

Investigating the Utility of the Weather Context for Point of Interest Recommendations

Christoph Trattner · Alexander Oberegger ·
Leandro Marinho · Denis Parra

Received: date / Accepted: date

Abstract Point of Interest (POI) recommender systems for location-based social networks, such as Foursquare or Yelp, have gained tremendous popularity in the past few years. Much work has been dedicated to improving recommendation services in such systems by integrating different features (e.g., time or geographic location) that are assumed to have an impact on people's choices for POIs. Yet, little effort has been made to incorporate or even understand the impact of weather on user decisions regarding certain POIs. In this paper, we contribute to this area of research by presenting the novel results of a study that aims to recommend POIs based on weather data. To this end, we have expanded the state-of-the-art Rank-GeoFM POI recommender algorithm to include additional weather-related features such as temperature, cloud cover, humidity and precipitation intensity. We show that using weather data not only significantly improves the recommendation accuracy in comparison to the original method, but also outperforms its time-based variant. Furthermore, we investigate the magnitude of the impact of each feature on the recommendation quality. Our research clearly shows the need to study weather context in more detail in light of POI recommendation systems. This study is relevant for researchers working on recommender sys-

tems in general, but in particular for researchers and system engineers working on POI recommender systems in the tourism domain.

Keywords POI Recommender Systems; Location-based Services; Weather Context

1 Introduction

Location-based social networks (LBSNs) are online services that enable users to check-in and share places and relevant content such as photos, tips and comments with other users. This may help other users to explore novel and interesting places that they may not have been to before. LBSNs are now an essential tool for any tourist or traveler around the world who goes on holiday to explore a city, interesting sights or places [3, 12]. Foursquare¹, for example, is a popular LBSN with millions of subscribers making millions of check-ins everyday all over the world². This vast amount of check-in data, publicly available through the data access API of Foursquare, has inspired many researchers to investigate human mobility patterns with the aim of assisting users by means of personalized POI (point of interest) recommendation services in order to find new, interesting and relevant places [31, 32].

Most of the current approaches to POI recommendation exploit three main aspects of the data, namely, social, time and geo-location [7, 18, 31]. Although these approaches work reasonably well, surprisingly little attention has yet been paid to weather, a factor (aka context) that plays an important role in our daily activities [14] and in areas such as tourism [5, 6]. As such,

C. Trattner
University of Bergen, Norway &
MODUL University Vienna, Austria
E-mail: trattner.christoph@gmail.com

A. Oberegger
Graz University of Technology, Austria

L. Marinho
Universidade Federal de Campina Grande, Brasil

D. Parra
Pontificia Universidad Católica de Chile, Chile

¹ <https://Foursquare.com/>

² <https://Foursquare.com/infographics/10million>

weather also plays a potentially important role in the context of POI recommendation services. For example, if it is raining at a certain place and time, people may prefer to visit indoor POIs rather than outdoor POIs. Thus, not taking the weather into consideration may lead to recommendations that will very likely displease users.

In this paper, which is an extended and updated version of a paper [26] presented at the Workshop on Recommenders in Tourism (RecTour) 2016 [9], we focus on investigating the utility of weather context in order to improve the current state-of-the-art in POI recommender systems. To drive our research, the following four research questions were defined:

- **RQ1.** To what extent does the weather context impact the check-in behaviour of users in Foursquare?
- **RQ2.** Is it possible to extend and improve a current state-of-the-art POI recommendation model by exploiting weather features?
- **RQ3.** How does the model perform in comparison to other methods and which of the weather features investigated provides the highest accuracy gain?

To answer these questions, we take the following steps: (i) we analyze check-in data from Foursquare, one of the largest location-based social network services on the Web, and show that weather may indeed have an impact on user check-in behaviour; (ii) we introduce a weather-aware POI recommender model (WPOI) that is based on an existing POI recommendation approach considered to be state-of-the-art in the field; and (iii) we perform a set of offline experiments in four different cities in the US to evaluate how our approach performs in comparison to the original model and other state-of-the-art recommender approaches. This work contributes to the tourism recommender systems literature by investigating in detail the extent to which weather features impact user check-in behaviour on Foursquare and how these features perform in the context of a POI recommender system.

The paper is structured as follows: in Section 2 we highlight relevant related work in the field, while Section 3 introduces the dataset we used for our study. Subsequently, Section 4 presents results from our empirical analysis, while Section 5 introduces WPOI, a weather-aware POI recommender method based on Rank-GeoFM. Thereafter, Section 6 shows how the model performs compared to other POI recommender methods. Finally, Section 7 discusses the findings and limitations and Section 8 concludes the paper with a summary of our main findings.

2 Related Work

With the advent of LBSNs, POI recommendation rapidly became an active area of research within the recommender systems, machine learning and Geographic Information Systems research communities [2]. Most of the existing research in this area exploits some sort of combination between some (or all) of the following data sources: check-in history, social relations (e.g., friendship relations), time and geographic positions [1, 7, 8, 11, 18, 23, 31]. While these different sources of data, also called context information, affect user decisions about visiting a POI in different ways, weather data, which, according to common sense and related studies, may have a great influence on this decision, is still rarely used.

One example of such an investigation is the work of Horanont et al. [14], who conducted an analysis of GPS traces of 31, 855 mobile users in the grand Tokyo area (collected between 2010-2011). In particular, they studied patterns of places that users visited and the amount of time that they spent there, correlating these patterns to different weather conditions. Among other things, they found that temperature had a significant effect on how long people spent at a certain location. In detail, they found that people were more likely to stay longer at restaurants, eateries or other food stores as well as retail and shopping areas when the weather was very cold or when conditions were calm (non-windy). They also found that the effect of weather on daily activities was not uniformly distributed across different geographical areas. Accessibility to trains severely affected people’s choice of activities, since the farther the person lived from a train station, the more particular weather conditions affected their activities.

Another interesting aspect of the effect of weather on human activities is related to human mobility under different types of disasters. Different studies [27–29] have examined the perturbations caused in human mobility by disasters such as typhoons, winter storms and thunderstorms. Interestingly, most of them show that the distribution of human mobility (in terms of travel distances) under these circumstances keeps being power-law or truncated power-law. In the case of the study of mobility during Hurricane Sandy in New York City in 2012 [27], for example, Wang and Taylor showed that the people’s trajectories during a disaster (measured by center of mass and radius of gyration) were indeed affected by the disaster, but remained highly correlated with their normal trajectories.

Using weather information to personalize services or systems has been studied in the past, however, these studies lack details on evaluation. Martin et al. [20] pro-

poses a mobile application architecture that considers the use of weather data to personalize a geo-coding mobile service. However, the quality of the approach is not evident from their work since no evaluation is presented. A similar contribution was made by Meehan et al. [21], who proposes a hybrid recommender system based on time, weather and media sentiment when introducing the VISIT mobile tourism recommender. Unfortunately, again, neither implementation details the approach, nor is an evaluation presented in their work.

Among the few works that have actually used weather features in the recommendation pipeline and also presented an evaluation of the system are the two papers of Braunhofer et al. [5,6], who introduces a recommender system designed to run in mobile applications for recommending touristic POIs in Italy. The authors conducted an online study with 54 users and found that recommendations that take weather information into consideration were indeed able to increase user satisfaction. Compared to this study, our implementation is based on a more recent state-of-the-art POI recommendation model. Furthermore, we provide details about which weather features contribute most to the model performance and also provide results of an exploratory data analysis that shows to what extent weather affects human check-in behaviour in cities.

In summary, we know little about how weather features could be useful in the context of POI recommender systems. The current state-of-the-art mostly exploits check-in history, social relations, time and geo-locations. Moreover, we know little about which weather features can be exploited in a POI recommender scenario or how much they can improve the accuracy of a state-of-the-art POI recommendation model. This is exactly the gap the present article is trying to fill by (i) presenting a weather-aware POI recommender model and (ii) a large-scale offline study showing how POI recommenders can profit from weather data.

3 Dataset

To address our research questions, we made use of a Foursquare dataset that was obtained from Yang et al. [30]. The dataset was crawled from April 2012 to September 2013 and comprises over 33 million check-ins from over 200,000 users all over the world. In detail, the dataset contains the following information: User ID (anonymized), Venue (Latitude, Longitude, Foursquare Category, Country code) and UTC check-in time. In addition to this, an extra table is available in the dataset that contains, for POIs of type ‘city’, the following information: City name, Latitude (of City center), Longi-

Table 1: Basic statistics of the dataset used in the empirical data analysis.

Statistic	Original	Pruned
Num. Check-ins	33,278,683	3,545,288
Num. POIs	3,680,126	501,415
Num. Users	266,909	50,812
Num. Cities	415	60
Num. Countries	77	1

tude (of City center), Country code, Country name and City type (e.g., national capital, provincial capital).

For the purpose of our research, which deals with the problem of recommending useful POIs to people in cities based on the current weather context, we selected only POIs in cities and only those present in the US. This reduced the original dataset to approximately 3 million check-ins across 60 cities made by around 50,000 users in 765 distinct categories. Table 1 provides an overview of the basic statistics in our dataset.

Concerning the process of collecting weather context information, the weather API named forecast.io (recently renamed to Dark Sky API³) was used. For the purposes of our research, we obtained eight weather context features provided by the API, namely, ‘cloud cover’, ‘visibility’, ‘moonphase’, ‘precipitation intensity’, ‘pressure’, ‘temperature’, ‘humidity’ and ‘wind speed’. This kind of information was collected for all (time, POI) tuples present in our dataset. Table 2 provides a more detailed overview of the weather data used in our study and their ranges according to the check-ins in the Foursquare dataset.

In order to query the weather information of a certain POI at (latitude, longitude, time), the following API call can be used

`https://api.darksky.net/.../[latitude],[longitude],[time]`

, wherein one can distinguish between the forecast and the actual weather conditions at that certain place and time. For our study, we relied on actual weather conditions rather than on forecast data. One might argue that this is a limitation of this work, as people often plan their trips in advance by looking at the weather forecast for a particular day. However, as is also well known, forecasts are not always reliable and weather conditions can change rapidly. As such, people need to adjust their plans accordingly and, thus, other alternatives may be considered. Investigating in more detail how long in advance people plan their trips to cities is an interesting research question, however, it is beyond the scope of this paper. The idea of this study is to

³ <https://darksky.net/dev>

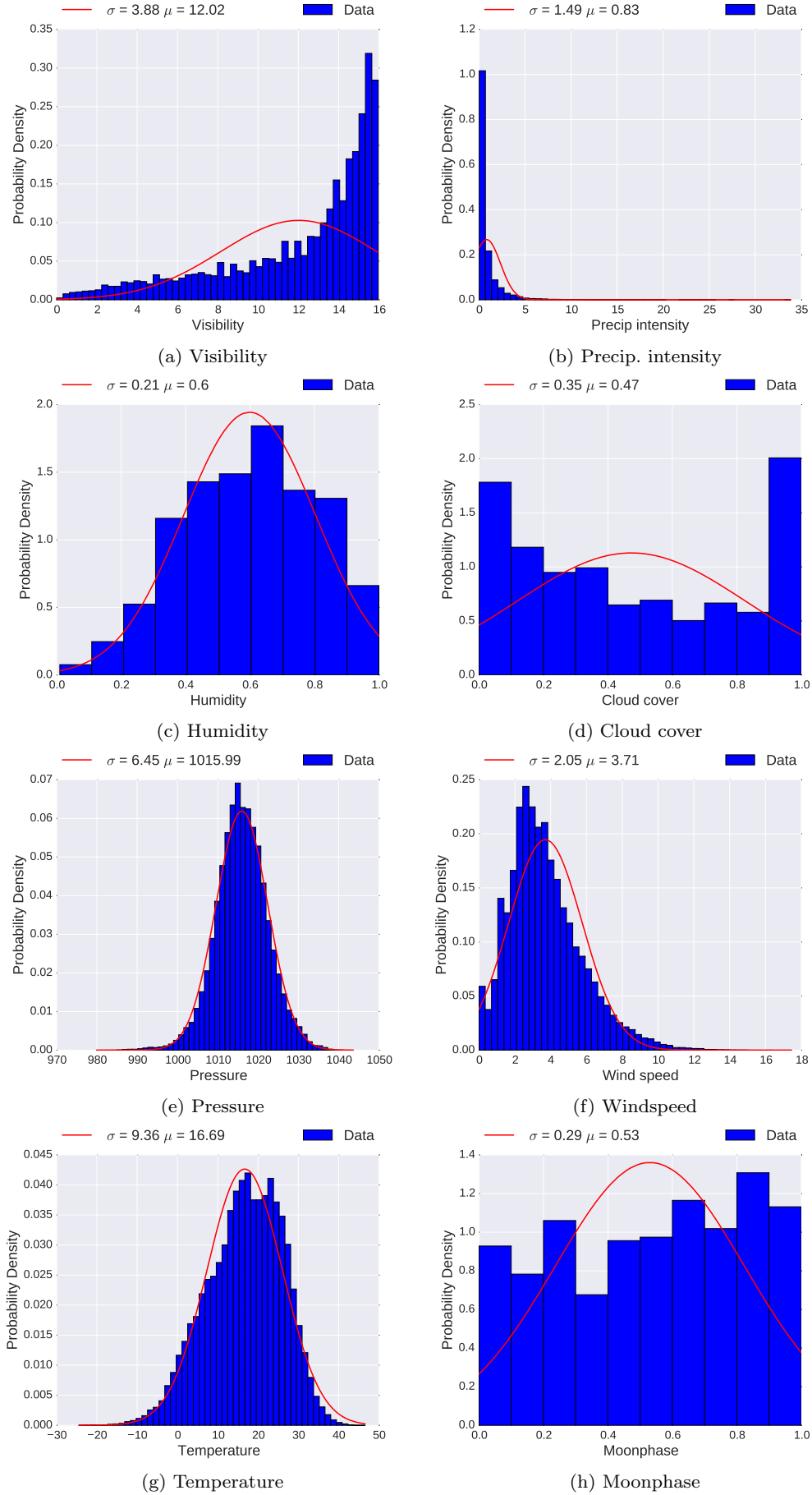


Fig. 1: Check-in distributions over the eight weather features. As shown, there are observable differences between the distributions. While some closely resemble a normal distribution (see ‘pressure’ for instance), others are more skewed or appear to be more uniformly distributed (see ‘moonphase’ for instance).

Table 2: The different weather features used in this work and their properties.

Feature	Properties	Range (Min. - Max.)
Visibility	Value representing the average visibility in kilometers capped at 16,09.	0km – 16,09km
Precip. intensity	Precipitation intensity measured in milimeters of liquid water/hour.	0mm/h – 34,29mm/h
Humidity	Value between 0 and 1 representing the “Percentage relative humidity” is defined as the partial pressure of water vapor in air divided by the vapor pressure of water at the given temperature.	0,02φ – 1,00φ
Cloud cover	Value between 0 and 1 displaying the percentage of the sky covered by clouds.	0 – 1
Pressure	Atmospheric pressure measured in hectopascals.	957,11hPa – 1046,05hPa
Wind speed	Wind speed measured in meters/second.	0m/s – 19,13m/s.
Temperature	Temperature measured in degree Celsius.	–24,48° – 46,58°
Moonphase	Value from 0 to 1 representing the range between new moon and full moon	0 – 1

Table 3: Basic statistics (Mean, Median, 1st and 3rd Quartile) of the eight weather features investigated as shown in Figure 1.

Feature	Mean	Median	1 st Quartile	3 rd Quartile
Visibility	14.87	16.09	15.03	16.09
Precip. intensity	0.08	0	0	0
Humidity	0.65	0.67	0.49	0.82
Cloud cover	0.44	0.31	0.05	0.75
Pressure	1016.20	1016.11	1012.2	1020.21
Windspeed	3.31	2.99	2.02	4.28
Temperature	16.26	17.48	9.34	23.84
Moonphase	0.52	0.53	0.26	0.78

show that there is a correlation between weather context and user check-in behaviour and that this signal can be used for the purpose of improving the quality of POI recommendations for people considering the actual weather conditions of a POI.

4 RQ1: Impact of the Weather Context on the User Check-in Behaviour

To show the extent to which weather features might have an impact on user check-in behaviour, a set of exploratory data analyses were performed. Within this section we present an analysis of: a) the distribution of weather features; b) the correlation between weather features; c) the distance distribution of pairs of check-ins in respect to different weather contexts; d) check-in distribution by POI category in respect to different weather contexts; and e) city-wise weather features.

Distribution Analysis of Weather Features. Figure 1 presents the distribution of check-ins for each of the eight weather features analyzed. As one can see from the plots, the respective distribution for ‘pressure’, ‘temperature’, ‘hu-

midity’ and ‘wind speed’ resembles a normal distribution (see also the red-coloured curves that represent a superimposed Gaussian distribution over the empirical distribution). Unlike these four features, the distribution of ‘precipitation intensity’ is very skewed, showing that users have a strong preference towards checking into places where the ‘precipitation intensity’ is low (i.e., it is not raining). The opposite is observed for ‘visibility’, where users check-in more when visibility is high. This makes sense considering that precipitation (as well as cloud cover) diminishes visibility. The weather feature ‘cloud cover’ has an interesting distribution, where check-ins are more likely where very low and very high values occur at the same time, especially considering that different POIs are more or less popular depending on the season (see Figure 2, showing the seasonal check-in probabilities of some popular and representative sub-categories out of the total of 765 distinct categories present in the Foursquare dataset). Finally, ‘moonphase’ displays, comparatively, a more uniform distribution than the other features, although marginal differences can still be observed between the minimum and maximum values.

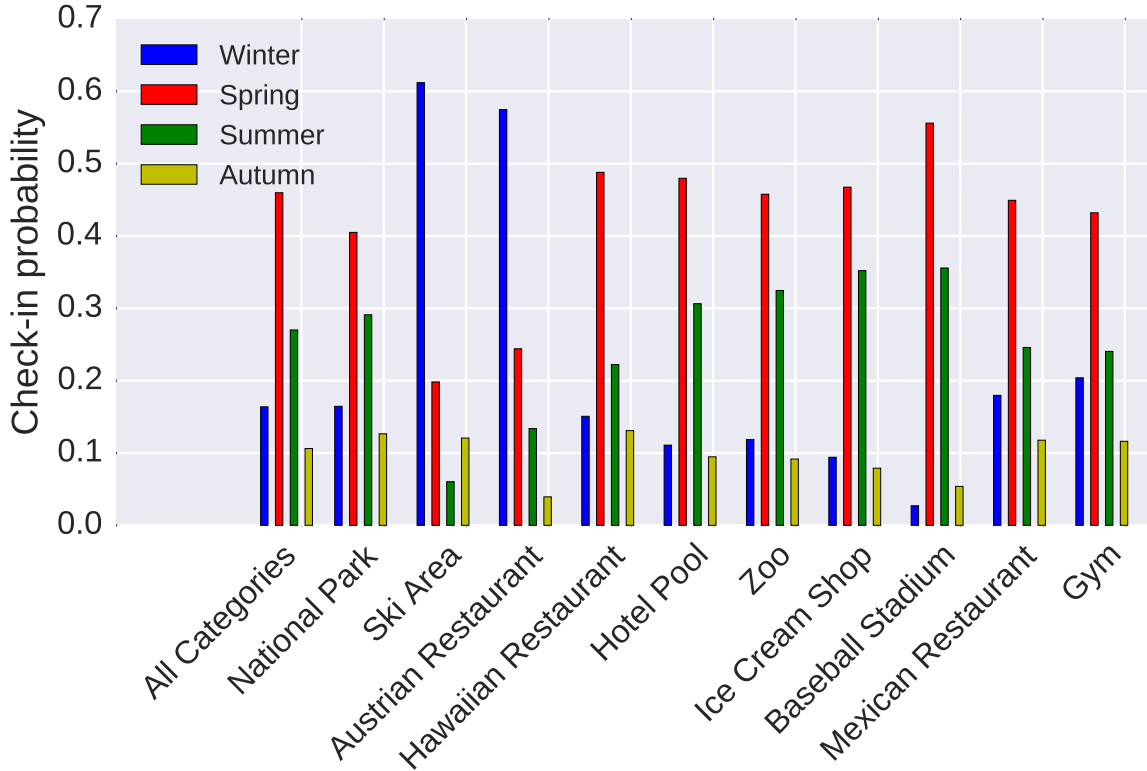


Fig. 2: Seasonal check-in probabilities of some popular and representative sub-categories. “All Categories” is used as a ground truth and displays the probabilities of the whole Foursquare dataset. As shown in the “All Categories” histogram, most check-ins happen during spring time over the course of a year. Furthermore, the plot reveals some interesting check-in patterns for categories such as “Ski Area” and “Austrian Restaurant” that are significantly more popular during winter time than the rest of the year.

The exact mean, median, minimum and maximum values for these eight weather features/distributions can be found in Table 2 and Table 3.

Pairwise Correlation Analysis of Weather Features.

Evaluating a recommender system is typically a computationally intensive task, especially when huge amounts of data points and features need to be considered at the same time. It is, therefore, advisable to begin with building and evaluating a recommender system model that simultaneously operates on a smaller feature set and fewer data points. According to Hall [13], “A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other.” That means that if there are two features that highly correlate with each other, it is possible to eliminate one of them, because there is no additional information added when keeping both of them. Furthermore, some classifiers such as the Naïve Bayes decrease their performance when

the features are correlated. For this experiment, a pairwise Pearson correlation (ρ) was calculated between the eight weather features.

Figure 3 shows some very reasonable correlations between the investigated features. For instance, there is a strong negative correlation ($\rho = -0.77$) between ‘visibility’ and ‘humidity’ that describes the natural weather phenomenon in which high humidity blurs visibility. The negative correlation ($\rho = -0.67$) between ‘cloud cover’ and ‘visibility’ also shows the negative impact of clouds on visibility. Of particular interest is the negative correlation between relative ‘humidity’ and ‘temperature’ ($\rho = -0.99$), which reflects the fact that relative humidity represents the saturation of moisture in the air and cold air does not need that much moisture to be saturated. Yet another interesting high positive correlation exists between ‘windspeed’ and ‘temperature’ ($\rho = 0.93$), which leads to the assumption that the foehn effect took place, culminating in a concurrent high wind speed and high temperatures. Finally,

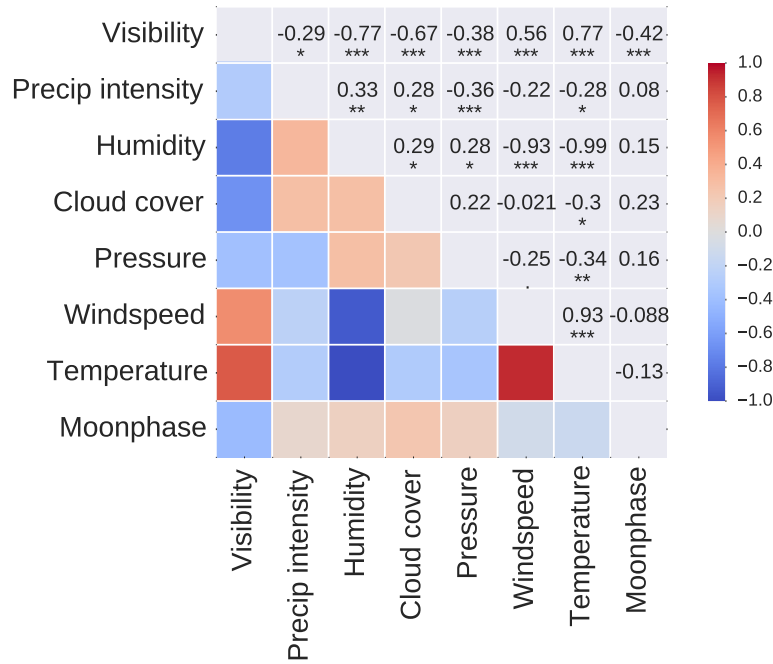


Fig. 3: Correlation matrix (Pearson) for the eight weather features (* $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$). The lower left of the plot shows on a general level to what extent the features are correlated, while the upper-right part shows actual numbers and the asterisks show whether this effect is also significant. The gradient bar to the right explains the color schema and the magnitude of the correlation ranking from dark blue ($-1.0 = 100\%$ negatively correlated) to dark red ($1.0 = 100\%$ positively correlated).

we observe a strong negative correlation between the ‘moonphase’ and the ‘visibility’ features ($\rho = -0.42$), which we did not expect to find. In recent research, Kohyama et al. [16] found a relationship between the ‘moonphase’, ‘tidal variation’, ‘humidity’ and ‘rainfall’. In particular, by extensive analysis of 15 years of weather data by NASA and Japan Aerospace Agency, they found that when the moon is overhead, its gravitational force attracts Earth’s atmosphere, which increases pressure (atmosphere’s weight) in that part of the planet. This phenomenon then increases temperature and, consequently, the capacity to hold moisture goes up. In his own words, Kohyama explains: “Relative humidity affects rain, because lower humidity is less favorable for precipitation”. Although further analysis should be performed to establish a link between our study and theirs, it might be the case that the ‘moonphase’ feature will turn out to be more important than expected as a contextual variable for POI recommendation.

In summary, this correlation analysis provides several insights that may be useful for informing the design of effective techniques for fully exploiting weather information in POI recommender systems.

Check-in Distance Analysis. In order to find out whether or not weather also has an impact on people’s travels, we performed an experiment in which we had a closer look at the travel distance distributions regarding different weather characteristics. As shown in our previous work [19], travel distance is an important feature in a location recommendation scenario. Typically, people prefer to go to places that are closer to their current location or to those previously checked into than those farther away. For this reason, we calculated the geographic distances of any pair of check-ins for each user in our experimental dataset and plotted the distributions on a log-log scale with a maximum distance between two POIs of 100km.

Figure 4 shows the results of this analysis by filtering the distributions in terms of lower and higher weather feature values (above or below the mean) in the target POI. As presented, the plots follow a power law distribution in the range between 1 and 10 kilometres (i.e., the distributions follow a straight line).

What is also highlighted in the plots is that there are features such as ‘moonphase’ or ‘pressure’ that do not show observable differences between the two applied filters, while the rest of the features do show greater differences in distribution. This indicates that these fea-

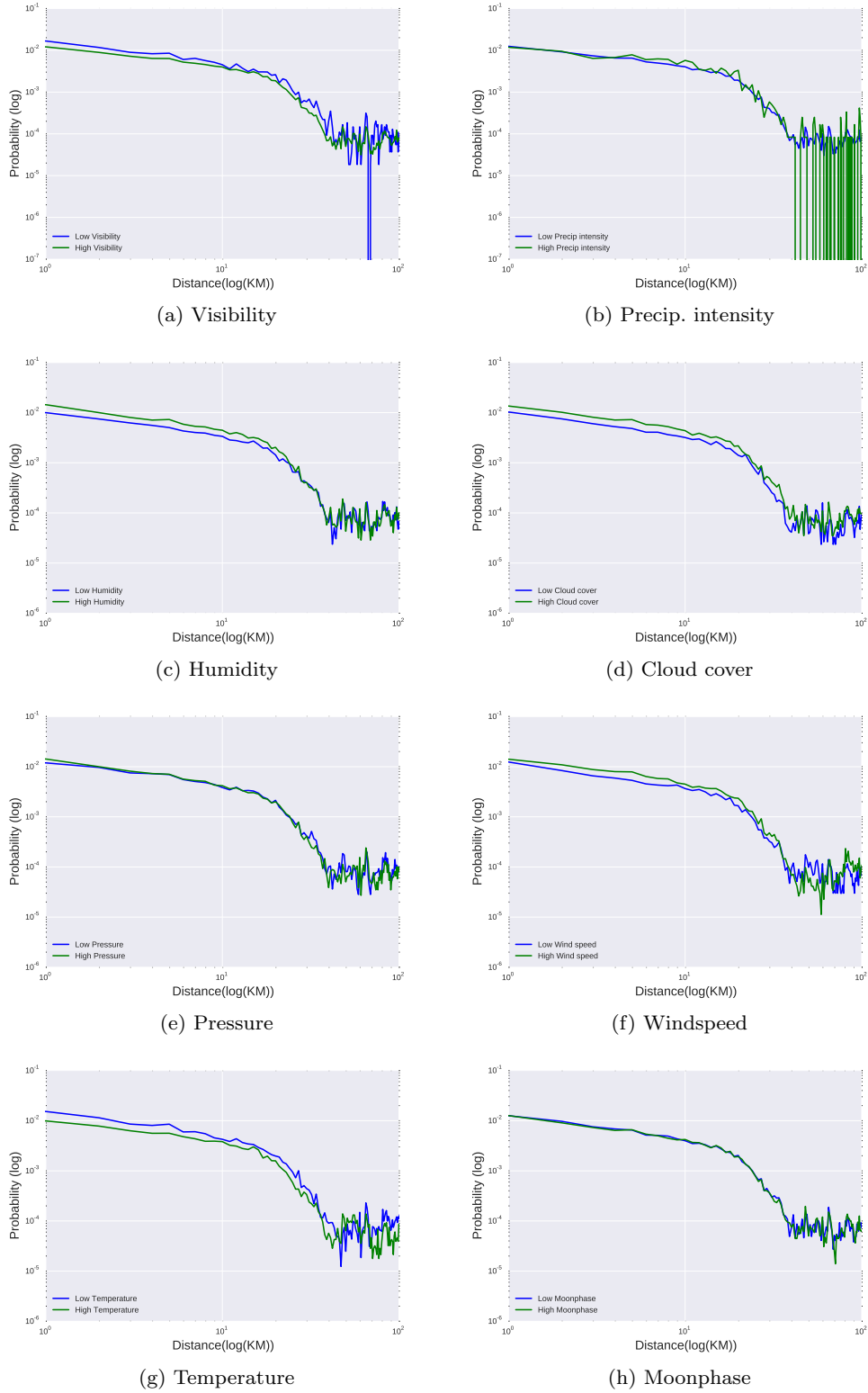


Fig. 4: Check-in probabilities in respect to travel distance under different weather conditions. While ‘pressure’ and ‘moonphase’ do not show a high impact on the travel distance at extreme conditions, the other features do. The insignificant influence of ‘precipitation intensity’ was unexpected. Nevertheless, a p -value (Kruskal-Wallis test) smaller than .001 for all eight weather features proves a statistically significant difference between the two distributions in the plots.

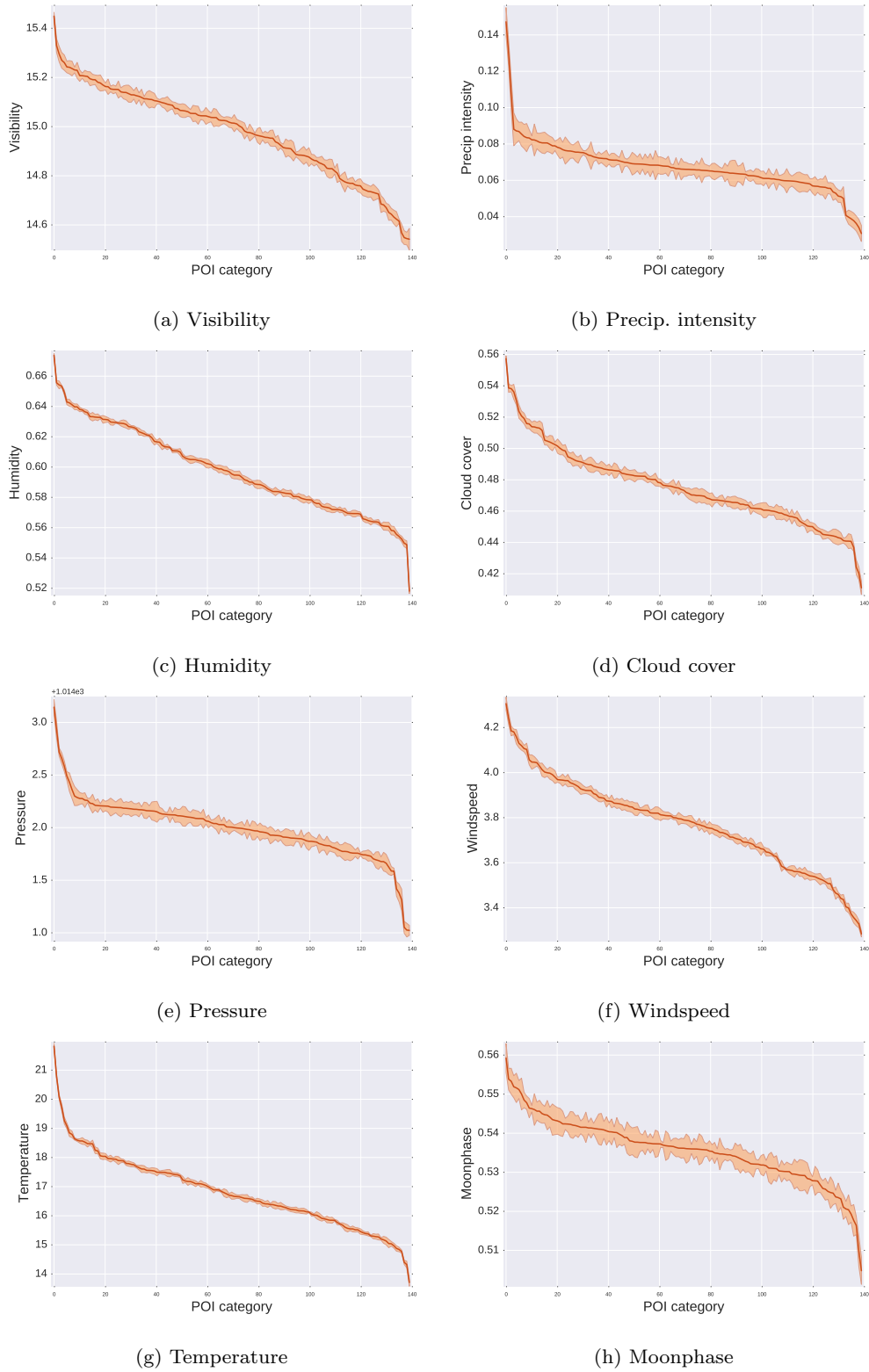


Fig. 5: Means for the eight weather features (sorted) in the context of the top-120 categories in the Foursquare dataset. As presented, there are observable differences between the categories, implying that certain categories are checked into under certain weather conditions. Furthermore, a Kruskal-Wallis test confirms that all eight distributions are not drawn from a uniform distribution ($p < .001$).

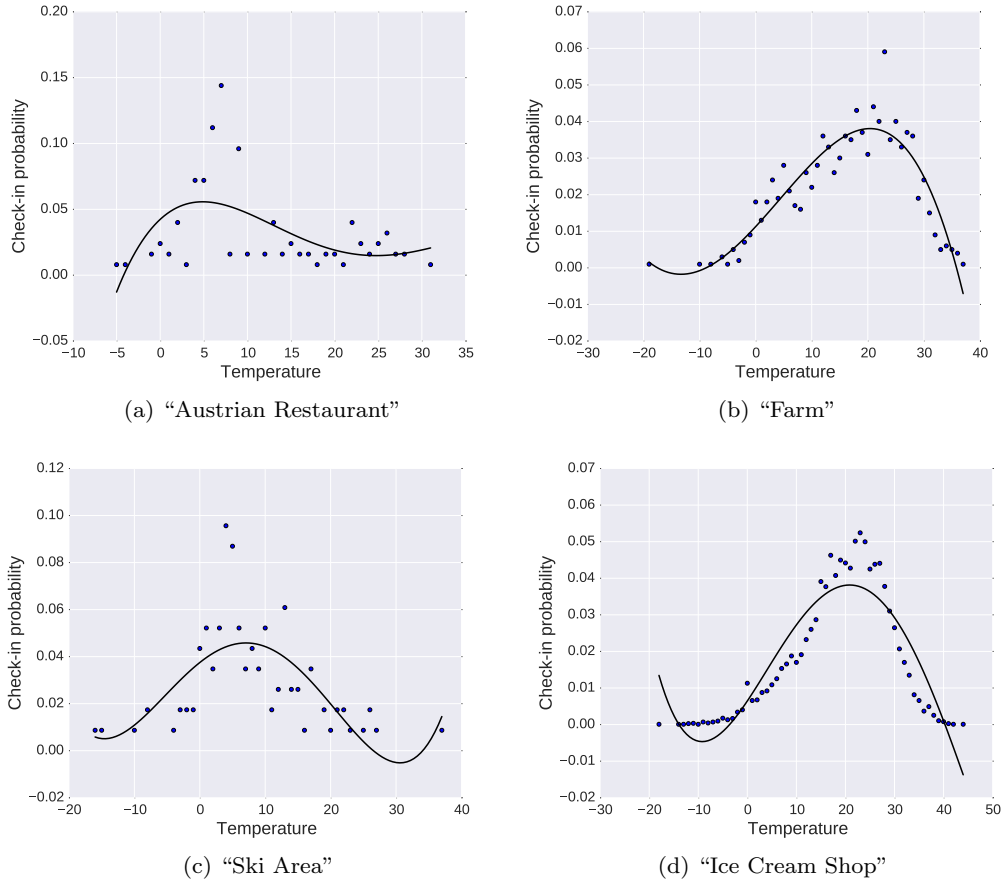


Fig. 6: Examples of check-in distributions over different categories in Foursquare. On the left hand side, places where people check-in at lower temperatures are shown vs on the right, where higher temperature places are featured.

tures may have a higher impact when considering travel distance and weather at the same time in a POI recommender scenario. Nevertheless, a Kolmogorow-Smirnow-Test with $p < .001$ shows that all eight distributions are significantly different when compared to one another.

POI Category Analysis. Yet another interesting sub-question regarding RQ1 is whether or not it is possible to find a link between the category of POIs visited by users and weather conditions. To investigate this in more detail, we had a closer look at the Foursquare category tree and how people check into these categories under different weather conditions. Figure 5 shows the results of this investigation by presenting the top-120 categories measured in terms of number of check-ins that have been checked into at least 5,000 times. Due to space limitations, the actual category names are not shown. As presented, there are observable differences between how people check into categories. The values shown are means over the eight weather features investigated in the context of the categories. The hulls in the plots further denote that the standard errors are rather

small. A Kruskal-Wallis test also confirms that all eight distributions are not drawn from a uniform distribution ($p < .001$). Hence, certain categories are checked into under certain weather conditions.

It is interesting to point out that even the ‘moon-phase’ feature shows some differences here. To have a better understanding of these trends, we also looked further into the sub-category level for the 60 cities in our dataset. Figure 6 shows an excerpt of the former investigation by presenting four plots of different POI types and certain trends that are observable, no matter in which city the POIs occur. A different observation can be made when looking at Figure 7, which presents an example of the POI category ‘Beach’ in four different places and their respective check-in temperature distributions.

In summary, we can confirm that many of these patterns are observable when inspecting a sub-set of the data and that relationships exist between the different types of POIs and certain weather conditions.

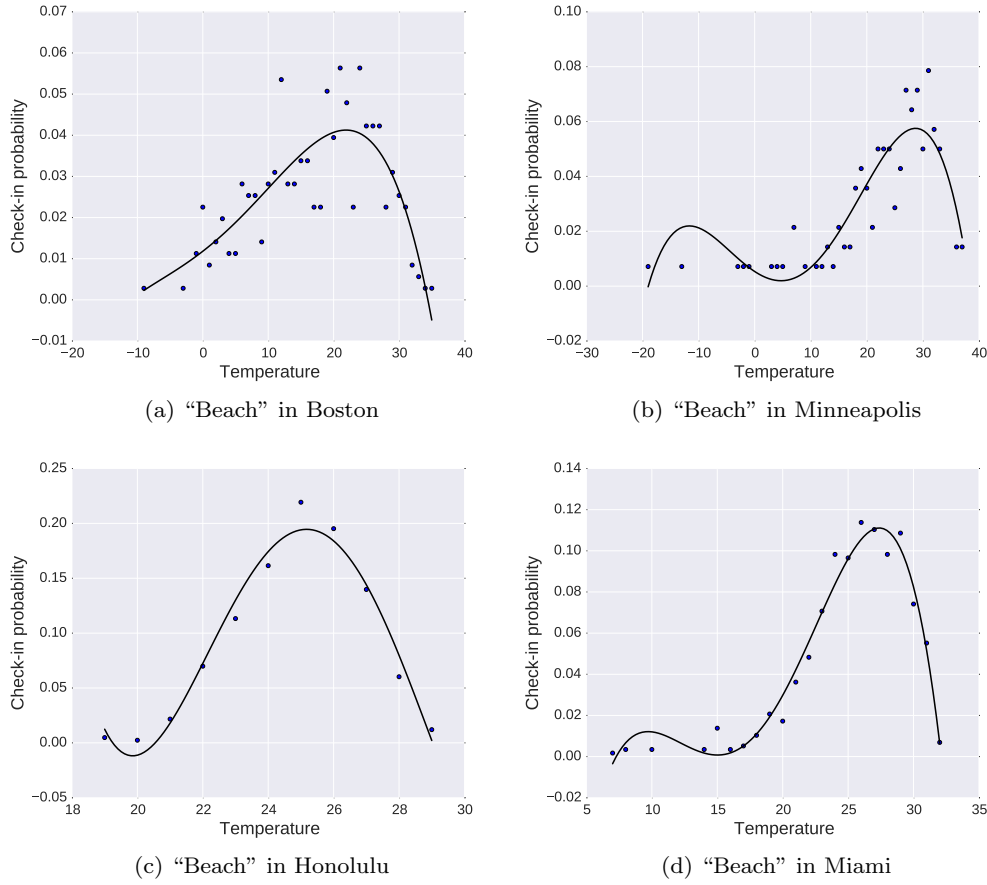


Fig. 7: The distributions of the category “Beach” shows the diverse behaviour of people in different cities. While people in Boston already go to the beach at lower temperatures as well as when it is hot, people in Minneapolis typically go to the beach at higher temperatures.

Feature Analysis at City Context. Figure 8 shows how the weather features vary in each city in the Foursquare dataset. Variability is expressed here through standard deviation. To obtain a variability score for each city, we obtained a mean weather feature score for each POI in the city (calculated over the user check-ins and corresponding weather conditions) and used this value to calculate the standard deviations. Northern and southern cities were classified manually by investigating their latitude positions using Google Maps and Google’s Geo location API.

It is interesting to note that with the exception of ‘moonphase’, all features present a dependency regarding the city where they are measured, indicating that a recommender model employing the same weather features may work differently in different regions. This is also confirmed by a Kruskal-Wallis test that shows significant differences ($p < .001$) when performed on each of the weather features except for the ‘moonphase’ feature. Moreover, our analysis shows that there is a

higher variability in the northern states of the US and a very low variability in the southern ones, with Honolulu showing the lowest variability in general (i.e., it is easy to find it to the left side of the x-axis in many plots, such as ‘temperature’ and ‘humidity’).

Summary. Following this analysis, we can confidently state that there is indeed a relationship between weather conditions and the check-in behaviour of Foursquare users. Our analysis clearly shows that weather features vary in their distribution and that there are several features that are highly correlated with each other. Furthermore, we find that travel distance is also affected by weather conditions and some POI categories are mostly visited under specific weather conditions. Finally, we find that the weather features investigated vary across cities in the US, hinting towards the need to employ different weather features for different cities.

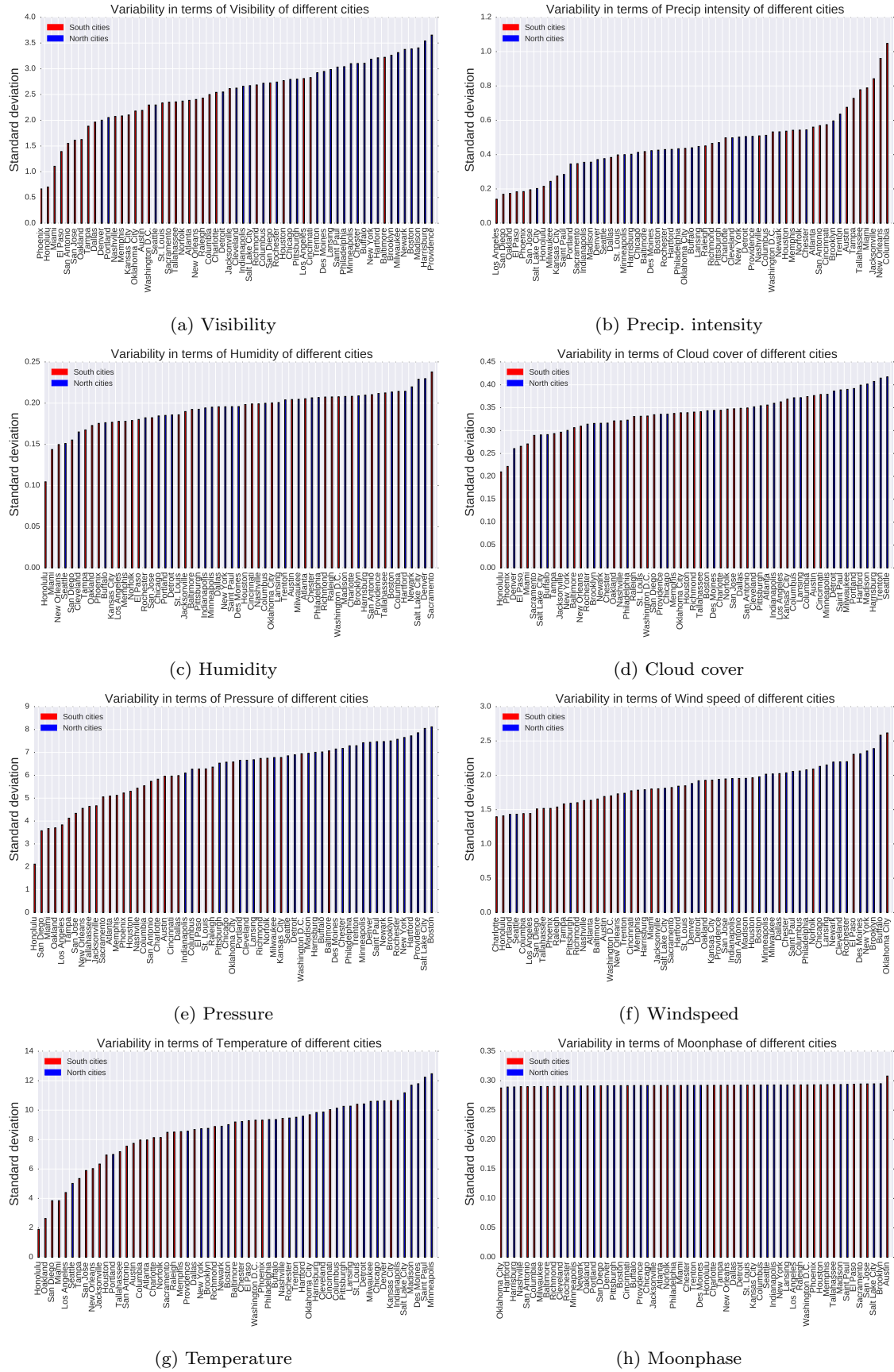


Fig. 8: Weather feature variability (sorted) measured via standard deviation over cities. Left: cities with the lowest variability. Right: cities with the highest variability.

5 RQ2: A Weather-Aware POI Recommendation Model

The previous section provided proof that weather information does indeed have a correlation with the check-in behaviour of Foursquare users and, hence, may yield a positive impact on POI recommendations. In this section, we describe how we exploit this information in order to improve the quality of POI recommendations. As mentioned in Section 2, several studies have appeared that propose different approaches to this problem. Among those, the Rank-GeoFM [18] model appears to be one of the most interesting, given that it has proven to provide more accurate recommendations than several other strong baselines. Moreover, it is flexible enough to allow an easy incorporation of additional contextual data. Therefore, we have decided to build our solution on top of Rank-GeoFM by outfitting it with weather features. Rank-GeoFM is based on matrix factorization with stochastic gradient descent (SGD) [17], an iterative approach used to learn the model parameters. The original Rank-GeoFM basically exploits three sources of input data: geographical distances between venues, user check-in history and time. We build upon Rank-GeoFM by incorporating weather data with the aim of modeling its influence on the preferences of users for venues. Table 4 presents the notation used in this section to describe our approach.

User-Preference-Score. Rank-GeoFM is based on a MF-based recommender for personalized ranking that models user preferences for POIs. To this end, a function was defined to measure the incompatibility between the predicted y_{ul} and true preference score x_{ul} represented by the frequency of visits of user $u \in U$ to location $l \in L$. The assumption is that POI $l \in L$ is preferred over $l' \in L$ by user $u \in U$ if the check-in frequency of l is higher than the one of l' (i.e. $x_{ul} > x_{ul'}$). Equation 1 measures the incompatibility between the true and predicted rankings of POIs by the factorization model where $I(\cdot)$ is the indicator function and ϵ is introduced to soften the ranking incompatibility.

$$Incomp(y_{ul}, \epsilon) = \sum_{l' \in L} I(x_{ul} > x_{ul'}) I(y_{ul} < y_{ul'} + \epsilon) \quad (1)$$

Notice that the incompatibility function counts the number of locations $l' \in L$ that should have been ranked lower than l , but are falsely ranked higher by the model and vice-versa. The overall goal is to learn the model parameters (more details in the *Learning Approach* description) that minimize the ranking incompatibility for all user-POI pairs (D_1) whose check-in frequency is greater than 0 (i.e. the set D_1 defined in Table 4).

Table 4: Notation used to describe our approach.

Sym.	Description
U	set of users $u_1, u_2, \dots, u_{ U }$.
L	set of POIs $l_1, l_2, \dots, l_{ L }$.
FC_f	set of classes for feature f .
Θ	latent model parameters containing the learned weights $\{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F^{(1)}\}$ for locations, users and weather features.
X_{ul}	matrix of dimensions $ U \times L $ containing the frequency of check-ins of users at POIs.
x_{ul}	frequency of check-ins of user u in POI l (a cell in matrix X_{ul} in row u and column l).
X_{ulc}	tensor of dimensions $ U \times L \times FC_f $ containing the check-ins of users at POIs at a specific feature class c .
x_{ulc}	frequency of check-ins of user u in POI l regarding c (a cell in X_{ulc}).
D_1	user-POI pairs: $(u, l) x_{ul} > 0$.
D_2	user-POI-feature class triples: $(u, l, c) x_{ulc} > 0$.
W	geographical probability matrix of size $ L \times L $ where $w_{ll'}$ contains the probability of l' being visited after l has been visited according to their geographical distance.
WI	matrix containing the probability that a weather feature class c is influenced by feature class c' . $wi_{cc'} = \cos_sim(c, c')$.
$N_k(l)$	set of k nearest neighbors of POI l . Distances are calculated based on the geographic distances of venues.
y_{ul}	the recommendation score of user u and POI l .
y_{ulc}	the recommendation score of user u , POI l and weather feature class c .
$I(\cdot)$	indicator function returning $I(a) = 1$ when a is true and 0 otherwise.
ϵ	parameter used to soften ranking incompatibility.
γ_w	learning rate for updates on weather latent parameters.
γ_g	learning rate for updates on latent parameters from base approach.
$E(\cdot)$	a function that turns the ranking incompatibility function into a loss of the form $E(r) = \sum_{i=1}^r \frac{1}{i}$.
$\delta_{ucll'}$	function to approximate the indicator function with a continuous sigmoid function $s(a) = \frac{1}{1 + \exp(-a)}$. $\delta_{ucll'} = s(y_{ul'c} + \epsilon - y_{ulc})(1 - s(y_{ul'c} + \epsilon - y_{ulc}))$.
$\lfloor \frac{ L }{n} \rfloor$	if the n^{th} location l' was ranked incorrect by the model the expectation is that overall $\lfloor \frac{ L }{n} \rfloor$ locations are ranked incorrect.
$I(\cdot)$	indicator function.
g, μ	auxiliary variable that save partial results of the calculation of the stochastic gradient.

Equation 2 below defines the objective function more formally, wherein y_{ul}^Θ denotes the preference score prediction parametrized by Θ .

$$\Theta = \arg \min_{\Theta} \sum_{(u,l) \in D_1} E(Incomp(y_{ul}^\Theta, \epsilon)) \quad (2)$$

$E(\cdot)$ defined in Equation 3 below is a function that converts the ranking incompatibility into a weight decay-based loss function, i.e., $E(\cdot)$ represents a smooth function over the absolute number of incompatibilities.

$$E(r) = \begin{cases} \sum_{i=1}^r \frac{1}{i} & \text{if } r > 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

So when iterating over the set D_1 (or D_2) and having already n incorrectly ranked items, the $n+1$ th one just contributes $\frac{1}{n+1}$ to the overall loss.

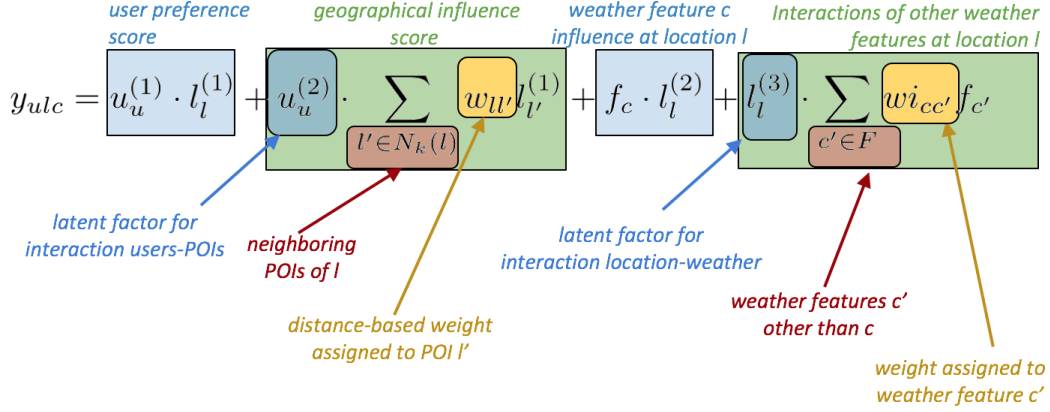


Fig. 9: Graphical explanation of the recommendation score computation.

Another contribution of Rank-GeoFM is the ability to mitigate data sparsity. The idea is to exploit the unvisited POIs of user u since $I(x_{ul} > x_{ul'})$ always amounts to true on not visited POIs l' (whereby it always amounts to false for $(u, l) | x_{ul} = 0$). Therefore, while often ignored in conventional MF algorithms, the unvisited POIs are used as additional training data, thus, contributing to the overall loss.

Geographical Influence Score. The next step of Rank-GeoFM is to use the geographical distance between pairs of locations as additional information for modeling user check-in behaviour. Figure 4 shows that the check-in distribution over distances resembles a power law distribution and, therefore, one can assume that the probability of visiting POI l' decreases the farther away l' is from the previously visited POI l . Similar studies from, e.g., [19], reveal the same behaviour and, hence, the assumption of Rank-GeoFM was that nearby POIs tend to have a higher probability of being visited. Based on these observations, Equation 4 shows the calculated probability of l' being visited after l , where $d(l, l')$ is the geographical distance based on the latitude and longitude of l and l' respectively. Notice that if l is not among the k -nearest neighbors of l' (i.e. $l' \notin N_k(l)$), the probability $w_{ll'}$ is set to 0.

$$w_{ll'} = \begin{cases} (0.5 + d(l, l'))^{-1} & \text{if } l' \in N_k(l) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Weather Influence Score. Our model (WPOI) is summarized in Figure 9. It shows that in order to recommend POIs for a user, we calculate a score y_{ulc} considering user u , location l and weather context c data. In the equation, the initial two terms are adopted from Rank-GeoFM, which indicate user preference score and geographical influence score. Our model WPOI adjusts

Rank-GeoFM to include two terms of weather-related information.

In order to incorporate weather data into Rank-GeoFM, the weather feature values needed to be discretized. This was done to reduce data sparsity. For example, in many cases it would be hard to find check-ins associated with very specific temperature values since temperature is a real number. Thus, transforming continuous values of weather features (e.g., temperature) into intervals might alleviate this problem. A mapping function is introduced (see Equation 5) that converts the weather features into interval bins (in this paper also called feature classes), where $|FC_f|$ defines the size of the bin for the current weather feature.

$$c_f(\text{value}) = \left\lfloor \frac{(\text{value} - \min(f))(|FC_f| - 1)}{\max(f) - \min(f)} \right\rfloor \quad (5)$$

The best results were obtained for $|FC_f| = 20$ (found on a validation set as described in Section 6). For example, consider that we have temperatures ranging from -10°C to 10°C . The set of classes for this feature (i.e., temperature) could be as follows:

- $FC_1 = [-10^\circ\text{C}, -9^\circ\text{C})$
- $FC_2 = [-9^\circ\text{C}, -8^\circ\text{C})$
- ...
- $FC_{20} = [9^\circ\text{C}, 10^\circ\text{C}]$

Recommendation Score Computation. In order to compute recommendations, a function is introduced that calculates the recommendation score. As in conventional MF approaches, latent model parameters are needed. These parameters are proposed as matrices in a K dimensional space. For incorporating user-preference-score into the recommendation score, the matrices $U^{(1)} \in \mathbb{R}^{|U| \times K}$ and $L^{(1)} \in \mathbb{R}^{|L| \times K}$ are used to model the preferences of users to specific POIs. Including the geographical-influence-score in the model leads to the creation of the

matrix $U^{(2)} \in \mathbb{R}^{|U| \times K}$ that constitutes the geographical influence in the interaction between users and locations. As a last step, the extension of the algorithm with weather context requires three more latent parameters. The first one is for incorporating the weather-popularity-score, which models whether a location is popular within a specific feature class or not and is named $L^{(2)} \in \mathbb{R}^{|L| \times K}$. Furthermore, a matrix $L^{(3)} \in \mathbb{R}^{|L| \times K}$ is introduced to model the interactions between locations and feature classes. In other words, $L^{(3)}$ models the likelihood that a location will be visited given certain weather conditions. The third latent parameter $F \in \mathbb{R}^{|FC_f| \times K}$ is used to parametrize the feature classes of a specific weather feature. Additionally, a matrix $WI \in \mathbb{R}^{|FC_f| \times |FC_f|}$ is used to store the probability that a weather feature class c will be influenced by feature class c' , which is computed by Equation 6. The intuition is borrowed from the geographic influence modeling of Rank-GeoFM, i.e., if a location is strongly related to a feature class, e.g., $[20^\circ C, 22^\circ C]$, it will also very likely be related to its neighbor feature class, e.g., $[22^\circ C, 24^\circ C]$.

$$w_{cc'} = \frac{\sum_{u \in U} \sum_{l \in L} x_{ulc} x_{ulc'}}{\sqrt{\sum_{u \in U} \sum_{l \in L} x_{ulc}^2} \sqrt{\sum_{u \in U} \sum_{l \in L} x_{ulc'}^2}} \quad (6)$$

Having these latent parameters defined, the recommendation score for a user $u \in U$ POI $l \in L$ and feature class $c \in C$ is then computed as shown in Equation 7 below

$$\begin{aligned} y_{ul} &= u_u^{(1)} \cdot l_l^{(1)} + u_u^{(2)} \cdot \sum_{l' \in N_k(l)} w_{ll'} l_{l'}^{(1)} \\ y_{ulc} &= y_{ul} + f_c \cdot l_l^{(2)} + l_l^{(3)} \cdot \sum_{c' \in F} w_{cc'} f_{c'} \end{aligned} \quad (7)$$

, wherein each multiplication is a dot product between latent vectors of the entities that moderate the recommendation score. For example, $u_u^{(1)} \cdot l_l^{(1)}$ computes the preference of user u for location l , while $f_c \cdot l_l^{(2)}$ represents the degree of influence between weather feature class c on location l .

Learning Approach. The standard gradient descent algorithm (GD) is an iterative algorithm that updates the latent parameters in each iteration with the objective of minimizing the error function of the training dataset [17]. In other words, the algorithm minimizes an objective function J with parameters Θ by updating them in the negative direction of the gradient of J . Equation 8 shows the update of the parameters with the learning rate α for each training example x, y .

$$\Theta = \Theta - \alpha \frac{1}{n} \sum_{i=1}^n \nabla_{\Theta} J(x_i, y_i, \Theta) \quad (8)$$

This approach is computationally intensive for large samples since the gradient has to be computed over the whole dataset for every update. For this reason SGD approximates the true gradient with the gradient of each training example and performs an update on each training example. Equation 9 demonstrates the update of the parameters with the simplified gradient of one example.

$$\Theta = \Theta - \alpha \nabla_{\Theta} J(x_i, y_i, \Theta) \quad (9)$$

Although this simplification leads to a noisy representation of the gradient due to the stochastic process of randomly chosen examples at each iteration, SGD should behave like GD while having performance advantages on large scale datasets. Asymptotic analysis in [4] has shown that the “time to accuracy” ρ decreases from $n \log \frac{1}{p}$ to $\frac{1}{p}$ and is, therefore, independent of sample size n . As a result of the high update frequency, the learning rate usually has to be a small value.

To learn the latent model parameters defined as $\Theta = \{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F\}$, a learning algorithm has to be proposed in order to find the latent parameters that minimize a version of the objective function defined in Equation 2, wherein the loss is computed for each training triple $(u, l, c) \in D_2$. The authors of [18] state two issues for optimizing Equation 2:

- $E(Incomp(y_{ul}, \epsilon))$ is not differentiable; and
- Calculating $Incomp(y_{ul}, \epsilon)$ is computationally intensive.

Therefore, they first introduce a continuous approximation of $E(Incomp(y_{ul}, \epsilon))$ by approximating the indicator function $I(a)$ with the sigmoid function $s(a) = \frac{1}{1 + \exp(-a)}$, which leads to the computation of the stochastic gradient of E w.r.t Θ in Equation 11 with $\delta_{ull'}$ defined in Equation 10.

$$\delta_{ull'} = s(y_{ul'} + \epsilon - y_{ul})(1 - s(y_{ul'} + \epsilon - y_{ul})) \quad (10)$$

$$\begin{aligned} & \frac{\partial E(Incomp(y_{ul}, \epsilon))}{\partial \Theta} \\ & \approx E(Incomp(y_{ul}, \epsilon)) \frac{\sum_{l' \in L} I(x_{ul} > x_{ul'}) \frac{\partial s(y_{ul'} + \epsilon - y_{ul})}{\partial \Theta}}{Incomp(y_{ul}, \epsilon)} \\ & = E(Incomp(y_{ul}, \epsilon)) \frac{\sum_{l' \in L} I(x_{ul} > x_{ul'}) \delta_{ull'}}{Incomp(y_{ul}, \epsilon)} \end{aligned} \quad (11)$$

The computation of $Incomp(y_{ul}, \epsilon)$ is still very intense. For this reason, a fast learning scheme is employed that estimates $Incomp(y_{ulc}, \epsilon)$ by means of sampling. Since just incorrectly-ranked POIs contribute to the loss, the idea is to sample POIs and calculate incompatibility

until the first POI is incorrectly ranked, with n being the number of sampled POIs at this point. The gradient for one incorrectly ranked POI is then denoted in Equation 12, which approximates Equation 11 and disperses the summation.

$$\frac{\partial \bar{E}}{\partial \Theta} = E(Incomp(y_{ul}, \epsilon)) \delta_{ul'} \frac{\partial(y_{ul'} + \epsilon - y_{ul})}{\partial \Theta} \quad (12)$$

Then n is the number of sampled POIs before the incorrect one and $\frac{1}{Incomp(y_{ul}, \epsilon)}$ the chance of each POI to be the incorrect one. The more incorrect POIs exist, the higher n will be on average. Therefore, it can be assumed that n follows a geometric distribution dependent on parameter $p = \frac{Incomp(y_{ul}, \epsilon)}{|L|}$. Using the fact that the expectation of a geometrical distribution with parameter $p = \frac{1}{p}$ and $n \approx \left\lceil \frac{1}{p} \right\rceil = \left\lceil \frac{|L|}{Incomp(y_{ul}, \epsilon)} \right\rceil$, the incomp. function can be estimated with $Incomp(y_{ul}, \epsilon) \approx \left\lceil \frac{|L|}{n} \right\rceil$, leading to the computation of the gradient of one incorrectly-ranked POI as shown in Equation 13.

$$\frac{\partial \bar{E}}{\partial \Theta} \approx E\left(\left\lceil \frac{|L|}{n} \right\rceil\right) \delta_{ul'} \frac{\partial(y_{ul'} + \epsilon - y_{ul})}{\partial \Theta} \quad (13)$$

This estimation follows the intuition that if, for example, the third POI was ranked incorrectly, it is expected that $Incomp(y_{ul}, \epsilon) = \frac{|L|}{3}$. Using γ as the learning rate, the update of the parameters Θ is then calculated as follows:

$$\Theta \leftarrow \Theta - \gamma \frac{\partial \bar{E}}{\partial \Theta} \quad (14)$$

Using the estimation of the incompatibility function, it turns out that the complexity reduces from $O(K|L|k)$ to $O(Knk)$, whereby n will be very small in the beginning, because the parameters are not well fitted to the training data and an incorrect POI will be obtained very quickly. It will then grow a bit during the training process. Nevertheless, it is not expected that every item will be ranked correctly, so n will be still smaller than $|L|$ when the algorithm converges.

To avoid overfitting, the authors added a constraint to the adjustment of the latent factors at each step of the learning process. They introduced a hyperparameter C that regularizes the magnitude of the latent factors as shown in equations 15-20. Additionally, the regularization terms for geographical influence and weather context influence are kept in a smaller value αC and βC in order to balance the contributions of these two factors.

$$\|u_u^{(1)}\|_2 \leq C \xrightarrow{reg.} u_u^{(1)} \leftarrow C \frac{u_u^{(1)}}{\|u_u^{(1)}\|_2}, \quad u = 1, 2, \dots, |U| \quad (15)$$

$$\|u_u^{(2)}\|_2 \leq \alpha C \xrightarrow{reg.} u_u^{(2)} \leftarrow \alpha C \frac{u_u^{(2)}}{\|u_u^{(2)}\|_2}, \quad u = 1, 2, \dots, |U| \quad (16)$$

Algorithm 1: WPOI

Input: check-in data D_1 and D_2 , geographical influence matrix W , weather influence matrix WI , hyperparameters $\epsilon, C, \alpha, \beta$ and learning rate γ_g and γ_w

Output: parameters of the model

$$\Theta = \{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F\}$$

```

1 init: Initialize  $\Theta$  with  $\mathcal{N}(0, 0.01)$ ; Shuffle  $D_1, D_2$  randomly
2 repeat
3   for  $(u, l) \in D_1$  do
4     Compute  $y_{ul}$  as Equation 7 and set  $n = 0$ 
5     repeat
6       Sample a POI  $l'$ , Compute  $y_{ul'}$  as
7       in Equation 7 and set  $n++$ 
8     until  $I(x_{ul} > x_{ul'})I(y_{ul} < y_{ul'} + \epsilon) = 1$  or
9      $n > |L|$ 
10    if  $I(x_{ul} > x_{ul'})I(y_{ul} < y_{ul'} + \epsilon) = 1$  then
11       $\eta = E\left(\left\lceil \frac{|L|}{n} \right\rceil\right) \delta_{ul'}$ 
12       $g =$ 
13       $\left(\sum_{l^* \in N_k(l')} w_{ll^*} l_{l^*}^{(1)} - \sum_{l^+ \in N_k(l)} w_{ll^+} l_{l^+}^{(1)}\right)$ 
14       $u_u^{(1)} \leftarrow u_u^{(1)} - \gamma_g \eta (l_{l'}^{(1)} - l_l^{(1)})$ 
15       $u_u^{(2)} \leftarrow u_u^{(2)} - \gamma_g \eta g$ 
16       $l_{l'}^{(1)} \leftarrow l_{l'}^{(1)} - \gamma_g \eta u_u^{(1)}$ 
17       $l_l^{(1)} \leftarrow l_l^{(1)} + \gamma_g \eta u_u^{(1)}$ 
18    end
19  end
20  for  $(u, l, c) \in D_2$  do
21    Compute  $y_{ulc}$  as Equation 7 and set  $n = 0$ 
22    repeat
23      Sample a POI  $l'$  and feature class
24       $c'$ , Compute  $y_{ul'c'}$  as in
25      Equation 7 and set  $n++$ 
26    until
27     $I(x_{ulc} > x_{ul'c'})I(y_{ulc} < y_{ul'c'} + \epsilon) = 1$  or
28     $n > |L|$ 
29    if  $I(x_{ulc} > x_{ul'c'})I(y_{ulc} < y_{ul'c'} + \epsilon) = 1$ 
30    then
31       $\eta = E\left(\left\lceil \frac{|L|}{n} \right\rceil\right) \delta_{ul'}$ 
32       $g =$ 
33       $\left(\sum_{c^* \in FC_f} w_{cc^*} f_{c^*} - \sum_{c^+ \in FC_f} w_{cc^+} f_{c^+}\right)$ 
34       $f_c \leftarrow f_c - \gamma_w \eta (l_{l'}^{(2)} - l_l^{(2)})$ 
35       $l_{l'}^{(3)} \leftarrow l_{l'}^{(3)} - \gamma_w \eta g$ 
36       $l_{l'}^{(2)} \leftarrow l_{l'}^{(2)} - \gamma_w \eta f_c$ 
37       $l_l^{(2)} \leftarrow l_l^{(2)} + \gamma_w \eta f_c$ 
38    end
39  end
40  end
41  Project updated factors to accomplish constraints
42 until convergence
43 return  $\Theta = \{L^{(1)}, L^{(2)}, L^{(3)}, U^{(1)}, U^{(2)}, F\}$ 

```

$$\|l_l^{(1)}\|_2 \leq C \xrightarrow{reg.} l_l^{(1)} \leftarrow C \frac{l_l^{(1)}}{\|l_l^{(1)}\|_2}, \quad l = 1, 2, \dots, |L| \quad (17)$$

$$\|l_l^{(2)}\|_2 \leq \beta C \xrightarrow{reg.} l_l^{(2)} \leftarrow \beta C \frac{l_l^{(2)}}{\|l_l^{(2)}\|_2}, \quad l = 1, 2, \dots, |L| \quad (18)$$

$$\|l_l^{(3)}\|_2 \leq \beta C \xrightarrow{reg.} l_l^{(3)} \leftarrow \beta C \frac{l_l^{(3)}}{\|l_l^{(3)}\|_2}, \quad l = 1, 2, \dots, |L| \quad (19)$$

$$\|f_c\|_2 \leq \beta C \xrightarrow{reg.} f_c \leftarrow \beta C \frac{f_c}{\|f_c\|_2}, \quad c = 1, 2, \dots, |FC_f| \quad (20)$$

Weather-Aware POI Recommender System (WPOI).

This paragraph describes in detail how we incorporated the weather context into the Rank Geo-FM algorithm. First of all, the hyperparameters have to be initialized. As in [18], a parameter tuning for the hyperparameters α , k , β and K was performed. As observed in [18], it turns out that changing K does not significantly affect the performance of the algorithm. In our experiments, this parameter was set to 100. The size of the neighborhood k showed the best results at 300.

Furthermore, tuning the regularization terms α and β for geographical and time influence had the best accuracy at .2. As shown in Algorithm 1, a loop over the set D_1 of check-ins containing each (u, l) tuple is made. Lines 3-16 describe the original Rank Geo-FM algorithm as proposed in [18], iterating over all pairs (u, l) and adjusting the latent parameters accordingly. After that, Lines 17-30 show the incorporation of the weather context into the original Rank-GeoFM approach. In order to adjust the latent parameters to the respective weather context, an iteration over all $(user, venue, feature-class)$ triples $(u, l, c) \in D_2$ was introduced to adjust the latent parameters on the incorrectly ranked venues according to the specific weather context. That has to be done because the algorithm might rank a triple (u, l, c) correctly with no need for adjustments, while (u, l, c') might be ranked incorrectly, which would involve a parameter adjustment. The modifications of the latent parameters at each learning step are then done according to the base algorithm. The latent parameters having the best performance on the validation-set are then returned to be validated on the test set. During our studies, it turned out that adjusting the latent parameters of the weather context with a learning rate $\gamma = .0001$ as used in [18] was too high and the algorithm did not find the minimum. As such, we decided to use a learning rate of $\gamma_w = .00001$ for the weather context parameters, wherein a stable learning rate was achieved as one can see in Figure 10.

6 RQ3: Investigating the Model Performance

In the previous section, we have described how we extended Rank-GeoFM to incorporate weather data. This yielded a new model which we call WPOI. In this section, we describe in detail how the model was evaluated and how it performs in comparison to the original model. Furthermore, we evaluate to what extent each of the eight weather features performs and how this compares to standard recommender algorithms.

Dataset pre-processing. In order to answer RQ1, we have used all the collected check-in data performed in

the US. In this section, we focus our investigations on only four cities in the US, namely, Minneapolis, Boston, Miami and Honolulu. The reason for choosing these four cities is the result of: (a) the observation that they had the highest and lowest weather variability as shown in Figure 8; (b) a decent amount of users available that had checked into different places at least 20 times; and (c) venues that had been checked into at least two times. This pruning ensures that noisy POIs and users are removed and data sparsity issues are reduced. The basic statistics of this dataset are shown in Table 5.

Evaluation protocol. As evaluation protocol, we chose the same protocol as proposed in Li et al.'s [18] work on the original Rank-GeoFM model. Hence, we split the dataset (according to the time line) into training, validation and test sets for each city by adding the first 70% of the check-ins of each user to the training set, the following 20% to the test set and the rest to the validation set (=10%). The training set was then used to learn the latent model parameters. During the training phase of the algorithm, the validation set was used to tune the algorithm convergence. When convergence was observed (typically around 3,000 – 5,000 iterations with the fast learning scheme enabled), the training was stopped and the learned parameters were used to evaluate the model on the test set.

Baseline methods. We used the original Rank-GeoFM approach that takes into account both the check-in history of users and geographical influence as baseline. We also compare to the time-based method of Rank-GeoFM that was also introduced in Li et al. [18]. Finally, we also make use of classic recommender systems models such as the Most Popular Items approach (MP), user and item-based collaborative filtering (denoted as UserKNN and ItemKNN) [25] and Weighted matrix factorization (WRMF) [15].

Evaluation metric. At a given time stamp t and weather condition c , a user u is just able to check into a single POI l . Thus, we assume the more rigid scenario in which there can only be one correct recommendation at each recommendation request. Due to this fact, evaluation over the test dataset is done for each (u, l, c) triple separately, wherein u indicates *user*, l indicates *location* and c indicates *weather feature class*. Therefore, we chose NDCG@k (Normalized Discounted Cumulative Gain) at $k = 20$ to measure the performance of the recommender. NDCG is a ranking-dependent metric that not only measures how many POIs can be correctly predicted, but also takes the position of the POI in the recommended list with length k into account. The

Table 5: Basic statistics of the pruned dataset used in the recommender evaluation experiments.

City	Num. Check-Ins	Num. POIs	Num. Users	Sparsity
Minneapolis	37,737	797	436	89.1%
Boston	42,956	1141	637	94.3%
Miami	29,222	796	410	91.0%
Honolulu	16,042	410	173	77.4%

NDCG metric is based on the *Discounted Cumulative Gain* ($DCG@k$), which is given by [24]:

$$DCG@k = \sum_{k=1}^{|r_u^k|} \left(\frac{2^{B(k)} - 1}{\log_2(1 + k)} \right) \quad (21)$$

, where $B(k)$ is a function that returns 1 if the recommended product at position k in the recommended list is relevant. $NDCG@k$ is calculated as $DCG@k$ divided by the ideal DCG value $iDCG@k$, which is the highest possible DCG value that can be achieved, if all of the relevant POIs would be recommended in the correct order. Taken together, it is given by the following Equation [24]:

$$NDCG@k = \frac{1}{|U|} \sum_{u \in U} \left(\frac{DCG@k}{iDCG@k} \right) \quad (22)$$

6.1 Results

Parameter Learning. Figure 10 shows the iterative learning approach of the recommender algorithm for the city of Boston as an example. The learning for all experiments was set to $\gamma_w = .00001$. Depending on the size of the city, convergence of the algorithm was accomplished at $\approx 3,000 - 5,000$ iterations. Learning the parameters for the model in Boston took around 22 hours and 5,000 iterations, whereby convergence for the other cities was often already achieved at $\approx 3,000$ iterations with only very little performance increase thereafter (differences in respect to $NDCG@20$ were .001 or less). In real word applications, one will have to weigh between performance increase and time needed to compute the remaining iterations in order to possibly obtain an even more accurate model. It also must be mentioned that the different weather features show different learning behaviours. On the one hand, ‘precipitation intensity’ in Figure 10b gains its major performance increase in the first 1,000 iterations, while ‘moonphase’ in Figure 10h needs $\approx 2,000$ to reach the same level of performance.

WPOI vs Rank-GeoFM. Figure 11 shows the results of our offline experiment. As presented, in all cases, Rank-GeoFM, enriched by our proposed weather features enhancement, significantly outperforms the original Rank-GeoFM algorithm (denoted with a dashed red line in the plots), which answers RQ2. In detail, we could increase the performance of the original Rank-GeoFM method by 18-36% for Minneapolis, 40-76% for Boston, 59-81% for Miami and 52-102% for Honolulu. For all pairwise-comparisons (recommenders with weather context vs without), a standard t-test further confirmed that these increases in performance are also statistically significant, $p < .001$.

What is even more interesting to note here is the performance of Rank-GeoFM, which utilizes the ‘time’ feature as a contextual factor. As highlighted, in all cases, Rank-GeoFM with weather features such as ‘visibility’ and ‘precipitation intensity’ outperforms the time-based variant, showing that indeed weather conditions help to improve the recommendation quality.

We also highlight the fact that certain weather features perform better than others and that this ranking seems to be city dependent. This can be clearly observed in Figure 11, which shows the results of Rank-GeoFM for each weather feature. This answers RQ3, showing which features provide the highest gain in recommendation quality. For example, in Honolulu, the best performing feature is ‘precipitation intensity’, while in Minneapolis, ‘visibility’ seems to work best among all investigated weather features. Similar patterns can be observed for other features such as ‘temperature’ or ‘cloud cover’, changing their relative importance across the four cities. These observations are in line with the results in Figure 1, showing a strong tendency of check-ins into POIs under certain weather conditions. However, what is also interesting to note is the good performance of the ‘moonphase’ feature, which appeared to be uniformly distributed in general (see Figure 1). Hence, it appears that at the level of locations there is indeed a strong preference for check-ins in different phases of the moon. Finally, the relative performance improvement over the original Rank-GeoFM also seems to be dependent on location. Hence, while our approach works to a great extent better compared to the base-

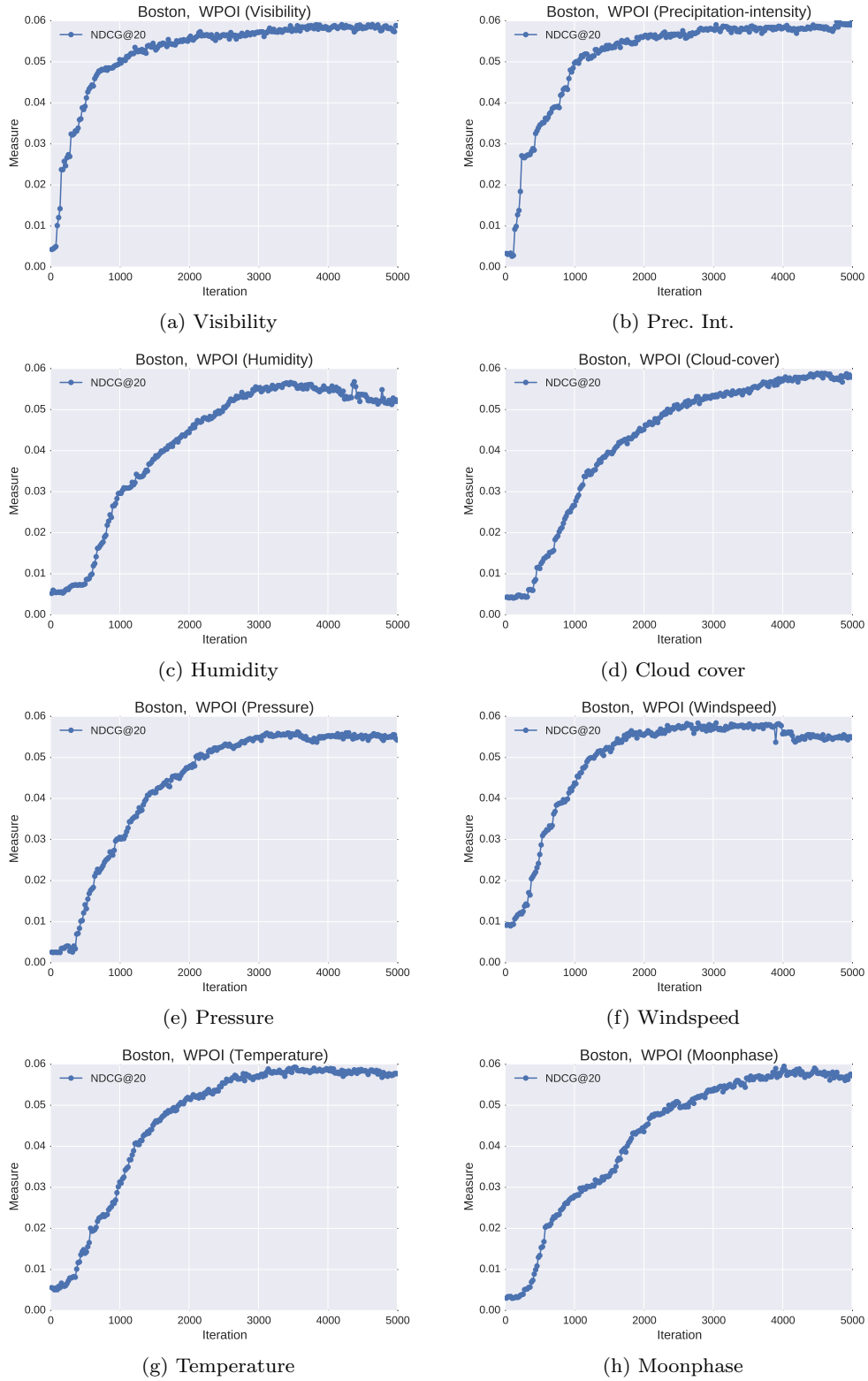


Fig. 10: Learning of the latent parameters of the WPOI model with the eight different weather features in Boston. As presented, stable learning rates and a convergence are archived between 3,000 and 5,000 iterations.

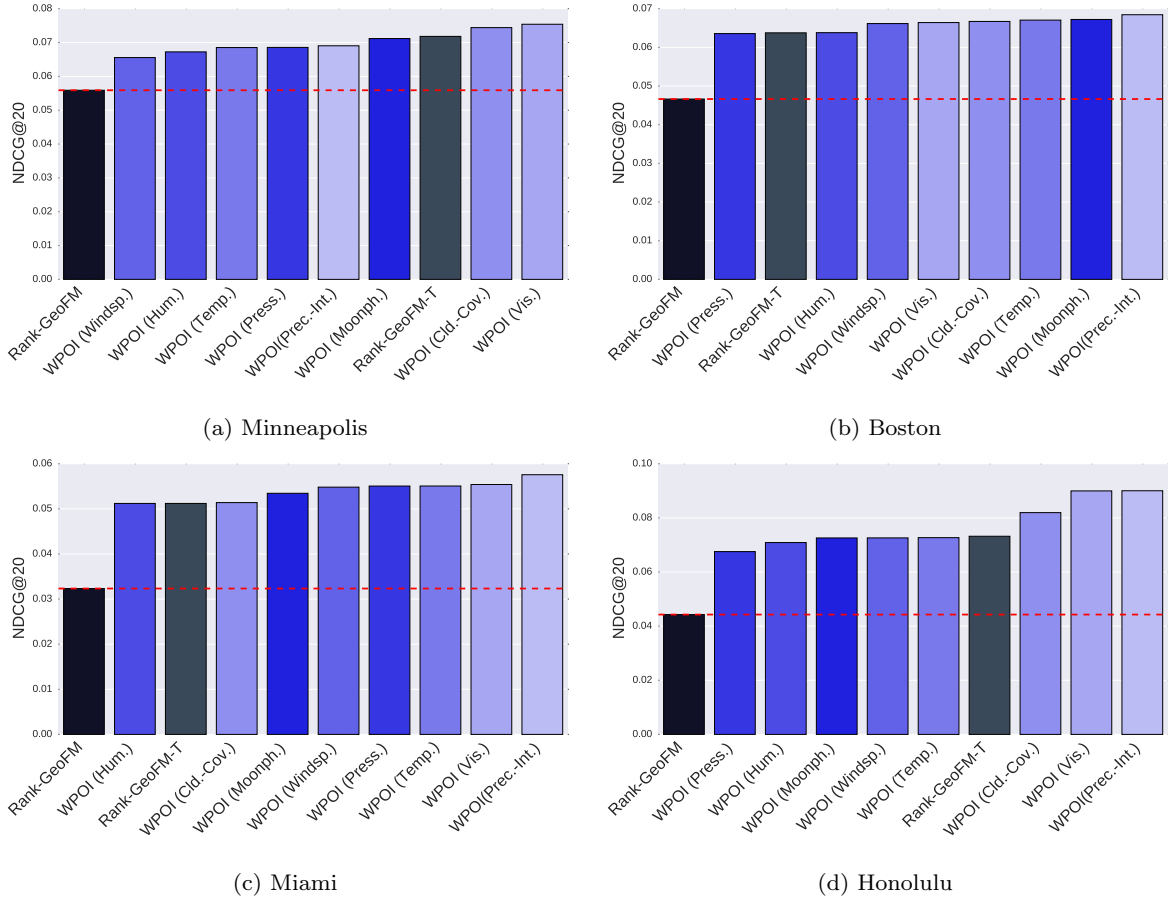


Fig. 11: Comparison of the eight weather features with Rank-GeoFM and Rank-GeoFM-T in the four cities. Rank-GeoFM is outperformed by WPOI significantly. Furthermore, this analysis leads to the assumption that weather has more impact on people living closer to the tropical zone (Miami, Honolulu) than on people living closer to the boreal zone (Minneapolis, Boston).

line for Miami and Honolulu, the differences are less pronounced for Minneapolis. One reason for this observation could be that there are more POIs available that demonstrate similar weather profiles. However, to further confirm these hypotheses, additional analyses are needed.

WPOI vs Baselines. In addition to the previous experiments, we also conducted a performance evaluation against some well-known and classic recommender algorithms. Further comparisons with other baselines can also be found in the original Rank-GeoFM paper. The results of this experiment are shown in Figure 12. As highlighted in all cases, WPOI outperforms the other methods significantly (pairwise comparison with a standard t-test and Bonferroni correction shows that this is statistically significant at $p < .001$). In numbers, that means an improvement of 1400% over ItemKNN and 28% over UserKNN for Minneapolis, an increase of 1280% over ItemKNN and 19% over MP for Boston, an

improvement of 1000% over ItemKNN and 18% over UserKNN for Miami and an increase of 1700% over ItemKNN and 31% over UserKNN for Honolulu.

What is also presented in the plots is the fact that the simple most popular algorithm (MP) proves to be a very strong baseline outperforming, e.g., in the city of Boston, the more sophisticated WRMF and CF methods. Interesting to note here is also the strong performance of the UserKNN method that appears to be the second strongest method in that experiment when averaged over all cities. As already shown in Noulas et al. [22], user-based collaborative filtering builds a very strong baseline in the field of POI recommender systems. Nevertheless, the authors also mention the high potential of matrix factorization methods that include contextual information in the model in their work. This potential is now revealed in the results of the WPOI recommender whose results apparently outperform con-

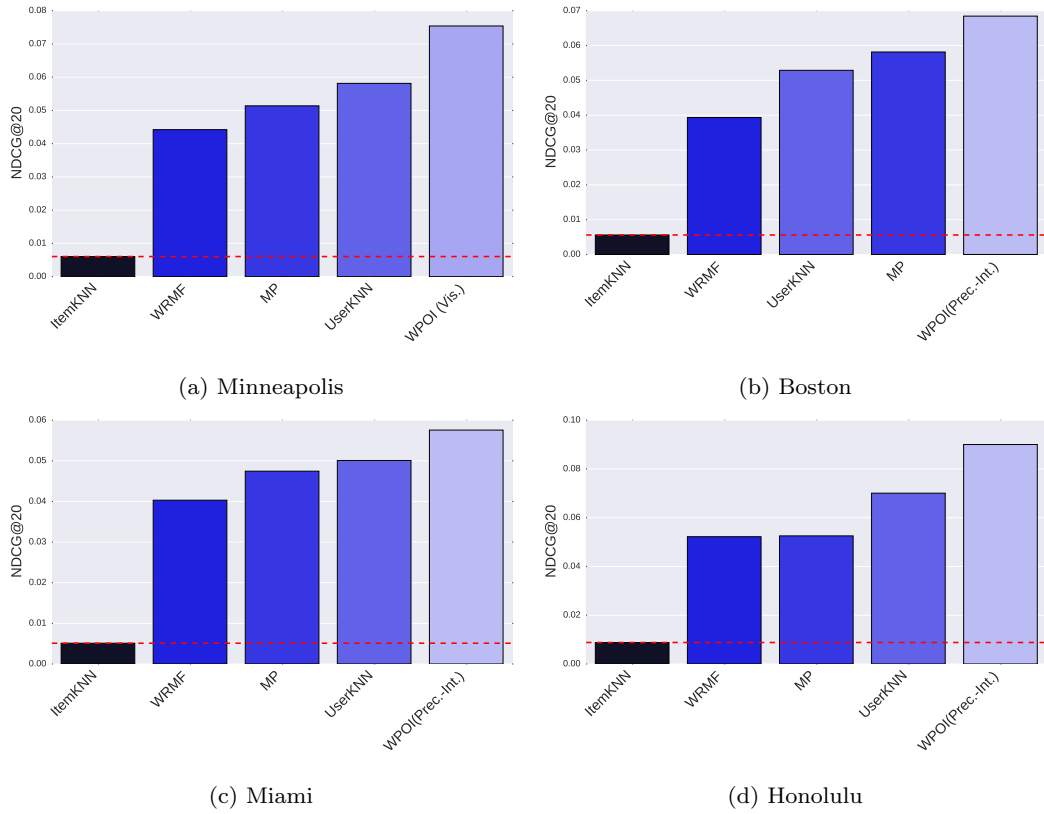


Fig. 12: Comparing WPOI with the strongest weather feature in the respective city to state-of-the-art recommender systems algorithms. As presented, there are observable differences between the methods, while WPOI outperforms the other methods.

ventional matrix factorization methods such as WRMF and CF approaches as shown in our experiment.

7 Summary & Discussion

The main findings with respect to our RQs can be summarized as follows:

- **RQ1.** The analyses in Section 4 show that the weather context does indeed have a significant impact on user check-in behaviour, presenting different check-in profiles for different kinds of contextual weather variables and places.
- **RQ2.** Furthermore, our work reveals how to build a weather-aware POI recommender system (WPOI) by extending a POI state-of-the-art POI matrix factorization method.
- **RQ3.** Finally, our study shows that the weather context can significantly increase the recommender accuracy of a POI recommender method, outperforming even the time context in some cases. Among the considered weather features, ‘precipitation intensity’ and ‘visibility’ are the most significant ones

in use to improve the ranking in a weather-aware POI recommender system.

Taken together, our results show that weather context has a significant impact on user check-in behaviour and movement within cities, which can be exploited to significantly improve the accuracy of a POI recommender system. Not only does this research have implications for the design of future technology, for instance, mobility apps for more efficient movement of tourists in cities, but it will also hopefully lead to further studies that help to better understand mobility patterns and the behaviour of users in cities in general and tourists in particular.

In our study, we relied only on actual weather conditions in place, rather than on weather forecasts. One might argue that this is a limitation of this work, as people often plan their trips beforehand by looking at the weather forecast for a particular day. However, as we also know, forecasts are not always reliable and weather conditions can change (also rapidly). As such, people need to adjust their planning accordingly and consider other alternatives. Furthermore, our study only focused on the city area and rather short check-in distances

(max. 100km). Hence, the influence on planning travel or trips on forecasts, rather than the actual weather conditions in place, might be smaller than assumed. The question of how far in advance people plan their trips to cities is an interesting question that we hope to look at more closely in our future work.

Currently, our work only investigates one weather feature at a time. Investigating different hybridization methods would also be an interesting focus in future work. Comparing the method developed in this paper with other context-aware recommender strategies such as contextual sparse linear models (CSLM) [33] employing the CARSKit⁴ framework is also of interest. A limitation to our work may be that we performed our recommender analyses on a relatively small sample of cities, albeit on a huge number of POIs. As such, it would be interesting to further study the performance of our WPOI approach in even more cities and across different countries.

What our analyses revealed, in the context of distances between POIs, is that people tend to check into POIs that are closer in distance to each other, rather than POIs that are farther away when compared to the users previously checked-in at POIs. Furthermore, it was shown that travels are weather dependent. One limitation of this investigation is that we did not apply a time filter. For instance, looking at a sequence of POIs with check-ins within a certain time frame (1hr, 12hrs or 24hrs) may show a different movement behaviour. Furthermore, we limited our investigations exclusively to POIs in cities and with a maximum distance of 100km from the previously checked-in POI. Considering travels beyond the city level, for example, between cities and countries, may also lead to interesting new insights.

In addition, we would like to extend our investigations on user, POI and POI category level. Our current study shows, for instance, that there are significant differences in terms of check-in behaviour at the category level. Furthermore, preliminary work also reveals that there are users, who are more sensitive to some of the weather features as opposed to others. Finally, at the category level, it would be interesting to study different check-in patterns related to certain types of POIs, e.g., indoor and outdoor POIs. A better understanding of city mobility patterns would not only be of particular interest for governmental bodies (e.g., to provide better public transportation), but also for tourism stakeholders, who are, for example, engaged in city tour planning activities, etc.

8 Conclusions

This paper investigated the utility of weather context for point of interest (POI) recommender systems. Our work showed that (i) weather context does indeed have a significant impact on user check-in behaviour in cities and that (ii) we can significantly improve state-of-the-art POI recommender methods by incorporating weather context into models, wherein ‘precipitation intensity’ and ‘visibility’ are the most useful features in the process. In our opinion, this research provides a baseline for a range of further studies related to recommender systems and tourism, using weather as a context variable.

Open Science. In order to make the results obtained in this work reproducible, we have shared the code and data used in this study. The proposed method Rank-GeoFM with weather context is implemented within the MyMediaLite framework [10] and can be downloaded for free from our GitHub repository⁵. Furthermore, the data samples used in the experiments can be requested via email from the corresponding authors of this study.

Acknowledgements

We thank the reviewers for their valuable comments. Furthermore, we would like to acknowledge Prof. Rodrygo L. T. Santos, who provided us with useful feedback to improve the model section.

The authors Denis Parra and Leandro Marinho were supported by CONICYT (project FONDECYT 11150783) and the EU-BR BigSea project (MCTI/RNP 3rd Coordinated Call).

References

1. J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12*, pages 199–208, New York, NY, USA, 2012. ACM.
2. J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. Recommendations in location-based social networks: A survey. *Geoinformatica*, 19(3):525–565, July 2015.
3. J. Borras, A. Moreno, and A. Valls. Intelligent tourism recommender systems: A survey. *Expert Systems with Applications*, 41(16):7370–7389, 2014.
4. L. Bottou. *Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics Paris France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, chapter Large-Scale Machine Learning with Stochastic Gradient Descent, pages 177–186. Physica-Verlag HD, Heidelberg, 2010.

⁴ <https://github.com/irecsys/CARSKit>

⁵ <https://github.com/aoberegg/WPOI>

5. M. Braunhofer, M. Elahi, M. Ge, F. Ricci, and T. Schievenin. STS: design of weather-aware mobile recommender systems in tourism. In *Proceedings of the First International Workshop on Intelligent User Interfaces: Artificial Intelligence meets Human Computer Interaction (AI*HCI 2013) A workshop of the XIII International Conference of the Italian Association for Artificial Intelligence (AI*IA 2013), Turin, Italy, December 4, 2013.*, 2013.
6. M. Braunhofer, M. Elahi, F. Ricci, and T. Schievenin. Context-aware points of interest suggestion with dynamic weather data management. In *Information and communication technologies in tourism 2014*, pages 87–100. Springer, 2014.
7. C. Cheng, H. Yang, I. King, and M. R. Lyu. Fused matrix factorization with geographical and social influence in location-based social networks. In *Proc. of AAAI*, pages 17–23, 2012.
8. G. Ference, M. Ye, and W.-C. Lee. Location recommendation for out-of-town users in location-based social networks. In *Proceedings of the 22Nd ACM International Conference on Information & Knowledge Management, CIKM '13*, pages 721–726, New York, NY, USA, 2013. ACM.
9. D. R. Fesenmaier, T. Kuflik, and J. Neidhardt. Rectour 2016: Workshop on recommenders in tourism. In *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys '16*, pages 417–418, New York, NY, USA, 2016. ACM.
10. Z. Gantner, S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. MyMediaLite: A free recommender system library. In *In Proc. of RecSys'11*, 2011.
11. H. Gao, J. Tang, X. Hu, and H. Liu. Exploring temporal effects for location recommendation on location-based social networks. In *Proceedings of the 7th ACM Conference on Recommender Systems, RecSys '13*, pages 93–100, New York, NY, USA, 2013. ACM.
12. D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou. Mobile recommender systems in tourism. *Journal of Network and Computer Applications*, 39:319–333, 2014.
13. M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.
14. T. Horanont, S. Phithakkitnukoon, T. W. Leong, Y. Sekimoto, and R. Shibasaki. Weather effects on the patterns of people's everyday activities: a study using gps traces of mobile phone users. *PloS one*, 8(12):e81153, 2013.
15. Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. of ICDM'08*, pages 263–272. Ieee, 2008.
16. T. Kohyama and J. M. Wallace. Rainfall variations induced by the lunar gravitational atmospheric tide and their implications for the relationship between tropical rainfall and humidity. *Geophysical Research Letters*, 43(2):918–923, 2016. 2015GL067342.
17. Y. Koren. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09*, pages 447–456, New York, NY, USA, 2009. ACM.
18. 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 *Proc. of SIGIR'15*, pages 433–442, New York, NY, USA, 2015. ACM.
19. A. Q. Macedo, L. B. Marinho, and R. L. Santos. Context-aware event recommendation in event-based social networks. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys '15*, pages 123–130, New York, NY, USA, 2015. ACM.
20. D. Martin, A. Alzua, and C. Lamsfus. *A Contextual Geofencing Mobile Tourism Service*, pages 191–202. Springer Vienna, Vienna, 2011.
21. K. Meehan, T. Lunney, K. Curran, and A. McCaughey. Context-aware intelligent recommendation system for tourism. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, pages 328–331. IEEE, 2013.
22. A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Conference on Social Computing (SocialCom)*, pages 144–153, Sept 2012.
23. I. Nunes and L. Marinho. A personalized geographic-based diffusion model for location recommendations in lbsn. In *Proceedings of the 2014 9th Latin American Web Congress, LA-WEB '14*, pages 59–67, Washington, DC, USA, 2014. IEEE Computer Society.
24. D. Parra and S. Sahebi. Recommender systems: Sources of knowledge and evaluation metrics. In J. D. Velsquez, V. Palade, and L. C. Jain, editors, *Advanced Techniques in Web Intelligence-2*, volume 452 of *Studies in Computational Intelligence*, pages 149–175. Springer Berlin Heidelberg, 2013.
25. B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. of WWW'01*, pages 285–295. ACM, 2001.
26. C. Trattner, A. Oberegger, L. Eberhard, D. Parra, L. B. Marinho, et al. Understanding the impact of weather for poi recommendations. In *RecTour@ RecSys*, pages 16–23, 2016.
27. Q. Wang and J. E. Taylor. Quantifying human mobility perturbation and resilience in hurricane sandy. *PLOS ONE*, 9(11):1–5, 11 2014.
28. Q. Wang and J. E. Taylor. Resilience of human mobility under the influence of typhoons. *Procedia Engineering*, 118(Supplement C):942 – 949, 2015. Defining the future of sustainability and resilience in design, engineering and construction.
29. Q. Wang and J. E. Taylor. Patterns and limitations of urban human mobility resilience under the influence of multiple types of natural disaster. *PLOS ONE*, 11(1):1–14, 01 2016.
30. D. Yang, D. Zhang, and B. Qu. Participatory cultural mapping based on collective behavior in location based social networks. *ACM Transactions on Intelligent Systems and Technology*, 2015. in press.
31. M. Ye, P. Yin, W.-C. Lee, and D. L. Lee. Exploiting geographical influence for collaborative point-of-interest recommendation. In *Proc. of SIGIR'11*, pages 325–334. ACM, 2011.
32. H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen. Lcars: a location-content-aware recommender system. In *Proc. of KDD'13*, pages 221–229. ACM, 2013.
33. Y. Zheng, B. Mobasher, and R. Burke. Cslim: Contextual slim recommendation algorithms. In *Proceedings of the 8th ACM Conference on Recommender Systems*, pages 301–304. ACM, 2014.