

Cicerone: Design of a Real-Time Area Knowledge-Enhanced Venue Recommender

Daniel Villatoro, Jordi Aranda, Marc Planagumà, Rafael Gimenez and Marc Torrent-Moreno

Barcelona Digital Technology Centre, Barcelona, Spain

{dvillatoro,jaranda,mplanaguma,rgimenez,mtorrent}@bdigital.org

Abstract

Smart-devices with information sharing capabilities *anytime and anywhere* have opened a wide range of ubiquitous applications. Within urban environments citizens have a plethora of locations to choose from, and in the advent of the smart-cities paradigm, this is the scope of location-based recommender systems to provide citizens with the adequate suggestions. In this work we present the design of an *in-situ* location-based recommender system, where the venue recommendations are built upon the users' location at request-time, but also incorporating the social dimension and the expertise of the neighboring users knowledge used to build the recommendations. Moreover, we propose a specific easy-to-deploy architecture, that bases its functioning in the participatory social media platforms such as Twitter or Foursquare. Our system constructs its knowledge base from the accessible data in Foursquare, and similarly obtains ratings from geopositioned tweets.

1 Introduction

Urban environments host a plethora of interesting locations such as restaurants, shops, museums, theaters and a wide range of other venues that neither can be known by all users nor they might be interested in visiting all. However, each citizen can potentially become an expert of the neighborhood he visits more often or lives, as he will know, and maybe have visited, more venues in such area. Therefore, it is straightforward to see how for an specific citizen might not be a problem to find an adequate venue for his taste on his neighborhood of expertise, but it potentially becomes cumbersome to do the same task when in a different less-known neighborhood. It becomes then a problem for citizens to find locations they might enjoy when away of their area of expertise. The problem of finding adequate items for specific users is that classically solved by recommender systems.

In our case, we focus on location-based recommender systems [Zheng *et al.*, 2009; Park *et al.*, 2007], where users are recommended locations to visit expecting to maximize users' satisfaction. These type of recommenders complement the

previously analyzed ones as they also have to take into account the context, distance from the user to the recommended venue, and maybe several other factors.

With the penetration of smart-devices, users have the possibility to access information anytime anywhere, and the system we present in this work profits from those ubiquitous computing capabilities; our on-site location-based recommender system allows users to obtain the most adequate venue with respect to their current position. Our approach profits from a different dimension of the users' parameters space, namely their social relationships and their relative geographical knowledge with respect to the location of the items. This model provides an alternative solution to the problem of providing personalized recommendations in a geospatial domain: user expertise in this type of domain conveys an implicit continuum knowledge of the surrounding geospatial area and the locations within that area. Our solution intelligently combines this user geospatial knowledge to the classical social distances amongst users used in state-of-the-art recommenders.

Our system personalizes recommendations of locations not only considering the past history of a specific user, but also (1) the current location of the user, (2) the social distance with other similar users and (3) their expertise in the area where the recommendation is going to be provided. This aggregation function basically expresses a tendency of a user to visit a certain location given its distance to the location, and the past history of the user and its friends and their knowledge of the area.

To the best of our knowledge, there are no existing recommender systems that profit from the inherent characteristics of the geographical location, such as continuity in space, user's area expertise or word-of-mouth location suggestions, to generate recommendations to users.

2 State of the Art

As we have discussed previously, the problem of finding adequate venues for citizens to visit is a problem already tackled by the recommender community, under the location-based recommender systems [Zheng *et al.*, 2009; Park *et al.*, 2007]. Despite the impressive amount of literature in such area, this is still an open problem, even for those with access to complete datasets and user profiles [Sklar *et al.*, 2012], as new methods and algorithms are being proposed to boost their ef-

iciency.

In this work however, we propose the integration of social information into the calculation of the recommendations. Some authors have investigated the potential of the explicit inclusion of information of user’s relationships from social networks to generate the neighborhood used in classical collaborative filtering (CF) algorithms (*social filtering*), improving the results obtained by the classic CF in the analyzed scenarios [Groh and Ehmig, 2007].

Others [Bonhard and Sasse, 2006] have analyzed how the relationship between advice-seeker and recommender is extremely important in the user-centered recommendations, concluding that familiarity and similarity amongst the different roles in the recommendation process aid judgement and decision making. As well as in our approach, some researchers have considered the important role of *experts* [Amatriain *et al.*, 2009; Bao *et al.*, 2012], however in our case, these experts are calculated automatically for each specific area of the city and weighted with respect to the social distance amongst the advice-seeker and recommender.

A similar recommendation approach is presented [Ye *et al.*, 2010], where authors also propose the usage of Foursquare information to provide venue recommendations to users; more importantly the social perspective is integrated into their recommendations, developing a *Friend-Based Collaborative Filtering* (where the neighbours for CF are selected from the social network of users), and an extension of this method *Geo-measured Friend-Based Collaborative Filtering* (where only closely located friends are selected as neighbours for CF).

Our method then proposes a combination of the *Geo-measured Friend-Based Collaborative Filtering* [Ye *et al.*, 2010] and *experts* [Amatriain *et al.*, 2009; Bao *et al.*, 2012], in our specific case, neighborhood or area experts.

3 Cicerone Recommender System

In this section we provide the theoretical framework of the Cicerone location-based recommender system. Firstly we describe the basic terminology used later in the recommendation algorithm. As we have sketched previously, our system bases its functioning in three information elements: the users’ social network, the users’ area knowledge and the current location of the requesting user.

3.1 Basic Terminology

As used herein, the term “location data item” stands for any location item or representation of a location. A “location item” is intended to encompass any type of location which can be represented in a map using a latitude, a longitude, and possibly a category.

The location recommender may be capable of selecting relevant locations for a given target user. To do so, users should be comparable entities and locations as well. It should be understood that the implementations described herein are not item-specific and may operate with any other type of item visited/shared by a community of users. For the specific case of bars or restaurants items, users may interact with the items by visiting them. The Recommendations Set is the locations set

formed by the items the user is being recommended. A User A ’s Recommendations set will be denoted herein as R_A .

An essential concept is the one of “Check-in” ($CI_{U,L}^t$) which represents the attendance¹ of a user U to a certain location L in the last t days, and therefore. Our system will generate Location recommendations to the users by considering not only its geographical position, but also its social relationship with other users and their degree of knowledge of the visited locations.

In order to obtain more adequate recommendations in this type of environments, we envision the necessity of certain estimators. Firstly, we need to quantify how well a certain user knows a specific area (by considering the attendance frequency to locations in such area with respect to the rest of the city). Moreover, it becomes necessary to understand the social distance between the target user and the other users, whose opinions are being used to create the recommendations for the target user.

These measures are clearly described and specified next: The **Area Knowledge** ($AK_{U,L}^t$) of a user U with respect to a location L is calculated:

$$AK_{U,L}^t = \frac{\sum_{l' \in \text{PostalCode}_L} CI_{U,l'}^t}{\sum_{\forall a \in \text{Locations}} CI_{U,a}^t} \quad (1)$$

and represents how familiar a user is within an specific area of the city (represented by its postal code).

The **Location Frequency** ($LF_{U,L}^t$) of a user U in a certain location L is calculated:

$$LF_{U,L}^t = \frac{CI_{U,L}^t}{\sum_{\forall a \in \text{Locations}} CI_{U,a}^t} \quad (2)$$

and normalizes the number of visits of the user U to the location L .

The **Social Importance** ($SI_{U,U'}^t$) of a user U' for a user U is calculated:

$$SI_{U,U'}^t = \frac{(\text{Degree}_{U'})^{\frac{1}{d(U,U')}}}{(\text{nodes} - 1)} \quad (3)$$

where Degree_U represents the number of connections that U has in its social network, nodes represents the total number of nodes in the social network (and used to normalize the SI), and $d(U, U')$ represents the geodesic distance, i.e. minimum number of hops necessary to reach U' from U using the shortest path in their social network².

The **Location Value** ($LV_{L,U}^t$) of a location L for a user U at time t is calculated:

$$LV_{L,U}^t = \frac{\sum_{\text{users in } L} (LF_{U,L}^t \times AK_{U,L}^t \times SI_{U,U'}^t)}{|\text{users}|} \quad (4)$$

¹The attendance of a user to a certain location can be captured in several ways, for example, a Foursquare Check-in, a geopositioned tweet, or a CDR trace of a phone call.

² $d(U, U') < 0$ means that there is no possible path that connects U and U'

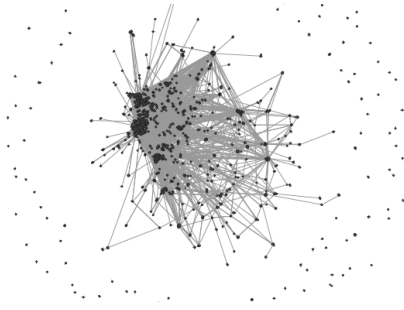


Figure 1: Twitter-sensed Barcelona Social Network

where $|users|$ represents the number of users that have “checked-in” to that Location.

The resulting value basically aggregates the information of surrounding detected locations considering the social distance of our specific user to the users that visited that location (social information), and their familiarity in the area of the location (geographical information).

3.2 Cicerone Recommendation Algorithm

The general algorithm for the functioning is the following:

1. Once the position of a user A is detected, the system automatically captures its Latitude and Longitude and launches the process that builds the personalized recommendation set for that position and user at that certain moment.
2. The system retrieves all the locations in 100m radius of the current position.
3. The recommendation set for user A, R_a , is a set constructed with all the locations in 100m radius of the user current position.
4. The system calculates the location value of each of the locations in that set.
5. The system orders the retrieved locations according to the calculated Location Values and constructs the Recommendation Set with the 3 with a highest value.

4 Functional implementation of Cicerone

As explained previously, the theoretical framework to build the recommendation needs from a number of data sources, namely, users locations, venues and the social relationships amongst users. As this recommendation process is envisioned to be executed when users are *in-situ*, the main functional requirement for our system is to work from a mobile device. Working prototypes have decided to opt for the development of a dedicated app (such as Yelp, TimeOut or TripAdvisor), that users have to download within their devices. The app provides several advantages as the explicit user profiling as well as the definition of the necessary information to obtain the recommendation. However, for us it implies the big problem of reaching a critical mass of users that would made the knowledge base and the recommendation more accurate. To avoid this limitation we have opted to develop our system as

a service embedded within already massive social networks. Twitter seems to be the ideal candidate for us for the following reasons:

- Twitter shows a widespread uniform penetration almost worldwide, with an continuously increasing numebr of users (288 million monthly active users in July 2012, showing an increase of 40% since July 2009 [Global-WebIndex, 2013]).
- It allows users to associate their location when posting a message, and associate the specific coordinates as meta-information.
- It provides developers with an accesible API to obtain in near real-time the publicly published tweets.
- Twitter captures a social network of followers and followings, publicly available for each user.

As we have initially decided to deploy our application in the city of Barcelona, Twitter confirms to be an ideal candidate. The number of captured geopositioned tweets daily within Barcelona is 6200 (from data coming from 2012), and the social network inferred from the users posting them can be seen in Figure 1 (with an average degree of 2.93).

Similarly, and to populate our items database, we opt to use the crowdsourced database of Foursquare. Foursquare is a location-based social media platform to communicate the venues a user is in. This platform allows users to input into their databases new locations, by introducing not only the venue’s name and specific location (with the GPS position and postal address) but also a semantic category. Foursquare describes the places according to a rather complete taxonomy, where about 400 kinds of places are identified and grouped in 9 wide categories³. Foursquare provides an accesible API allowing us to take snapshots of the existing locations in a certain city. Within the city of Barcelona, from a snapshot taken April 2013, we have detected over 66.000 foursquare locations, uniformly distributed amongst the different districts.

Moreover, within the OpenData movement, the city of Barcelona provides a machine-readable administrative division necessary for our theoretical calculations (namely the District divisions).

5 System Architecture

In this section we will describe the system architecture (sketched in Figure 2) needed to implement a functional instantiation of the theoretical framework previously described. Firstly we will describe the social network monitoring used as data input for our platform, and also as user interface interaction with the recommendation engine. After that, we will sketch the persistence infrastructure used to save the information related to venues and users, and finally we will describe the information update process and the component needed to develop the recommendation platform.

³The detailed categorization of Foursquare categories and parent categories can be found at <http://aboutfoursquare.com/foursquare-categories/> (Last access April 2nd 2013).

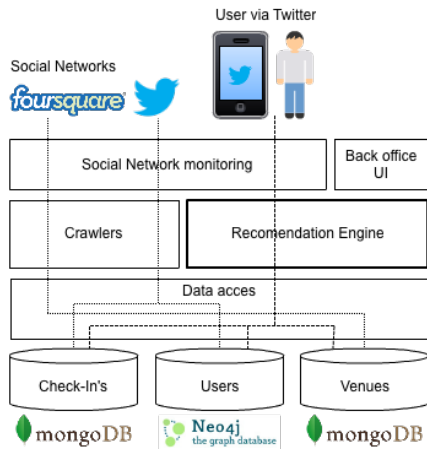


Figure 2: Cicerone Architecture

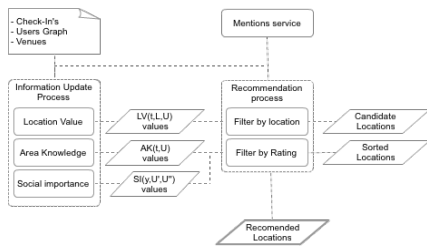


Figure 3: Cicerone Workflow

5.1 Social Networks Monitoring: Sensing the City

The usage of social media platforms in our system are twofold: (1) information acquisition to feed the knowledge base of our platform, and (2) a channel for users' interaction with our technology.

The participatory information provided by users in Foursquare will be used to populate our items database; similarly, we will use geopositioned tweets to calculate users' *Area Knowledge*. Therefore, the social network monitor is the first layer of our architecture and it is composed by two crawlers: a Foursquare crawler and a Twitter crawler. The Foursquare crawler is in charge to scan the target city for new venues. Once a new venue is identified, it is stored in the items database with its associated metainformation such as its specific coordinates, the address or the category. The Twitter crawler is in charge to capture all the tweets generated in the target city. Its scope is threefold: (1) build and update the users' social network, (2) update the user area knowledge using its geopositioned tweets, and (3) permits users' communication with the system.

Moreover, and given its popularity, we use Twitter as the communication channel of our recommender system through a bot account managed by our intelligent agent.

5.2 Persistence Infrastructure: Urban data Model

Any recommender system bases its functioning in three main elements: users, items and ratings. These three elements have to be stored according to the inherent properties of the system, which in this case, imply real-time information access

and update. Fed by the crawlers, the data required for our recommendation solution arrives to the persistence manager and each of these elements are stored in a persistence infrastructure in the following way: *Users*: One of the main functional requirements of the recommendation algorithm is the access to the social network of users. In order to effectively store this information, we opt for using a graph-oriented database, namely *Neo4j*⁴. These type of databases allow us to persist users' social network in the form of a directed weighted graph. In this database, we persist users as nodes and then establish edges amongst nodes if there exist a social relation amongst them. Consequently, an edge between two nodes is created if there exists a social relation amongst them, according to users' Twitter profiles; specifically, an edge is created amongst from user *A* towards user *B* if user *A* follows user *B* in Twitter. At this edge level, the edge's weight will be defined depending on the users interactions: different types of Twitter interactions (such as mentions, retweets or favoured) will affect the weight differently.⁵ Another important information about users is saved, namely his "Check-ins" (as described in Sec.). These "check-ins" (the specific coordinates of each user geopositioned tweet) is stored into a *MongoDB*⁶, as we can profit from the implemented geo-spatial index.

Items: The items in our system are the locations within the target city. The locations database needs to provide efficient information access, as the recommender algorithm needs a high average number of accesses to it to build the recommendations. Moreover, an important item's characteristic is its location within space, that is profited from when using geo-spatial indexing. Given these two characteristics (rapid information access and geo-spatial indexing), as well as the potential for distributed computing, we opt to implement this database using *MongoDB*.

Ratings: The notion of rating is classically treated as an explicit evaluation of users about an item. However, in this work, we take an alternative approach for ratings: we consider as a constant rating value the users presence in a location, sensed through the geopositioned tweets posted from or close to the venue location. This value is not obtained directly, the Ratings will be part of knowledge obtained by the recommendation engine and their information update process capturing user's visits to specific locations but this will be explained on the Section 5.3.

5.3 Recommendation Engine: Information Update Process

The last component in our proposed architecture is the recommendation engine containing the implementation of the theoretical algorithms previously explained in the Section 3. Once we have our social networks monitor as an urban data sensor, and the ability to persist all the raw data required by the system, this component will be the responsible of the knowledge extraction process and the business logic triggered

⁴<http://www.neo4j.org>

⁵Specific values and functions for edge weight determination will be developed at later versions of our software using empirical information.

⁶<http://www.mongodb.org>

to generate a recommendation.

Because of the real-time aspect of our system, our recommendation platform (whose workflow is detailed in Figure 3) needs to continuously update some information elements such as users' *Area Knowledge* and *Location Frequency*, the creation or update of social relationships amongst users or the appearance of new locations. Specifically, we envision the users' communication with our system through a Twitter personality that encapsules our recommendation platform; everytime a user mentions our system's username, the platform will capture this tweet (through the Mention's Service sketched in Figure 2) and identify it as an explicit request for a recommendation that will trigger the whole intelligent process. Eventhough our technological platform allows us to generate recommendations everytime a user's location is captured (with every geopositioned tweet), we rather restrict its functioning with a mention system reducing the overall intrusiveness.

After the recommendation is generated, it is returned to the user also through Twitter with a message posted by our intelligent agent.

6 Conclusion and Future Work

The designed recommender system plans to profit from the information proactively shared by users in the analyzed participatory platforms. However, as recently argued in [Jeske, 2013], these type of crowdsourced systems is sensible to malicious attacks: in our case, and given the lack of restrictions to post geo-positioned content from Twitter, someone could easily envision the method to create a fake user to become the one with higher area knowledge in every area of the city, and then influence directly the resulting recommendations to his own will.

Despite this potential problem associated to the publishing policy of Twitter and Foursquare, and as we have analyzed in Sec. 2, many others have used information from these sources to generate location-based recommendations. However, and to the best of our knowledge, the presented algorithm is the first to include explicitly the user's expertise about one of the fundamental properties of the items: the area where it is located. By combining this information, with some social information, we hypothesize that our system will be able to outperform other location-based recommender systems.

Our main long term research task to be performed is the development of a user profiling in term of the type of venues the user attends to, with the overall objective of combining the area expertise and with specific user profiles.

Acknowledgments

This work has been completed with the support of ACCIÓ, the Catalan Agency to promote applied research and innovation.

References

[Amatriain *et al.*, 2009] Xavier Amatriain, Neal Lathia, Josep M. Pujol, Haewoon Kwak, and Nuria Oliver. The wisdom of the few: a collaborative filtering approach

based on expert opinions from the web. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 532–539, New York, NY, USA, 2009. ACM.

[Bao *et al.*, 2012] Jie Bao, Yu Zheng, and Mohamed 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.

[Bonhard and Sasse, 2006] P. Bonhard and M. A. Sasse. 'knowing me, knowing you' – using profiles and social networking to improve recommender systems. *BT Technology Journal*, 24(3):84–98, July 2006.

[GlobalWebIndex, 2013] GlobalWebIndex. Twitter now the fastest growing social platform in the world. Web Report, Jan 2013.

[Groh and Ehmig, 2007] Georg Groh and Christian Ehmig. Recommendations in taste related domains: collaborative filtering vs. social filtering. In *Proceedings of the 2007 international ACM conference on Supporting group work*, GROUP '07, pages 127–136, New York, NY, USA, 2007. ACM.

[Jeske, 2013] Tobias Jeske. Floating car data from smartphones: What google and waze know about you and how hackers can control traffic. <https://media.blackhat.com/eu-13/briefings/Jeske/bh-eu-13-floating-car-data-jeske-wp.pdf> (Last access April 1st 2013)., 2013.

[Park *et al.*, 2007] Moon-Hee Park, Jin-Hyuk Hong, and Sung-Bae Cho. Location-based recommendation system using bayesian users preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, pages 1130–1139. Springer, 2007.

[Sklar *et al.*, 2012] Max Sklar, Blake Shaw, and Andrew Hogue. Recommending interesting events in real-time with foursquare check-ins. In *Proceedings of the sixth ACM conference on Recommender systems*, RecSys '12, pages 311–312, New York, NY, USA, 2012. ACM.

[Ye *et al.*, 2010] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '10, pages 458–461, New York, NY, USA, 2010. ACM.

[Zheng *et al.*, 2009] Yu Zheng, Yukun Chen, Xing Xie, and Wei-Ying Ma. Geolife2. 0: a location-based social networking service. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. Tenth International Conference on*, pages 357–358. IEEE, 2009.