On the Semantic Annotation of Places in Location-Based Social Networks

Mao Ye[1†], Dong Shou[1†], Wang-Chien Lee[1†], Peifeng Yin[1†] and Krzysztof Janowicz[2‡]
[†]Department of Computer Science & Engineering, The Pennsylvania State University, PA, USA.
[‡]Department of Geography, University of California, Santa Barbara, CA, USA.
[1]{mxy177,dus212,wlee,pzy102}@cse.psu.edu [2]{jano}@geog.ucsb.edu

ABSTRACT
In this paper, we develop a semantic annotation technique for location-based social networks to automatically annotate all places with category tags which are a crucial prerequisite for location search, recommendation services, or data cleaning. Our annotation algorithm learns a binary support vector machine (SVM) classifier for each tag in the tag space to support multi-label classification. Based on the check-in behavior of users, we extract features of places from i) *explicit patterns* (EP) of individual places and ii) *implicit relatedness* (IR) among similar places. The features extracted from EP are summarized from all check-ins at a specific place. The features from IR are derived by building a novel *network of related places* (NRP) where similar places are linked by virtual edges. Upon NRP, we determine the probability of a category tag for each place by exploring the relatedness of places. Finally, we conduct a comprehensive experimental study based on a real dataset collected from a location-based social network, *Whrrl*. The results demonstrate the suitability of our approach and show the strength of taking both EP and IR into account in feature extraction.

Categories and Subject Descriptors
I.2.6 [**Artificial Intelligence**]: Learning; J.4 [**Computer Applications**]: Social and Behavior Sciences

General Terms
Algorithms, Design, Experimentation

Keywords
Semantic Annotation, Points of Interest, User Behavior, Location-Based Social Networks

1. INTRODUCTION
With the increasing availability of GPS-enabled smart phones, rapid development of location-based services, and growing interests in on-line social networking, a number of location-based social networking (LBSN) services such as Whrrl[1], Foursquare[2] and Facebook Places[3] have emerged. These services allow users to explore places, write reviews, and share their locations and experiences with others. The number of available *places* in LBSNs is growing continuously.[4] Many places have been labeled with useful tags such as *restaurant* or *cinema*, which are crucial for assisting users in searching and exploring new places as well as for developing recommendation services [2, 12, 28]. However, based on our analysis of data collected from Whrrl and Foursquare, about 30% of all places are lacking any meaningful textual descriptions. To address this problem, we develop a novel technique, namely *semantic annotation of places* (SAP), to automatically and precisely annotate all places with *semantic tags* for LBSNs.

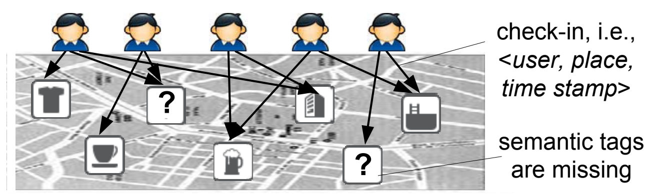


Figure 1: Users and places in an LBSN

Figure 1 shows a bipartite graph representation of the user-place relationship in an LBSN. Let U and P denote the set of all users and places in the system, respectively. Users and places are connected through a set of check-in activities $C = \{\langle u, p, h\rangle | u \in U \wedge p \in P \wedge h \in H\}$, where H is a set of time stamps. Each check-in $c \in C$ describes that "a user u has checked in a place p at time h". Note that a place p_i may be annotated by users with a set of semantic tags $T_i \subseteq T$, where T presents the tag space. Our proposed SAP technique assigns tags to places where semantic tags are missing. These targeted places are depicted with question marks in Figure 1.

The problem of *place semantic annotation* can be formulated as predicting appropriate tags for a given place. In LBSNs, a place may be associated with multiple tags. For instance, a place associated with a tag *restaurant* may also be tagged with *bar*. Hence, place semantic annotation in LBSNs may be addressed as a *multi-label classification* problem [3, 29]. While multi-label classification techniques have been developed for many applications, such as protein function classification [6], music categorization [13] and semantic scene classification [3], the problem has not been explored previously under the context of LBSNs, where we can only

[1]www.whrrl.com
[2]www.foursquare.com
[3]www.facebook.com/places
[4]*Points of Interest (POIs)* are usually referred to as *places* in LBSNs.

KDD'11, August 21–24, 2011, San Diego, California, USA.

operate over user check-in activities (i.e., $\langle u, p, h \rangle \in C$) for certain places and time stamps.

We propose to address the place semantic annotation problem by learning a binary SVM for each tag in the tag space in order to realize the multi-label classification. To do so, a fundamental issue is to identify and extract a number of descriptive features for each place in the system. Selecting the right features is important because those features have a direct impact on the effectiveness of the classification task. As mentioned earlier, the only data resource we have is the user check-in activities at various places and times. Therefore, we explore the user behaviors and seek unique features of places captured in the check-in activities. Fortunately, human behaviors are not completely random, e.g., people usually visit restaurants for lunch at around noon. Moreover, people exhibit patterns in their activities, e.g., different places visited by the same person at the same time may be similar (e.g., having the same tags).

By leveraging the observations hinted in the above-mentioned examples, we extract features of places in two different but complementary aspects: 1) explicit patterns (EP) at individual places; and 2) implicit relatedness (IR) among similar places. Features extracted from EP, corresponding to a given place, can be derived from all check-ins at the place based on statistical analysis. In this paper, we propose to extract *population features* (e.g., number of unique visitors) and *temporal features* (e.g., distribution of check-in time) as semantic descriptions of specific places. On the other hand, we extract features from IR to capture the relatedness among places by exploiting the regularity of user check-in activities to similar places. Since only some places are tagged, we could make good use of IR by deriving descriptive features of a given place from its "related" places.

To facilitate extraction of features from IR, we develop a novel algorithm to build a *network of related places* (NRP) that captures the relatedness amongst places by exploring regularities of user check-ins to similar places. We propose a family of graph representations that capture the user-place and time-place relationships from the user check-in activities. We employ the Random Walk and Restart technique [27] on these graphs to estimate the relatedness of places in order to build an NRP. In the obtained NRP, place pairs with high relatedness values imply high similarity in the tag space and thus are linked. Accordingly, we derive the probability for a specific tag being labeled to a place (called *label probability*) from its linked (similar) places. This label probability is thus treated as a feature of IR, along with population and temporal features derived from EP, to feed the binary SVM in our SAP algorithm.

This research work has made a number of significant contributions, as summarized below.

- We propose to tackle the problem of place semantic annotation in LBSNs, which is a crucial prerequisite for effective retrieval and recommendation of POIs in LBSNs.

- We formulate the task of place semantic annotation in LBSNs as a multi-label classification problem and propose a two-phase algorithm to learn a binary SVM classifier for each tag in the entire tag space. In the proposed semantic annotation of places (SAP) algorithm, we explore explicit patterns (EP) at individual places and implicit relatedness (IR) among similar places by exploiting the user check-in activities to extract descriptive features for places. To overcome several technical challenges in realizing the proposed algorithm, we develop a number of techniques for extracting *population* and *temporal* features, building a network of related places (NRP), and deriving label probability for each place in the system.

- Through a comprehensive experimental study, using a real dataset collected from Whrrl, we validate our proposed ideas and evaluate our SAP algorithm in terms of three different feature sets: i) features extracted from EP, ii) features extracted from IR, and iii) all features (i.e., combination of i) and ii)). The experimental results show that using all features achieves the best performance among the three tested feature sets. More importantly, features derived from both EP and IR contribute significantly in unique aspects to the classification of different semantic tags. For example, features from EP are effective in labeling tags such as *restaurant* and *nightlife* because most people exhibit similar behaviors in visiting restaurants and nightlife places. On the other hand, features from IR are excellent for tagging places related to *shopping* as some people exhibit strong patterns in such activities.

The remainder of this paper is organized as follows. In Section 2, we review related works. Next, in Section 3, we give an overview of the proposed semantic annotation of places (SAP) algorithm, describe how we extract EP and IR features, and detail the realization of our SAP algorithm. In Section 4, we further discuss the issue of extracting features from IR and detail our approach. In Section 5, we conduct an empirical study using the collected Whrrl dataset and analyze our results. Finally, in Section 6, we conclude this work and point out future directions.

2. RELATED WORK

In this section, we review a number of existing works in the areas of data mining, multi-label classification, and classification of networked data.

Due to the increasing availability of location-based services and GPS-enabled devices, real traces of user locations and activities have been collected and used in several studies [1, 14, 15]. A variety of approaches for projecting user trajectories from GPS data have been proposed, including particle filtering [10], Markov models [1], Dynamic Baysian Networks [14], and Eigenbehaviors [7]. Data traces used in these studies typically do not contain explicit information regarding user activities. The LBSN data investigated in our research is unique in two aspects: i) semantic tags associated with places provide rich information about categories and activities (e.g., food, restaurant, hotel, shopping, etc.); and ii) user check-ins logged in LBSNs usually are not continual, thus revealing partial views of user activities. These differences bring several new challenges to our research. The most related work is [15], which studies people's naming preferences. The authors argue that people have different naming preferences under different contexts, and thus use a wide range of terms such as *home* and *near Liberty Bridge* to disclose their locations to others. Notice that, even though our work also explores people's naming preferences on places, we focus on enriching places with semantic tags such as *restaurant* and *cinema* for supporting location search and retrieval.

Place semantic annotation has been formulated as a multi-label classification problem in this paper. Previous studies on multi-label classification have primarily been conducted in the application domains of text classification [25, 19], protein function classification [6], music categorization [13], and semantic scene classification [3]. In [25], BoosTexer, extending AdaBoost [8], has been developed to handle multi-label text categorization. In [19], a mixture model derived by expectation maximization (EM) has been trained to select the most probable set of labels from the power set of possible classes. In [11], a set of binary SVM classifiers have been developed to realize multi-label classification for text classification. In [6], the notion of entropy has been extended to include multi-label data for gene expression in order to generate accurate rules for gene expression comprehension. In [21], geographical knowledge has been employed to help assign semantic tags to geo-tagged Flickr photos. Note that place semantic annotation for LBSNs is a new research topic that has not been studied previously.

To derive correlations amongst places from patterns of individual check-ins, we depict users and places as nodes of a bipartite graph as shown in Figure 1. We then construct a network of related places to facilitate classification. There exists some work on classifying networked data, which are generally of the same type such as web-pages or text documents connected via various explicit relations (e.g., hyperlinks [16]). Studies on simultaneously inferring interrelated values over networked data have been reported in [4, 26]. In [17], a simple univariate classifier, called the weighted-vote relational neighbor (wvRN), has been developed by obtaining a weighted average of the estimated class membership scores of the nodes' neighbors. Moreover, similar to [4], a relaxation labeling method has been proposed for collective inference [24]. In [18], a case study on learning attributes of network data has been presented. In [20], a *cautious* collective classification that adopts only top-k most confidently predicted labels has been proposed. Gallagher et al. propose *ghost edges* to create edges between nodes based on the intrinsic structure of the networks to improve the classification of sparse labels [9].

3. SEMANTIC ANNOTATION OF PLACES

We design a two-phase algorithm to address the place semantic annotation problem. The first phase takes care of the feature extraction, while the second phase handles the semantic annotation. The task of feature extraction explores two lines of ideas as discussed earlier in the Introduction. On the one hand, we explore the explicit patterns (EP) corresponding to a specific place to abstract aggregated user behaviors as *population features* and *temporal features*. On the other hand, we explore the implicit relatedness (IR) amongst places in order to formulate descriptive features of a given place from its similar places. Features derived from EP and IR are used to learn a binary SVM for each tag in the tag space in the semantic annotation phase. Given a place, the prediction by a specific SVM classifier decides whether this place belongs to the category of the corresponding semantic tag or not. After checking all SVM classifiers, we obtain all qualified semantic tags for the place under examination.

3.1 Features from EP

Our goal is to extract discriminative EP features from places with the same tag. Intuitively, users behave differently at different places due to the nature of functions and activities offered by these places. As a result, different patterns, naturally formed in aggregated behaviors of visitors to various kinds of places, are embedded in the user check-in activities which are logged in LBSNs. In a check-in record, the most important information is user and time, besides the place itself. In the following, we propose to extract several population features and temporal features to depict places as below.

- F_1 (*total number of check-ins*) - Based on the observation from the Whrrl dataset, shown in Figure 2, we find the number of check-ins to a restaurant is usually larger than the number of check-ins to a hospital. Hence the number of check-ins, which is discriminative for the classification of places such as restaurants and hospitals, is a good population feature for semantic annotation.

- F_2 (*total number of unique visitors*) - This feature focuses on the number of unique visitors. Based on our analysis on the Whrrl dataset, we find F_2 to be a similar phenomenon to F_1. Thus, we aggregate the number of unique visitors at a specific location as the second population feature extracted from EP.

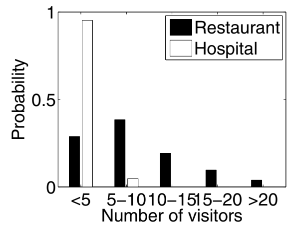


Figure 2: Distr. of # of visitors

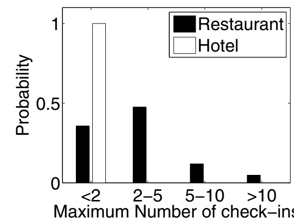


Figure 3: Distr. of Max # of check-ins by a single visitor

- F_3 (*maximum number of check-ins by a single visitor*) - As shown in Figure 3, people may check in a place tagged as restaurant for multiple times, while they may check in a hotel for only 1-2 times. Thus, the maximum number of check-ins by a single user at a place is useful to decide whether a place is a restaurant or a hotel. We use it as the third population feature extracted from EP.

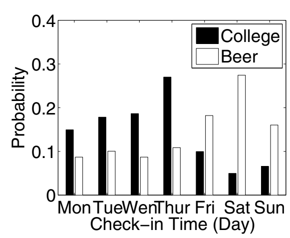


Figure 4: Distr. of check-in time (day)

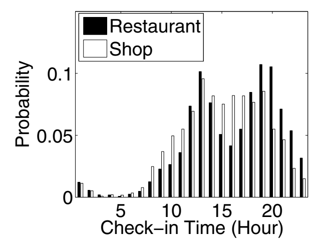


Figure 5: Distr. of check-in time (hour)

- F_4 (*distribution of check-in time in a week*) - We analyze the distribution of check-ins at different categories of places over the days of a week. As shown in Figure 4, users check in college campuses more often on weekdays than on weekends. On the contrary, they check in bars on weekends more frequently than on weekdays. Since there are different distributions of check-in days for different kinds of places, we consider the distribution to be a very useful temporal feature.

- F_5 (*distribution of check-in time in 24-hour scale*) - By plotting the distribution of check-ins in the 24-hours time scale, we show in Figure 5 two very different distribution patterns corresponding to two kinds of places (i.e., restaurant and shop). There are clearly two peak times, corresponding to lunch and dinner periods, for places associated with the tag *restaurant*. On the other hand, shopping time looks like a normal distribution with most activities between 7:00am and 8:00pm, while there is no obvious peak shopping time observed. These observations regarding the patterns of massive visitors at different kinds of places provide strong support that check-in time distributions in 24-hours time scale is a good temporal feature for semantic annotation.

Note that, besides of the aforementioned patterns, check-in activities at different places may show seasonal patterns, e.g., most people go to ski areas during winter. However, due to the limited time span in the period of data collection, we only consider F_4 and F_5 as the temporal features in this study.

3.2 Features from IR

As discussed in [7], there is regularity in people's activities. Take one of the Whrrl users as an example. We find that the user visits places in the *performing arts and entertainment* category (including museums and galleries) in the morning (at around 10:00am), visits places for food at lunch/dinner time, and usually goes shopping at around 4:00pm. Such regularity appears in certain users and thus can be used for correlating similar places. However, extracting features from implicated relatedness (IR) among similar places (e.g., checked in at the same time) is not as straightforward as extracting features from EP.

To capture the relatedness among places and extract discriminative features from IR, our approach is to build a *network of related places (NRP)*. In an NRP, places are linked based on their relatedness, measured by the information provided in user check-ins through the Random Walk and Restart technique [27]. Upon the NRP, we determine the label probability for each place by exploring the relatedness of places. As such, the label probability derived from IR serves as a feature for classification. Details of feature extraction form IR will be introduced in Section 4.

3.3 Semantic Annotation

After the feature extraction phase, features derived from both EP and IR are used as inputs for the semantic annotation phase to learn a binary SVM for each tag. We choose SVM as the binary classifier because it has shown excellent performance in similar tasks. In our approach, all places are used for each binary SVM training, i.e., an instance labeled with the specific semantic tag under examination is considered as a positive example, while places without this label serve as negative examples. For instance, places tagged *shopping* are positive examples for a classifier for shopping, but negative examples for a classifier for *nightlife*. For a place to be annotated with such a semantic tag, a binary classifier for each tag is expected to classify the place as an instance of the tag class. As a result, the place will be automatically annotated with proper semantic tags.

4. IR FEATURE EXACTION

To facilitate the extraction of features from implicit relatedness among similar places, we develop a new algorithm that builds a *network of related places* (NRP) to capture the relatedness between places, and further derive the label probability as an IR feature for each tag and each place upon the obtained NRP as follows.

4.1 Network of Related Places

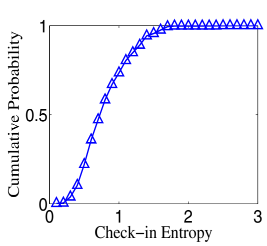
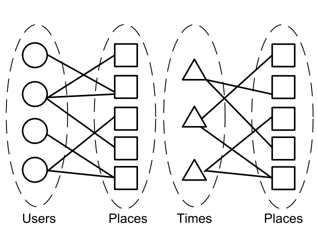


Figure 6: Entropy distribution



(a) UP graph (b) TP graph

Figure 7: Graph representations of LBSN data

As discussed earlier, we intend to exploit the behavior patterns of LBSN users for semantic annotation of places. By analyzing the *Whrrl* dataset, we find that the check-in activities of Whrrl users do exhibit a strong regularity that supports our idea. In the analysis, we study the diversity of places individual users visit by computing the entropy of semantic tags (in eight activity categories) in their check-ins. The result is shown in Figure 6. Smaller entropy indicates that places checked in by LBSN users usually have similar semantic tags. From the figure, we observe that about 22.07% of users have their check-in entropies smaller than 0.5 and about 75% of users have their entropies smaller than 1. In other words, a great number of Whrrl users visit similar places. Therefore, we build a *user-place (UP)* graph, which consists of users and places connected in accordance with the check-in records. Let $c(u_i, p_j, h_s) \in C$ denote a check-in record describing that user u_i has checked in place p_j at time stamp h_s, where C is the collection of all check-in records. Definition 1 gives the formal definition of the UP graph.

Definition 1. ***User-Place (UP) Graph****, denoted by $G_u(V_u, E_u)$, is an undirected bipartite graph (as illustrated in Figure 7(a)). Here $V_u = U \cup P$, where U and P are the sets of all users and places, respectively, and $E_u = \{e_{i,j} | c(u_i, p_j, \cdot) \in C\}$, where $c(u_i, p_j, \cdot)$ denotes that user u_i has visited place p_i at some time. In this graph, each edge $e_{i,j} \in E_u$ is associated with a weight $w_{i,j}$, denoting how often user u_i has visited place p_i. Formally, $w_{i,j} = \Big|\{c(u_i, p_j, h_s)\}\Big|$.*

On the other hand, the timing of check-ins at similar places may be similar. Therefore, we build a *temporal-place (TP)* graph, where the time space is discreterized into twenty-four hours, to capture the similarity between places in the temporal dimension. Definition 2 gives the formal definition of the TP graph.

Definition 2. ***Temporal-Place (TP) Graph****, denoted by $G_t(V_t, E_t)$, is an undirected bipartite graph (as illustrated in Figure 7(b)). Here, $V_t = H \cup P$, where H and P are the sets of all times (i.e. hours) and places, respectively, and $E_t = \{e_{j,s} | c(\cdot, p_j, h_s) \in C\}$, where $c(\cdot, p_j, h_s)$ denotes that a user has visited place p_j at time h_s. In this graph, each edge $e_{j,s} \in E_t$ is associated with a weight $w_{j,s}$, denoting how often p_j has been checked in at time h_s. Formally, $w_{j,s} = \Big|\{c(u_i, p_j, h_s)\}\Big|$.*

In the aforementioned graphs, places are indirectly connected through users and times. In the following, we propose to use the Random Walk and Restart method [27] to estimate the relatedness between pair-wise places in both user and time aspects in order to build a network of related places (NRP), where edges are explicitly established among places according to their relatedness values.

To construct the NRP, we need to derive the relatedness of places from the UP graph and TP graph. In our approach, we first obtain two relatedness values $r^u_{x,y}$ and $r^t_{x,y}$ for every pair of places $p_x, p_y (\in P)$ through Random Walk and Restart (RWR) over the UP and TP graphs, respectively, and then combine them into one relatedness value between place nodes in the NRP. Here we only describe how RWR proceeds on the UP graph since operations on the TP graph are similar. Given a node x, an RWR is performed by randomly following one of its links to another node y of the UP graph based on the transition probabilities of these links, in addition to a probability a to restart at node x. For the UP graph, we prepare a random walk transition matrix that consists of two zero matrices, i.e., user-user matrix (UU) and place-place matrix (PP), and a user-place matrix (actually UP) and its transpose UP^T, where the probability of transiting between a place p_j and a user u_i is proportional to $w_{i,j}$ (in Definition 1). The stationary, or steady-state, probabilities for each pair of nodes can be obtained by recursively processing Random Walk and Restart until convergence. The converged probabilities (i.e., relatedness values) give us the long-term visiting rates from any given node to any other node. In this way, we can calculate the relatedness of all pairs of location nodes, denoted by $r^p_{x,y}(\forall p_x, p_y \in P)$.

Note that the transition matrix for the TP graph can be derived in a similar way. Accordingly, we can obtain two relatedness values $r^u_{x,y}$ and $r^t_{x,y}$ for a pair of places p_x, p_y from the UP and TP graphs. Since both user and time information can help relate the semantic tags of places, we estimate the overall relatedness $r^p_{x,y}$ between place pairs p_x, p_y by integrating them as follows.

$$r^p_{x,y} = \eta r^u_{x,y} + (1-\eta) r^t_{x,y}, \forall p_x, p_y \in P$$

where η is a smoothing factor between 0 and 1. Based on the formula, places checked in by the same user at around the same time show strong relatedness because both relatedness from user and time aspects are considered.

Finally, we build a network of related places (NRP), where each place is connected to places with top-k relatedness values. More specifically, an NRP is defined as follows.

Definition 3. *A network of related places $NRP = \{P, E\}$ is a* directed *graph, consisting of only places. For each place $p_i \in P$, let P^k_i denote the set of top-k related places to p_i. Thus, $E = \{e(x,i) | \forall p_i \in P, p_x \in P^k_i\}$. Here $e(x,i)$ is a directed edge from p_x to p_i.*

4.2 Label Probability Derivation

As mentioned before, in a real LBSN, only some places have tags. The idea of extracting features from IR is to derive descriptive features of a given place from its "related" and tagged places. The network of related places (NRP) is constructed by connecting similar places together, so we aim to infer the tags of a given place by the tags of its neighbors. In order to derive an IR feature for use in the SVM, we derive the probability for a place to be labeled with a given semantic tag from its neighbors. More specifically, the label probability of a place can be estimated from the label probability of its neighbors recursively [16]. Let $\mathcal{N}_i$ be the set of immediate neighbors which have edges pointing to place p_i, and y_i be a variable denoting a tag of place p_i. For all possible tags $t \in T$, we adopt the relaxation labeling method [24] to find the final $Pr(y_i = t|\mathcal{N}_i)$ $(t \in T)$ for each place p_i. Relaxation labeling freezes the current estimations of each p_i so that, at round $n+1$, all places will be updated based on the estimations from round n. As shown below, the label probability of p_i is calculated by considering both the weighted average of the label probabilities of places in $\mathcal{N}_i$, and the current label probability of p_i itself.

$$Pr^{(n+1)}(y_i = t|\mathcal{N}_i) = \beta^{(n+1)}_t \frac{1}{Z} \sum_{p_j \in \mathcal{N}_i} r^p_{j,i} Pr^{(n)}(y_j = t|\mathcal{N}_j) + (1 - \beta^{(n+1)}_t) Pr^{(n)}(y_i = t|\mathcal{N}_i)$$

where $Z = \sum_{p_j \in \mathcal{N}_i} r^p_{j,i}$ is a normalization term and $r^p_{j,i}$ is the relatedness between places p_j and p_i, and $Pr^{(n)}(y_i = t|\mathcal{N}_i)$ denotes the estimation of $Pr(y_i = t|\mathcal{N}_i)$ at round n. Note that, we define the

$$\beta^{(n+1)}_t = \beta^{(n)}_t \alpha,$$

where $\beta^{(0)}_t$ $(t \in T)$ is a constant between 0 and 1, and α is a decay factor, i.e., $0 < \alpha < 1$. Note that in our daily activities, some of them exhibit more regularity than others (e.g., restaurants against shops). Therefore, we employ different $\beta^{(0)}_t$ values for different semantic tags. Note that different tags have different $\beta^{(0)}_t$ values, where label probability calculation with larger $\beta^{(0)}_t$ settings converges slower than the one with smaller $\beta^{(0)}_t$. More importantly, a larger $\beta^{(0)}_t$ implies that the label probability of a given place should be estimated not only according to the immediate neighbors, but also influenced by places in multi-hops away as there are multiple rounds of calculation. A smaller $\beta^{(0)}_t$ suggests that the label probability is only affected by close-by neighbors as there are very few rounds of calculation.

Here, we discuss how to initialize $Pr^{(0)}(y_i = t|\mathcal{N}_i)$ for each $p_i \in P$. Let $P_{\texttt{test}}$ denote the set of testing places, i.e., places that do not have any semantic tags. The label probability of a testing place is initialized as 0.5; while the label probability of a place already labeled with semantic tags is initialized as 1 or 0 according to the labels. Formally, the label probability is initialized as follows.

$$Pr^{(0)}(y_i = t|\mathcal{N}_i) = \begin{cases} 0.5 & \text{if } p_i \in P_{\texttt{test}} \\ 1 & \text{if } p_i \in P - P_{\texttt{test}} \text{ and } t \in T_i \\ 0 & \text{if } p_i \in P - P_{\texttt{test}} \text{ and } t \notin T_i \end{cases}$$

Once we get the label probability estimation for each possible tag on a place p_i, they are treated as IR features for SVM training. Note that features extracted from IR do not consider the explicit patterns exhibited in each individual place. Thus, in our SAP algorithm, we propose to combine features extracted from both EP and IR to address the problem of semantic annotation of places.

5. PERFORMANCE EVALUATION

In this section, we conduct a comprehensive set of experiments to validate our proposed ideas and evaluate our SAP algorithm in terms of three different feature sets: i) features extracted from EP, ii) features extracted from IR, and iii) combination of i) and ii). Here, we use one of the most popular classification toolkits, LIBSVM [5], as the binary SVM classifier. In the following, we first discuss the collected dataset and the preprocessing steps for experiments, then introduce the metrics employed to evaluate the performance, and finally analyze the experiment results.

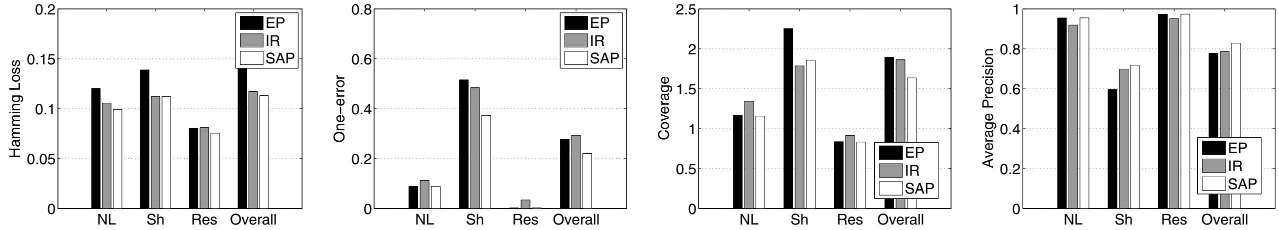
5.1 Dataset Description

We crawled the Whrrl website, a representative LBSN, for a month to collect a dataset consisting of 5,892 users, 53,432 places and 199 types of tags.[5] Among those places, 20% of them are not specified with any semantic tags. In the vocabulary of semantic tags, we find that a lot of tag words sharing the same topic could be grouped in the same category. For example, Pizzerias, Coffee, Bakeries, Snacks, Delis, Cafes, Ice Cream and etc, all belong to the same category, namely, *Restaurant & Food*. Without loss of generality, we build a tag hierarchy based on Yelp[6] to merge those 199 semantic tags into 21 categories to simplify the task of place semantic annotation. We show the top eight major categories and their corresponding percentages in Table 1. As shown, *Restaurants & Food*, *Shopping*, *Nightlife* are the most popular check-in places in Whrrl, i.e., 74% of places are within these three categories. Furthermore, we find that about 33.5% of places belong to multiple categories in our dataset.

In order to conduct the experiments, we pre-process this raw dataset to obtain a ground-truth dataset for performance evaluation. First, places in the ground-truth dataset should have category tags, so we filter out those places without category tags. Next, since we are interested in exploring

[5]Unfortunately, we cannot obtain the check-in time in Foursquare. Thus, we conduct the performance evaluation only upon the whrrl dataset.

[6]http://www.yelp.com



(a) Hamming loss (b) One-error (c) Coverage (d) Average precision

Figure 8: Performance comparison

Category	z%	Category	z%
Restaurants&Food	37%	Hotel & Travel	4%
Shopping	18%	Arts & Entertainment	3%
Nightlife	19%	Health and Medical	2%
Active life	5%	Beauty and Spas	2%

Table 1: Categories and their percentages (z%)

user behaviors, users who have less than 40 check-in records are not included in the ground-truth dataset. Third, we calculate the activity entropies of those users and select the users and their places with entropies less than 0.5 as the dataset to conduct performance evaluation. Moreover, we randomly remove the category tags of $x\%$ places (named testing places, and $x\% = 10\%, 20\%$ and 40% with default value 20%) over the ground-truth dataset. The SAP algorithm is used to recover the category tags for those testing places.

5.2 Performance Metrics

Given a testing place set $P_{\texttt{test}}$, we conduct a performance evaluation by measuring the following four metrics: hamming loss, one-error, coverage and average precision, as they are widely employed in previous multi-label classification studies [25, 29]. Hamming loss aims to measure the accuracy of the predicted tag set against the ground-truth tag set associated with a testing place. The other three metrics concern the ranking of tags annotated by the SAP algorithm, i.e., we consider that SAP performs well when the ground-truth tags are ranked high in the predicted ranked tag list. Note that although we define place semantic annotation as a classification problem, LIBSVM provides probability output [23] (i.e., the probability of the corresponding label), which can be used to rank the semantic tag for each place. Let $Pr(t_x|f_i)$ be the probability output for place p_i being with tag t_x ($\in T$), where f_i denotes the set of features of p_i. According to $Pr(t_x|f_i)$, we get a ranked list of semantic tags, denoted as Y_i, where semantic tags with the highest $Pr(t_x|f_i)$ are ranked at the top.

Hamming loss ($hl_{P_{\texttt{test}}}$): evaluating how many times a place-tag pair is misclassified, i.e., a tag not belonging to the place is predicted or a tag belonging to the place is not predicted. Formally, $hl_{P_{\texttt{test}}} = \frac{1}{|P_{\texttt{test}}|}\sum_{P_i \in P_{\texttt{test}}} \frac{HD(\overrightarrow{T}_i, \overrightarrow{Y}_i)}{|T|}$, where T is the whole tag space, $\overrightarrow{T}_i$ and $\overrightarrow{Y}_i$ are the ground-truth and predicted tag vectors for testing place p_i, and $HD(\overrightarrow{T}_i, \overrightarrow{Y}_i)$ is the hamming distance between $\overrightarrow{T}_i$ and $\overrightarrow{Y}_i$. In the ground-truth tag vector $\overrightarrow{T}_i$ of a place $P_{\texttt{test}}$, the vector element corresponding to a tag t is set to 1 if t is associated with $P_{\texttt{test}}$; otherwise, it is set to 0. The predicted vector $\overrightarrow{Y}_i$ is generated by the SAP algorithm accordingly.

One-error: evaluating how many times the first (or top) ranked predicted tag is not in the ground-truth tag set of the place. Formally, one-error$_{P_{\texttt{test}}} = \frac{1}{|P_{\texttt{test}}|}\sum_{p_i \in P_{\texttt{test}}} f([\arg\max_{t_x \in T} Pr(t_x|f_i)] \notin T_i)$, where for any predicate π, $f(\pi)$ equals 1 if π holds and 0 otherwise.

Coverage: evaluating how far we need, on average, to go down the list of predicted tags (Y_i) in order to recover all the ground-truth tags associated with the place p_i. Let $R(x)$ denote the rank of t_x in the ranked list Y_i generated by SAP. Formally, coverage$_{P_{\texttt{test}}} = \frac{1}{|P_{\texttt{test}}|}\sum_{p_i \in P_{\texttt{test}}} \max_{t_x \in T_i} R(x) - 1$.

Note that one-error and coverage measures are not sufficient for evaluating our SAP algorithm, which may achieve good coverage but suffer high one-error, or vice versa. Thus we introduce the average precision, which takes the ranking positions of all ground-truth tags into consideration, to evaluate the predicted ranked tag list.

Average Precision (AP): Given a place $p_i \in P_{\texttt{test}}$ and a ranked tag list Y_i generated by our SAP algorithm, the average precision for a test place p_i is defined as $AvePrec_i = \sum_{j=1}^{|T_i|} \frac{I(j)(n_j/j)}{|T_i|}$, where $|T_i|$ and n_j denote the total number of ground-truth tags and the number of ground-truth tags before the position $(j+1)$ in the tag list Y_i, respectively, and $I(j)$ is an indicator which takes value 1 if the tag at position j is a ground-truth tag and value 0 otherwise. Therefore, the overall average precision is measured as AP$_{P_{\texttt{test}}} = \frac{1}{|P_{\texttt{test}}|}\sum_{p_i \in P_{\texttt{test}}} AvePrec_i$.

5.3 Experimental Results

As mentioned earlier, we conduct a series of experiments to evaluate the proposed SAP algorithm by comparing three different feature sets. We label the results obtained using features derived from EP and IR by `EP` and `IR`, respectively, and label the results obtained using all features by `SAP`. We also perform sensitivity tests on a number of tuning parameters and different mark-off rates, as well as discreterized and continuous representations of temporal information. Note that we use $\eta = 0.2$, $k = 5$ and β as listed in Table 2 as the default parameter settings throughout the experiment.[7]

Category	β	Category	β
Restaurants&Food	0.9	Hotel & Travel	0.3
Shopping	0.1	Arts & Entertainment	0.1
Nightlife	0.9	Health and Medical	0.1
Active life	0.1	Beauty and Spas	0.1

Table 2: Optimal β settings under $\eta = 0.2$ and $k = 5$

5.3.1 Overall Performance

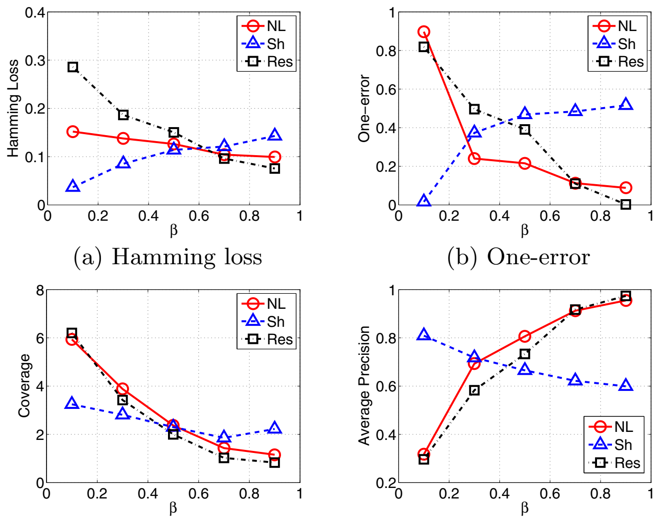
In order to evaluate the performance of our SAP algorithm in detail, we not only show the performance over the entire dataset (labeled with `overall` as show in Figure 8), but also the performance for subsets of testing places in the same category (according to the ground-truth). More specifically, we show three categories of places: *Restaurants & Food* (labeled with `Res`), *Nightlife* (labeled with `NL`) and *Shopping* (labeled with `Sh`). These categories were chosen since they constitute the majority (i.e., about 74%) of all places. As shown in Figure 8, under the default setting,

[7] β refers to $\beta_t^{(0)}$ in this experiment.

SAP shows the best performance consistently, while both EP and IR also demonstrate good strength for the task of semantic annotation in LBSNs. Note that EP performs better than IR for the places in the groups of Res and NL, particularly for the performance metrics one-error, coverage and average precision. The reason is that most people have the same routine for activities in those categories. As a result, those activities have very distinctive characteristics, such as the distribution of check-in times extracted from EP. Thus, EP is able to tag these kinds of places very well. On the other hand, IR shows great strength in labeling places with shopping tags. Regularity of shopping activities of individuals helps to discover the shopping places from other related shopping places, although different people may go shopping at different times.

5.3.2 Tuning Parameters

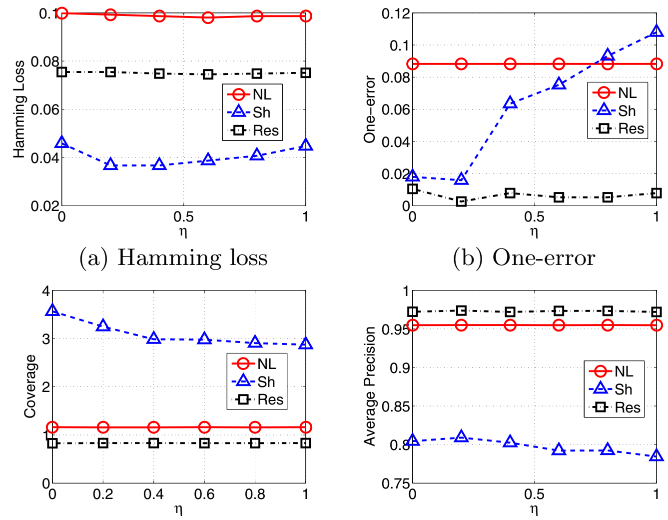
Next, we test the impact of tuning parameters, including β, η and k, on classification performance of SAP for Res, NL and Sh. Notice that the impact of each tuning parameters is tested by fixing all other parameters in default settings.



(a) Hamming loss (b) One-error (c) Coverage (d) Average precision

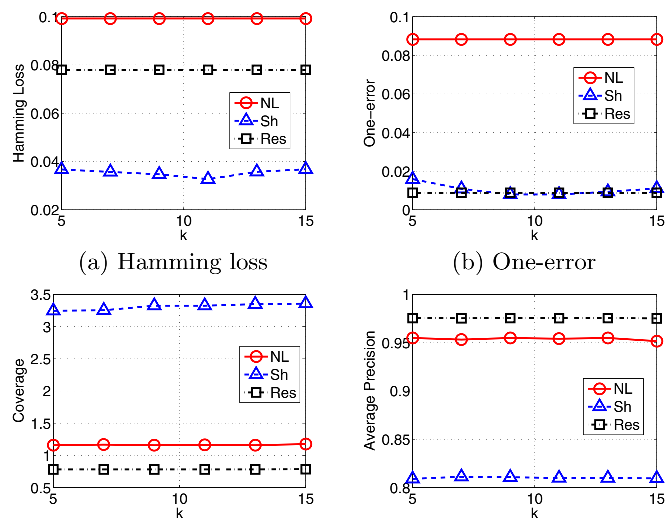
Figure 9: Impact of β

As mentioned earlier, the optimal settings for the classifiers for different categories are different as shown in Table 2. Note that β tunes the influence from immediate neighboring places and places in multiple hops away. Here, we check the impact of β on the performance of classification, with particular interests in Res, NL and Sh. As shown in Figure 9, β has significant impact on the classification performance for all categories. Both Res and NL show very similar behavior with the variation of β. The best performance setting of β for Res and NL is 0.9, implying that the the label probability of a given place should be estimated not only according to its immediate neighbors but also the places in multiple hops away. The reason is that in an NRP, Res (NL) places are clustered together, thus a larger β can provide more robust and accurate estimation. On the other hand, the best β setting of Sh is 0.1, indicating that the label probability estimation of shopping places is very sensitive to the influence from their neighbors. A smaller β suggests that the label probability of a given place is only affected by immediate neighbors. We find that Sh places are usually not clustered as well as Res because the regularity of Sh activities are not as regular as Res activities. Thus, it is better to only use information from immediate neighbors to estimate label probability for Sh places.



(a) Hamming loss (b) One-error (c) Coverage (d) Average precision

Figure 10: Impact of η



(a) Hamming loss (b) One-error (c) Coverage (d) Average precision

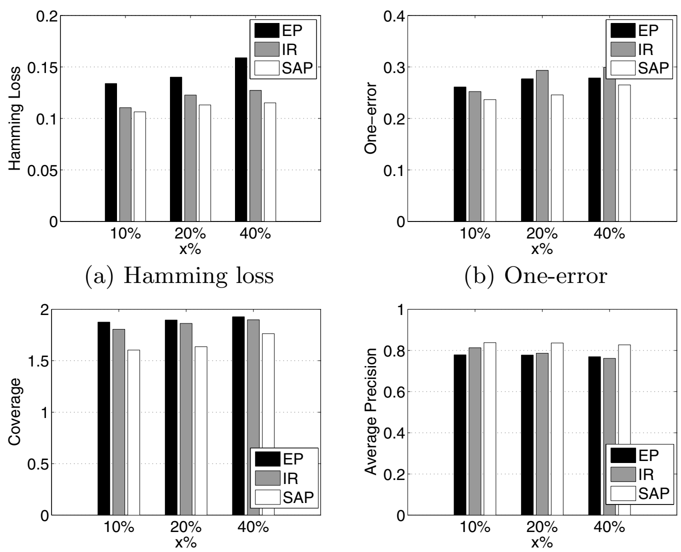
Figure 11: Impact of k

In Figure 10, we show our test on the parameter η, which is used to tune the weight of place relatedness values computed from user and time aspects. As shown, we find the impact of η on classification of Res and NL places is very limited. A possible reason is that places in both Res and NL categories are well clustered according to either common users or common time. Another reason is that the default β setting for Res and NL is 0.9, which means influence from places even in multiple hops away is contributing to accurate estimation of label probability. The selection of immediate neighbors is not that sensitive to η as long as places in those categories are clustered together. Nevertheless, η does affect the performance for Sh places, as label probabilities of Sh places are mostly affected by immediate neighbors (i.e., $\beta = 0.1$ for Sh activities). As shown in Figure 10, when $\eta = 0.2$, SAP shows the best performance on classification of Sh places in terms of the metrics of hamming loss, one error and average precision; when $\eta = 0.9$, the coverage performance turns out to be the best for Sh places. Accordingly, we consider $\eta = 0.2$ as a proper parameter setting and use it as the default setting throughout the experiment. It implies that both user and time are important to discover similar places through user behavior, particularly for the classification of Sh places. Besides, as the majority of check-ins for a visitor are usually Res places, time information plays an important role to link similar Sh places through the regular behaviors of people.

We further test the tuning parameter k in Figure 11, where k determines the number of neighboring places for a given place in an NRP. Similarly, we find that the variation of k has almost no impact on the classification performance of Res and NL places since places in both categories are clustered together. However, the selection of k affects the performance of classification for Sh places. In Figure 11, the best setting of k for the classification of Sh places is different for various performance metrics. Nevertheless, we find that k should be set to a proper value, in order to avoid the noise introduced by a large number of neighboring places.

5.3.3 Test on Mark-off Rate

Here, we investigate the impact of different mark-off rates to the performance of EP, IR and SAP. As shown in Figure 12, the performance of algorithms with different feature sets all degrade to some extent as the mark-off rate increases. Nevertheless, SAP shows the best performance consistently over all mark-off rates as it includes all the features.



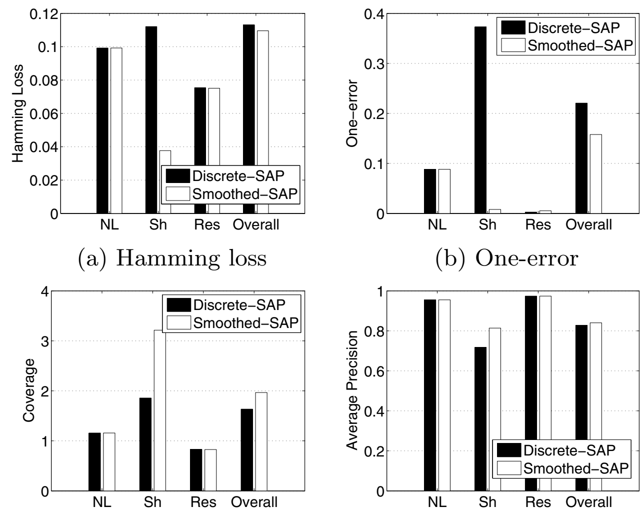
(a) Hamming loss (b) One-error

(c) Coverage (d) Average precision

Figure 12: Impact of mark-off rate, $x\%$

5.3.4 Test on Continuity of Time

Check-in time is continuous in the temporal dimension, even though we simplify it as discrete twenty-four hours in the initial design of TP graph. Here, we investigate the impact of the continuity of time on the performance of SAP. In order to capture the continuity of time, we propose a method to smooth hours following the intuition that a user who checks in a place at time h_s, would probably check in similar places around the times h_{s-1} and h_{s+1}, where h_{s-1} and h_{s+1} are adjacent times to h_s. More specifically, for each check-in at place p_j and time h_s, we establish additional m edges to the m most adjacent time nodes beside the time node h_s during the TP graph construction. For example, if h_s presents 20:00 and $m = 1$, we establish edges from the place to the time nodes 19:00 and 21:00, in addition to 20:00 in the construction of the TP graph.



(a) Hamming loss (b) One-error

(c) Coverage (d) Average precision

Figure 13: Discrete Hours Vs. Smoothed Hours

Finally, we test the SAP algorithm, with $m = 1$ and β, η and k following the aforementioned default settings. The SAP algorithm following the initial design is denoted as Discrete-SAP, while the SAP algorithm with smoothed-hour TP graph design denoted by Smoothed-SAP. As shown in Figure 13, the impact on classification of Res and NL places are marginal, since places in those categories have been clustered together with Discrete-SAP. However, considering continuity of time does help improve the classification performance for Sh places, as shopping places checked in around the same time period (although in different hours) are possibly discovered as neighboring places in an NRP.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we investigate the place semantic annotation problem, which aims to automatically annotate all places with semantic tags in location-based social networks. Such tags are a crucial pre-requisite for location search, recommendation services, or data cleaning. In order to tackle this problem, we propose a novel semantic annotation algorithm which learns a binary SVM for each tag. Based on the check-in behavior of users, we extract features of places from two aspects: explicit pattern (EP) at individual places and implicit relatedness (IR) among similar places. Specifically, we extract EP features by aggregating user check-in behaviors to the corresponding places and extract IR features by exploiting the place relatedness exhibited by regularity of user behavior. Finally, we conduct a comprehensive experimental study based on a real dataset collected from *Whrrl*. The results demonstrate the suitability of our approach and also support the assumption that both EP and IR need to be taken into account. Particularly, most people follow the same and distinctive pattern to visit restaurants and nightlife places. Thus, features extracted from EP hold very powerful discriminative capability. On the other hand, against EP, features from IR are excellent for tagging places related to shopping because some individuals exhibit strong patterns in certain shopping activities.

Through our analysis on the Whrrl dataset, we find some semantic tags usually co-occur, e.g., restaurant and bars. In the future, we plan to explore the correlation among semantic tags for the semantic annotation of places. In addition, we plan to include some alternative approaches (e.g., [22]) for comparison and to use multiple large-scale datasets to validate our proposed SAP algorithm.

7. REFERENCES

[1] D. Ashbrook and T. Starner. Using gps to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003.

[2] R. Baeza-Yztes and B.Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.

[3] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37(9):1757–1771, 2004.

[4] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 307–318, 1998.

[5] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001.

[6] A. Clare and R. D. King. Knowledge Discovery in Multi-label Phenotype Data. In *European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 42–53, 2001.

[7] N. Eagle and A. S. Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[8] Y. Freund and R. E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences (JCSS)*, 55(1):119–139, 1997.

[9] B. Gallagher, H. Tong, T. Eliassi-Rad, and C. Faloutsos. Using ghost edges for classification in sparsely labeled networks. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*, pages 256–264, 2008.

[10] J. Hightower and G. Borriello. Particle filters for location estimation in ubiquitous computing: A case study. In *ACM International Conference on Ubiquitous Computing (Ubicomp)*, pages 88–106, 2004.

[11] T. Joachims. Text categorization with suport vector machines: Learning with many relevant features. In *European Conference on Machine Learning (ECML)*, pages 137–142, 1998.

[12] C. Keßler, K. Janowicz, and M. Bishr. An agenda for the next generation gazetteer: geographic information contribution and retrieval. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (GIS)*, pages 91–100, 2009.

[13] T. Li and M. Ogihara. Detecting emotion in music. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2003.

[14] L. Liao, D. J. Patterson, D. Fox, and H. A. Kautz. Learning and inferring transportation routines. *Artifical Intelligence*, 171(5-6):311–331, 2007.

[15] J. Lin, G. Xiang, J. I. Hong, and N. M. Sadeh. Modeling people's place naming preferences in location sharing. In *ACM International Conference on Ubiquitous Computing (UbiComp)*, pages 75–84, 2010.

[16] S. A. Macskassy. Improving learning in networked data by combining explicit and mined links. In *Conference on Artificial Intelligence (AAAI)*, pages 590–595, 2007.

[17] S. A. Macskassy and F. Provost. A simple relational classifier. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD) -Workshop (MRDM)*, pages 64–76, 2003.

[18] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research (JMLR)*, 8:935–983, 2007.

[19] A. K. McCallum. Multi-label text classification with a mixture model trained by em. In *Conference on Artificial Intelligence (AAAI) - Workshop on Text Learning*, 1999.

[20] L. McDowell, K. M. Gupta, and D. W. Aha. Cautious inference in collective classification. In *Conference on Artificial Intelligence (AAAI)*, pages 596–601, 2007.

[21] E. Moxley, J. Kleban, and B. S. Manjunath. Spirittagger: a geo-aware tag suggestion tool mined from flickr. In *ACM international conference on Multimedia information retrieval (MIR)*, pages 24–30, 2008.

[22] C. Mülligann, K. Janowicz, M. Ye, and W.-C. Lee. Analyzing spatial-semantic interaction of points of interest in volunteered geographic information. In *International Conference on Spatial Information Theory (COSIT) (to appear)*, 2011.

[23] J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advance in Large Margin Classifiers*, pages 61–74. MIT Press, 1999.

[24] A. Rosenfeld, R. Hummel, and S. Zucker. Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man, and Cybernetics*, 6:420–433, 1976.

[25] R. E. Schapire and Y. Singer. Boostexter: A boosting-based systemfor text categorization. *Machine Learning*, 39:135–168, 2000.

[26] B. Taskar, E. Segal, and D. Koller. Probabilistic Classification and Clustering in Relational Data. In *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 870–878, 2001.

[27] H. Tong, C. Faloutsos, and J.-Y. Pan. Fast random walk with restart and its applications. In *Industrial Conference on Data Mining (ICDM)*, pages 613–622, 2006.

[28] M. Ye, P. Yin, W.-C. Lee, and D. L. Lee. Exploiting Geographical Influence for Collaborative Point-of-Interest Recommendation. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR) (to appear)*, 2011.

[29] M.-L. Zhang and Z.-H. Zhou. A k-nearest neighbor based algorithm for multi-label classification. In *IEEE International Conference on Granular Computing (GrC)*, pages 718–721, 2005.