:

1. For each venue v, sample $\theta_v \sim \text{Dir}(\alpha)$
2. For each topic k, sample $\phi_k \sim \text{Dir}(\beta)$
3. For user u_i in check-in/tip i of venue v, sample:
   (a) Topic $y_i \sim \text{Mult}(\theta_v)$, User $u_i \sim \text{Mult}(\phi_{y_i})$

2.2 Multi-modal Models

We now propose models Udoc and Vdoc which generate check-ins and tips in distinct weakly coupled modes, unlike previous LDA models. With Udoc, venues from check-ins and tips are treated as distinct entity modes generated by check-in and tip topics respectively. However we also tie the mentioned two modes of topics with a common topic indicator. This accounts for the weak coupling and can be viewed as a form of regularization between the two modes. Vdoc is defined in a similar way.

Udoc generates venues, tip content and is a direct, non-flipped extension of the base model Udoc-LDA. It seeks to exploit all information from tips, including the tip words. Since tips are short with a character limit of 200 imposed by Foursquare, we assume each to cover only a single topic. We also attribute each tip word to either the venue or topic with a Bernoulli switch $\text{Bern}(\eta)$, with a prior from a beta distribution $\text{Beta}(\lambda)$. The intuition is that certain venues may have a large influence on tip content.

For each user, venues are now differentiated as check-in venues $\widetilde{v}$ and tip venues $\hat{v}$, generated via check-in topics $\widetilde{z}$ and tip topics $\hat{z}$. Let each tip contains N_w words w. Udoc's generative process is listed below (best understood with the plate diagram in Figure 1).

1. For each user u, sample $\theta_u \sim \text{Dir}(\alpha)$
2. For each topic indicator k, sample distributions for tip topics: $\phi_k \sim \text{Dir}(\beta)$, $\gamma_k \sim \text{Dir}(\omega)$, and check-in topics: $\widetilde{\phi}_k \sim \text{Dir}(\widetilde{\beta})$
3. For each venue v, sample $\hat{\gamma}_v \sim \text{Dir}(\hat{\omega})$
4. Sample a global Bernoulli vector for flags: $\eta \sim \text{Beta}(\lambda)$
5. For tip i of user u, sample tip topics, tip venues and words:
   (a) Topic $\hat{z}_i \sim \text{Mult}(\theta_u)$, Venue $\hat{v}_i \sim \text{Mult}(\phi_{\hat{z}_i})$
   (b) For the j-th word $w_{i,j}$
      i. Sample a flag $x_{i,j} \sim \text{Bern}(\eta)$
      ii. Sample $w_{i,j} \sim \text{Multi}(\gamma_{\hat{z}_i})$ if $x_{i,j}$=0, else sample $w_{i,j} \sim \text{Multi}(\hat{\gamma}_{\hat{v}_i})$

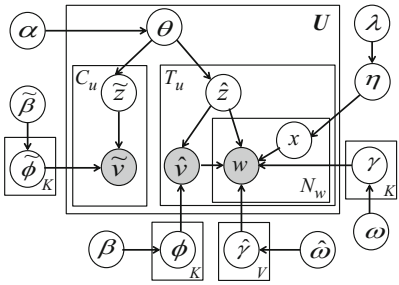


Fig. 1. Udoc model. Each user u has C_u check-ins and T_u tips.

6. For check-in i of user u, sample check-in topics and check-in venues:
   (a) Topic $\widetilde{z}_i \sim \text{Mult}(\theta_u)$, Venue $\widetilde{v}_i \sim \text{Mult}(\widetilde{\phi}_{\widetilde{z}_i})$

Vdoc is a flipped version of Udoc and regards each venue as a document unit. Intuitively, each venue attracts users to either check-in, write tips or do both. In addition, we observed in our Foursquare dataset of an Asian city, (refer Section 3.1) that 76% of venues have both check-ins and tips. In contrast, only 21% of users both check-in and write tips, with the rest being biased towards only one behavior mode. In this sense, more venue documents have both modes and can be regarded as more 'complete' than user documents. This will impact prediction accuracy as shown in our experiments (refer section 3.2).

For each venue, users are now differentiated as check-in/tip users ($\widetilde{u}/\hat{u}$), generated via check-in/tip topics ($\widetilde{y}/\hat{y}$). We also let tip words to be attributable to either tip topics or tip users. We now define Vdoc's generative process with the corresponding plate diagram shown in Figure 2.

1. For each venue v, sample $\theta_v \sim \text{Dir}(\alpha)$
2. For each topic indicator k, sample distributions for the tip mode: $\phi_k \sim \text{Dir}(\beta)$, $\gamma_k \sim \text{Dir}(\omega)$, and check-in mode: $\widetilde{\phi}_k \sim \text{Dir}(\widetilde{\beta})$
3. For each user u, sample $\hat{\gamma}_u \sim \text{Dir}(\hat{\omega})$
4. Sample a global Bernoulli vector for flags: $\eta \sim \text{Beta}(\lambda)$
5. For tip i at venue v, sample tip topics, tip users and words:
   (a) Topic $\hat{y}_i \sim \text{Mult}(\theta_v)$, User $\hat{u}_i \sim \text{Mult}(\phi_{\hat{y}_i})$
   (b) For the j-th word $w_{i,j}$
      i. Sample a flag $x_{i,j} \sim \text{Bern}(\eta)$
      ii. Sample $w_{i,j} \sim \text{Multi}(\gamma_{\hat{y}_i})$ if $x_{i,j}$=0, else sample $w_{i,j} \sim \text{Multi}(\hat{\gamma}_{\hat{u}_i})$
6. For check-in i at venue v, sample check-in topics and check-in users:
   (a) Topic $\widetilde{y}_i \sim \text{Mult}(\theta_v)$, User $\widetilde{u}_i \sim \text{Mult}(\widetilde{\phi}_{\widetilde{y}_i})$

2.3 Inference

We use Collapsed Gibbs Sampling (CGS) to infer parameters for all the models. CGS draws a sequence of samples to approximate joint distributions. It has been widely used for inference [3] in LDA-based models. For the multi-modal models,

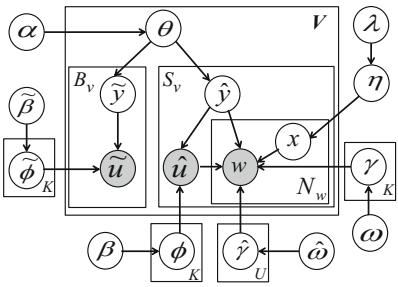


Fig. 2. Vdoc model. Each venue v has B_v check-ins and S_v tips.

Udoc and Vdoc's sampling equations are highly similar in form. Due to space constraints, we only present sampling equations for Vdoc topics.

The topic inference task is to sample for tip and check-in topics. For notation simplicity, we also omit hyperparameters which are implicitly conditioned upon during sampling. Recall that in Vdoc, venues $\boldsymbol{v}$ are not differentiated while users are differentiated as check-in users $\widetilde{\boldsymbol{u}}$ and tip users $\hat{\boldsymbol{u}}$. Given a tip i with bag of words $\boldsymbol{w}_i$, we sample its topic as follows:

$$p(\hat{y}_i = k|, \hat{\boldsymbol{y}}_{-i}, \hat{\boldsymbol{u}}, \boldsymbol{v}, \boldsymbol{w}, \boldsymbol{x}) \propto \frac{N^{TV}_{kv_i,-i} + \alpha}{\sum_{k'} N^{TV}_{k'v,-i} + K\alpha} \frac{N^{\hat{U}T}_{\hat{u}_i k,-i} + \beta}{\sum_{\hat{u}'} N^{\hat{U}T}_{\hat{u}'k,-i} + U\beta} \prod_{\substack{w \in \boldsymbol{w}_i, \\ x_w = 0}} \frac{N^{WT}_{wk,-i} + \omega}{\sum_{w'} N^{WT}_{w'k,-i} + W\omega} \quad (1)$$

where subscript $-i$ means contributions from tip i are excluded. N^{TV}, $N^{\hat{U}T}$ and N^{WT} are respective count matrices for assignments of topics to venues, tip users to topics and tip words to topics. Subscripts reference the matrix elements. For a check-in i, we sample its topic as:

$$p(\widetilde{y}_i = k|, \widetilde{\boldsymbol{y}}_{-i}, \widetilde{\boldsymbol{u}}, \boldsymbol{v}) \propto \frac{N^{TV}_{kv_i,-i} + \alpha}{\sum_{k'} N^{TV}_{k'v,-i} + K\alpha} \frac{N^{\widetilde{U}T}_{\widetilde{u}_i k,-i} + \widetilde{\beta}}{\sum_{\widetilde{u}'} N^{\widetilde{U}T}_{\widetilde{u}'k,-i} + U\widetilde{\beta}} \quad (2)$$

where $N^{\widetilde{U}T}$ counts assignments of check-in users to topics and N^{TV} is previously defined. Similarly, the sampling equations for flag assignments per tip word can be readily derived. We omit their discussion here for brevity.

2.4 Prediction

Our goal is to predict the venues that a user is likely to visit. We do not differentiate between check-in/tip venues in prediction, hence the targeted quantity is $p(v|u)$. This is used to rank candidate venues. While in practice, a user can tip without having actually visited a venue, extensive inspections of sample tips indicate that it is reasonable to assume most tips are generated post-visits.

For Udoc-LDA, $p(v|u)$ is computed via topic marginalization: $\sum_z p(v|z)p(z|u)$. To obtain $p(v|u)$ for Udoc, topic marginalization is done for each mode and then combined with the two observed empirical probabilities of u performing a check-in and tip. For Vdoc-LDA, we marginalized out topics over users and then apply Bayes theorem $p(v|u) \propto p(u|v)p(v)$. The same formula applies to Vdoc as well, however we first need to compute $p(u|v)$. Assume that a venue v generates check-ins and tips with conditional probabilities $p(c|v)$ and $p(t|v)$. We compute $p(u|v)$ by marginalizing over modes: $m = \{c, t\}$ and applying the chain rule:

$$p(u|v) = p(u, m = c|v) + p(u, m = t|v) = p(\widetilde{u}|v)p(c|v) + p(\hat{u}|v)p(t|v) \quad (3)$$

Note that $p(\widetilde{u}|v)$ and $p(\hat{u}|v)$ in (3) are obtained via marginalizing out the topics:

$$p(\widetilde{u}|v) = \sum_{\widetilde{y}} p(\widetilde{u}|\widetilde{y})p(\widetilde{y}|v), \quad p(\hat{u}|v) = \sum_{\hat{y}} p(\hat{u}|\hat{y})p(\hat{y}|v) \quad (4)$$

where $p(\widetilde{y}|v)$, $p(\hat{y}|v)$, $p(\hat{u}|\hat{y})$ and $p(\widetilde{u}|\widetilde{y})$ are estimated with count matrices from CGS in a similar fashion as proposed in [3].

3 Experiments

3.1 Data and Setup

In our experiments, we use two Foursquare datasets: United States (US) check-ins from [2] and check-ins plus tips which we extract from users in Singapore (SG), spanning Mar 2012 to Dec 2013. The latter comprises of check-ins posted as tweets on the user's Twitter timeline[3] and tips crawled directly using the Foursquare API. Following standard noise filtering practices [4,7,10,6], we exclude inactive users with too few venues and inactive venues with too few users. We used a common threshold of 6 for both user and venue filtering, i.e. ≥ 6.

For each user, we randomly select one of his venues as the test venue. We then hide *all* his tips and check-ins from the test venue. His remaining tips/check-ins are then included in the training set for model building. This process is repeated for all users. We generate 10 trials of training/test sets whereby trials differ due to random sampling of test venue per user. Also note that *prediction here is in terms of retrieving hidden venues*, and that to support multiple trials, we have not restricted hidden venues to be necessarily the most recent visited venues.

On average, the US training set contains 48,900+ users, 14,900+ venues and 252,000+ check-ins. The SG training set contains 24,400+ users, 17,600+ venues, 62,900+ tips and 1,062,200+ check-ins. Comparing both datasets, the US dataset has more users and fewer venues than the SG dataset.

Note that for the US dataset, we only apply LDA-Vdoc and LDA-Udoc since tips are not available. For the SG dataset, we ignore tip content when applying uni-modal models, such that there are no differentiation between entities (users or venues) from check-ins/tips. With each model, we rank candidate venues for

[3] Check-ins are visible only if posted as tweets, otherwise they are hidden.

each user (excluding those in his training set). Hence for each user, the number of candidates is slightly less than the number of venues per dataset. We then extract the rank of the hidden test venue and compute the Mean Reciprocal Rank (MRR), a standard information retrieval measure defined as:

$$MRR = \frac{1}{Q} \sum_{i}^{Q} 1/rank_i \tag{5}$$

where $rank_i$ is the rank of the hidden test venue i predicted by the model and Q is the total number of test cases. (Each test case consists of a user and his hidden test venue.) MRR lies between 0 and 1 with the latter implying perfect ranking accuracy. We compute the average MRR across the 10 trials.

All models are fitted using 500 iterations of CGS with a burn-in of 200 iterations. For estimating distributions required for prediction, we collect samples with a lag of 20 iterations in between. We have experimented with various number of topics and observed that relative prediction performance of models are fairly consistent, e.g. Vdoc being consistently the best performer. In subsequent discussion, we present results involving 20 topics.

3.2 Prediction Results

In this section, we compare the models quantitatively. We regard LDA-Udoc as the baseline and focus on how other models perform relative to it. Table 1 presents the prediction results. Also recall that our notion of documents depends on the models. For LDA-Vdoc and Vdoc, documents are venues while for LDA-Udoc and Udoc, documents are users.

Table 1. Average MRR with standard deviations (bracketed). Gain is % improvement over LDA-Udoc. (US: United States check-ins, SG: check-ins & tips in Singapore).

Dataset	Model	Ave. MRR	Gain (%)
US	LDA-Vdoc	**0.1302 (2.05E-3)**	**22.35**
	LDA-Udoc	0.1064 (1.77E-3)	-
SG	Vdoc	**0.0575 (1.34E-3)**	**7.06**
	Udoc	0.0532 (1.21E-3)	-0.89
	LDA-Vdoc	0.0564 (0.93E-3)	4.92
	LDA-Udoc	0.0537 (1.25E-3)	-

On both datasets, LDA-Vdoc easily outperforms the previously proposed LDA-Udoc model [4,5]. This supports the argument of flipping. Accuracy gain is especially large at over 20% on the US dataset. As described in section 3.1, the US dataset has more users and yet, fewer venues than the SG dataset. This means that in the former, LDA-Vdoc's characteristics are even more pronounced, i.e. modeling fewer and longer documents. Hence we expect a larger accuracy gain over LDA-Udoc, compared to the SG dataset.

On the SG dataset, Vdoc is the best performer with more than 7% improvement over the baseline. The difference is consistent across different runs and statistically significant (using the Wilcoxon signed rank test) with a p-value of less than 0.01. In addition, LDA-Vdoc consistently emerges as the second best performer (p-value < 0.01) when compared with LDA-Udoc). Hence models using venues as documents (as in Vdoc, LDA-Vdoc) consistently perform better than models with users as documents.

Vdoc's superiority over LDA-Vdoc indicates that tips contain useful information, which the former had exploited. However we note that while Udoc considers tips as well, its performance is essentially the same as LDA-Udoc. We attribute this to overly sparse co-occurrence information. Obviously, in breaking up entities into different modes, some co-occurrence information is lost. (To see this, imagine treating every entity as a unique mode. This leads to a total loss of co-occurrence information.) Thus additional information from tips may have been cancelled off in Udoc. Vdoc is however more robust to this effects.

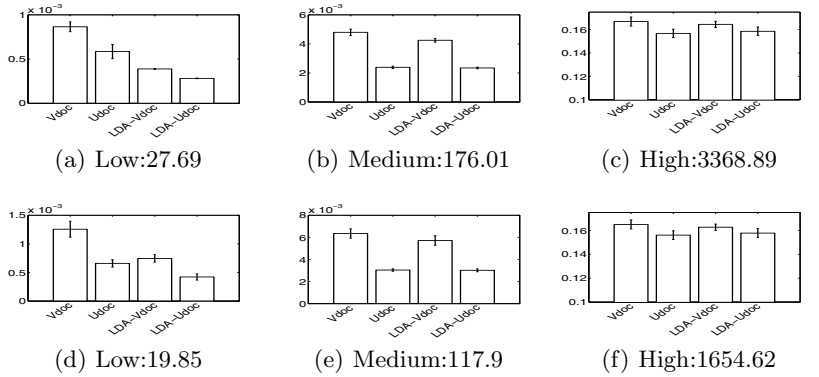
We attribute Vdoc's robustness to previously discussed characteristics such as having fewer, but longer documents. In addition, Vdoc's documents are more complete than Udoc's documents in containing entities from both modes. As mentioned in Section 2.2, 76% of venues (Vdoc's documents) from the SG dataset contain users from both tips and check-ins. In contrast, with users as documents (as in Udoc), only 21% contains both tip and check-in venues. This is a direct consequence of how users utilize Foursquare, i.e. leaning towards either generating check-ins or writing tips rather than doing both in a more balanced manner.

3.3 Prediction Results by Venue Popularity

For a more in-depth analysis, we bin test cases for the SG dataset by test venue popularity. This allows us to examine how various models perform on venues of different popularities. We quantify venue popularity by two measures: combined tip/check-in count and number of unique users per venue. We divide test cases into three bins of equal size, corresponding to venues of *low*, *medium* and *high* popularities. Figure 3 shows the MRR of venues with different popularities.

Figures 3(a) and 3(d) show that Vdoc's accuracy improvement over other models is biggest for the least popular venues. The improvement decreases as we consider more popular venues. For low popularity venues, Vdoc outperforms the baseline LDA-Udoc by around 200% for both popularity measures, hence indicating that Vdoc makes better use of sparse venue data. This takes on an even greater importance if we consider a common scenario in Foursquare: newly created venues will usually belong to the unpopular bins simply by virtue of having little or no previous data. Predicting for them is analogous to recommending for new items in recommender systems, which relates to the *cold start* problem. In such cases, prediction/recommendation difficulty increases due to data sparsity. Compared with other models, Vdoc is more accurate in such scenarios.

For highly popular venues, Figures 3(c) and 3(f) show that Vdoc's improvement over LDA-Udoc is smaller at 4-5% for both popularity measures. Hence, even though popular test venues are easier to predict for, Vdoc still manages



(a) Low:27.69 (b) Medium:176.01 (c) High:3368.89

(d) Low:19.85 (e) Medium:117.9 (f) High:1654.62

Fig. 3. MRR binned by combined tip/check-in count (a,b,c) and unique user count (d,e,f). Each sub-figure plots Vdoc, Udoc, LDA-Vdoc and LDA-Udoc (left to right). Numbers are mean tip/check-in count for (a,b,c), and mean user count for (d,e,f).

some improvement over Udoc-LDA. We also compare Vdoc with LDA-Vdoc. Their performance differs more for low popularity venues, and less with increased venue popularity. Since unpopular venues have much less data for models to exploit, content information in the few related tips will be relatively more important. Vdoc is able to exploit this additional information in contrast to LDA-Vdoc which totally ignores content.

4 Sample Topics

We illustrate some Vdoc's topics over tip words on the SG dataset. By inspecting the topics, one easily gets a understanding of user interests and the aspects that they care about enough to write tips. Table 2 shows the top 12 words of 6 sample topics (out of 20) from Vdoc. As can be seen, the topics are easily interpretable.

Table 2. Top 12 words of sample Vdoc topics. We manually annotate the displayed topics (labels in bold) for ease of understanding.

Service: service food staff slow bad time order long wait good don waiting
Transport: bus service time interchange train long will queue wait station mins morning
Pastry: ice cream chocolate cake tea nice good love best sweet caramel awesome
Tea/Coffee: tea milk ice coffee nice best good jelly drink love sugar green
Western Food: good chicken cheese beef pasta fries sauce great nice awesome fish pizza
Opening hours: hours closed open public till sat sun mon fri daily place opens

5 Related Work

As mentioned, [4,5] had applied LDA to model Foursquare check-ins. They presented qualitative analysis of the topics instead of quantitative results. In [6],

Kurashima et. al modeled venues conditional on both topics and each user's movement history. The model was used to predict the last visited venue of the user. Note that all the above mentioned works treated users as documents, venues as terms and topics as distributions over venues.

Some works [11,12,10] had explored topic models of geo-located tweets. Tweet contents and originating locations are used in [11,12] while [10] included time information as well. The aim is to predict geographic coordinates that tweets are sent from. This problem is less applicable on Foursquare since tips can possibly be generated post-visits by users when they may not be physically present at the venue locations. Nonetheless, we note that all the proposed models [11,12,10] had utilized users as documents, instead of spatial regions or locations as documents. Potentially model flipping can be investigated for accuracy gains.

Other researchers had explored non topic modeling approaches in next venue [8,13] and time-aware venue prediction [7,9]. In [8], dynamic Bayesian networks were used to model a user's locations as hidden states. Each state is conditional on the last state and emit observations such as time information and the locations of friends. Noulas et al. [13] trained M5 model trees with mobility and temporal features to predict a user's next check-in venue. Yuan et al. [7] constructed a time-aware collaborative filtering model to predict user locations conditional on time. Cho et al. [9] conducted a similar task with a mixture of Gaussians.

Lastly we mention works more applicable to our prediction task, but with models in the continuous space [9,15,14]. (We mentioned [9] earlier for time-aware venue prediction, but it can be adapted for this). Typically continuous distributions such as Gaussian mixtures [9,15] or kernel estimated densities [14], are fitted to model the spatial coordinates of venues. In contrast, we model venues in the discrete space and do not require spatial coordinates. Both continuous and discrete modeling have their strengths and weaknesses. For example, different venues can occur at the same coordinates, by being at different levels of the same building. Predicting between these venues is tricky with continuous modeling, which by far, had mainly utilized two dimensional distributions [9,14,15]. Nonetheless, in our further work we will be interested in fusing the models presented here with continuous techniques such that the strengths of both can be leveraged on. We describe a possible research direction in our conclusion.

6 Conclusion

We have explored several LDA based models for venue prediction in Foursquare. In particular, we consider flipped models such that venues are treated as documents and users as terms. Flipping is extremely easy to apply, and yet leads to significant accuracy gains in venue prediction. It also has the additional benefit of allowing us to exploit tips. Without flipping, it is uncertain that including tips can increase accuracy, e.g. Udoc does not improve over Udoc-LDA.

In ongoing research, we are exploring the fusion of the models here with continuous models [15,14,9]. Instead of designing ever more complex generative models, one possible approach is to combine different models, either linearly or

otherwise. This allows information from various diverse aspects, e.g. tips, spatial and social influence to contribute to the prediction task. In addition, the inferred combination weights serve to indicate the relative importance of various aspects.

Lastly, given the huge variety of topic models out there in different applications, many can potentially be flipped and the performance investigated. Researchers can also consider flipped/non-flipped versions in the design of any new models. Hence our works here has served as a motivating example.

Acknowledgements. This research is partially supported by DSO National Laboratories, Singapore; and the Singapore National Research Foundation under its International Research Centre@Singapore Funding Initiative and administered by the IDM Programme Office, Media Development Authority (MDA).

References

1. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993–1022 (2003)
2. Levandoski, J.J., Sarwat, M., Eldawy, A., Mokbel, M.F.: LARS: A Location-Aware Recommender System. In: ICDE (2012)
3. Griffiths, T.L., Steyvers, M.: Finding Scientific Topics. Proceedings of the National Academy of Sciences 101, 5228–5235 (2004)
4. Joseph, K., Tan, C.H., Carley, K.M.: Beyond "Local", "Categories" and "Friends": Clustering Foursquare Users with Latent "Topics". In: UbiComp (2012)
5. Long, X., Jin, L., Joshi, J.: Exploring Trajectory-driven Local Geographic Topics in Foursquare. In: UbiComp (2012)
6. Kurashima, T., Iwata, T., Hoshide, T., Takaya, N., Fujimura, K.: Geo topic Model: Joint Modeling of User's Activity area and Interests for Location Recommendation. In: WSDM (2013)
7. Yuan, Q., Cong, G., Ma, Z., Sun, A., Magnenat-Thalmann, N.: Time-aware Point-of-Interest Recommendation. In: SIGIR (2013)
8. Sadilek, A., Kautz, H.A., Bigham, J.P.: Finding your Friends and Following them to Where You Are. In: WSDM (2012)
9. Cho, E., Myers, S.A., Leskovec, J.: Friendship and Mobility: User Movement in Location-based Social Networks. In: KDD (2011)
10. Yuan, Q., Cong, G., Ma, Z., Sun, A., Magnenat-Thalmann, N.: Who, Where, When and What: Discover Spatio-temporal Topics for Twitter Users. In: KDD (2013)
11. Hong, L., Ahmed, A., Gurumurthy, S., Smola, A., Tsioutsiouliklis, K.: Discovering Geographical Topics in the Twitter Stream. In: WWW (2012)
12. Hu, B., Ester, M.: Spatial Topic Modeling in Online Social Media for Location Recommendation. In: ACM Conference on Recommender Systems (2013)
13. Noulas, A., Scellato, S., Lathia, N., Mascolo, C.: Mining User Mobility Features for Next Place Prediction in Location-Based Services. In: ICDM (2012)
14. Lichman, M., Smyth, P.: Modeling Human Location Data with Mixtures of Kernel Densities. In: KDD (2014)
15. Zhao, S., King, I., Lyu, M.R.: Capturing Geographical Influence in POI Recommendations. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) ICONIP 2013, Part II. LNCS, vol. 8227, pp. 530–537. Springer, Heidelberg (2013)