# Collaborative Recommendation of Photo-Taking Geolocations

Thomas Phan, Jiayu Zhou, Shiyu Chang, Junling Hu, and Juhan Lee
Samsung Research America – Silicon Valley
San Jose, CA

## ABSTRACT

We apply collaborative recommendation algorithms to photography in order to produce personalized suggestions for locations in the geocoordinate space where mobile users can take photos. We base our work on a collection of 3 million geotagged, publicly-available Flickr.com digital photos on which we applied a series of steps: first, unique locations are identified by discretizing the continuous latitude and longitude geocoordinates into geographic virtual bins; second, implicit feedback is calculated in a $user \times location$ matrix using normalized frequency; and third, missing feedback values are imputed through four different algorithms (one memory-based and three model-based). Our results show that two of the model-based algorithms produced the best RMSE and that the RMSE is sensitive to increasing hash bin size.

## Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*; H.2.8 [**Database Management**]: Database Applications—*Data mining*

## Keywords

geotagging; mobile photography; location recommendation; recommender system; collaborative filtering

## 1. INTRODUCTION

Digital photography empowers users, particularly tourists, to capture visual memories of visited locations and review them with immediate gratification. Further, mobile photography using smartphones with built-in digital cameras is a fast-growing market segment, with estimates stating that its sales growth is outpacing even traditional point-and-shoot cameras [7]. Although having a camera on hand is advantageous, taking interesting photos is not always simple, and many photographers may find inspiration from recommendations for either nearby or afar locations that are amenable to taking good photos.


*GeoMM'14,* November 7, 2014, Orlando, FL, USA.

http://dx.doi.org/10.1145/2661118.2661121.

Recommendations for such locations can come from a variety of sources, such as word-of-mouth from family and friends, social networks, tourist guidebooks, or simply seeing a scenic photo in a magazine. With modern digital photography, popular locations can be determined by aggregating photo geotags, i.e. the latitude and longitude geocoordinates that are embedded into photo metadata such as the EXIF header of JPEG images. As a result, visualizations such as heatmaps can illustrate the most popular locations where photos are taken [13]; invariably, well-known tourist locales like New York City, New York, will be identified. While popular locations are a good starting point for taking photos, not everyone has the same taste; just as a user's interest in books and movies may differ substantially from the most popular selections, so too may a user's interest in locations to take photographs.

In our work we thus looked to generate more *personalized* suggestions for photo-taking locations by applying recommender system techniques. Such systems are commonly used by e-commerce sites to predict which products may be of interest to users, and in particular, systems that employ collaborative filtering leverage the collective history across multiple users in order to generate candidate recommendations [9, 22]. In our context, we apply collaborative filtering on a collected data set of 3 million geotagged photos that were taken specifically with smartphones. All these photos were publicly available from the Flickr.com photo-sharing website.

Previous researchers have also sought to create geolocation recommendations using data sources such as raw GPS traces or "check-ins" from social networks, but all have held different assumptions, such as inferred user behavior [29, 16], inferred centers of activity [28], or inferred physical addresses [19]. Further, other efforts using geotagged media avoid the geocoordinate system by basing the location only on reverse-geocoded physical point-of-interest (POI) names and addresses such as those for major landmarks, museums, hotels, and other named places.

We keep our system in the geocoordinate space for two reasons. First, for many landmarks, there may be many different unnamed spots where photos can be taken; for example, the Golden Gate Bridge in San Francisco and the Cloud Gate sculpture in Chicago have been photographed from many different directions and distances because users have different tastes for photographing them, such as close up, overseeing a waterfront, or with an urban skyline in the background. As such, recommending a coarse-grained canonicalized POI name, such as "Golden Gate Bridge", is not as helpful to a photographer as a finer-grained latitude

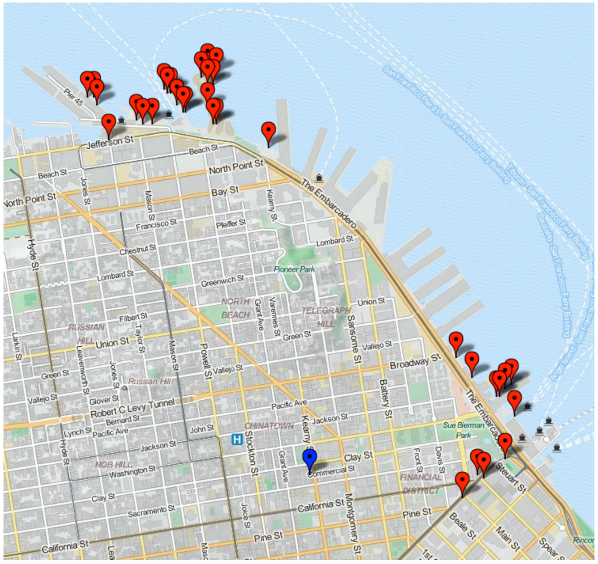


**Figure 1: An example user's photos are plotted in red, and a recommendation is shown in blue.**

and longitude geocoordinate. Second, photos may be taken at ad hoc locations that are far away from any POI landmarks, such as on beaches or inside large national parks. For both of these reasons, our system seeks to identify and recommend spots in the geocoordinate space as an aid to photographers.

Recommending such locations for photography poses several technical challenges that we addressed through a series of geospatial data-processing steps. To first identify unique locations, we used a cartographic hashing function to discretize the continuous latitude and longitude geocoordinates from our set of 3 million photos into geographic virtual bins of variable size. Each bin's hash key then serves as a columnar identifier within a co-occurrence matrix containing $users \times locations$, where the value of each cell is a normalized frequency that serves as an implicit feedback given by the users for the locations. Given this matrix, we evaluated four algorithms: a memory-based item-item algorithm and three model-based methods (non-negative matrix factorization, probabilistic matrix factorization, and a low-rank factorization). Our results show that two of the model-based algorithms produced the best RMSE and that the RMSE is sensitive to increasing hash bin size.

The rest of this paper is organized in the following manner. In Section 2 we review related work, and in Section 3 we describe our design methodology. We discuss our experiments in Section 4 and conclude in Section 5.

## 2. RELATED WORK

Recommender systems are often used by online e-commerce companies to suggest items for their users to purchase [18, 1, 20]. To the best of our knowledge, ours is the first work that applies recommendation algorithms to suggest real-world locations in the geocoordinate space where users can take photos based on collaborative filtering using a corpus of geotagged photos.

In our work we use a sample of publicly-available photos from Flickr.com, a popular photo-sharing website hosted in the USA. Other photos-sharing sites include Panoramio.com. Since these sites provide an API to access their millions of stored photos and metadata, they are a rich source of user-generated content that have been used by many data-mining researchers. Previous work using such data include classifying locations from image content [5], identifying people in photos to suggest travel [3], suggesting how to compose a shot [2], and determining the presence of tourists [8].

Prior efforts have also looked at geospatial data to make tourism recommendations; however, their work differs from ours in important assumptions. Some efforts used raw user mobility GPS traces, but since the collected points are diffuse, approaches have been developed to find "stay-points" [28, 29] to infer where activities (such as shopping or dining) may have occurred. In our work, we do not need to make any inferences about the user location because our geotagged photo data explicitly identifies a geocoordinate. Other researchers use data sets that identify a user's location but not the user's activity or intent, so an inference must be drawn from external data, such as by performing information extraction from social networks or digital journals [29, 16]. We again do not need to draw any inferences because the user's intent for photography is explicit in our data set.

Finally, previous researchers have used geolocation data containing POI names or addresses [6, 4, 16, 25] rather than using latitude and longitude; for example, the physical landmark location name of "Golden Gate Bridge" (or its postal address) is commonly used in lieu of the latitude and longitude geocoordinate pair of 37.813, -122.478. However, our work remains in the geocoordinate space through the use of our cartographic hashing scheme because (1) photos of a given POI may be taken from different locations and (2) photos may be taken at impromptu locations far away from named addresses.

## 3. DESIGN AND IMPLEMENTATION

### 3.1 User-facing application

The goal of our system is to generate relevant recommendations for geospatial locations (in the form of latitude and longitude geocoordinates) where a user can go take photos. Figure 1 shows our application's user interface displaying a result based on real user data. Here, a user's previously-taken photos along the northern pier of San Francisco, California, are plotted as red points. These geocoordinates are known because the user's smartphone embedded them into the photo metadata header, producing a geotag. Our system then generated a list of recommended photography geolocations for the user which we then prune to be recommended spots within some distance $d$ of the user's current location; the user can adjust this value, which we set by default to be 5 miles / 8 km. In this example, one such recommendation is visualized as a blue point, which upon inspection turns out to be a tourist locale in downtown San Francisco's Chinatown district.

We note here several points of our system. First, the recommendations are always for locations where the user has never visited. Second, as we discuss later, we identify unique locations using rectangular bins; while the red points are precise geocoordinates, the blue point is the centroid of a recommended rectangular bin. Finally, because our system

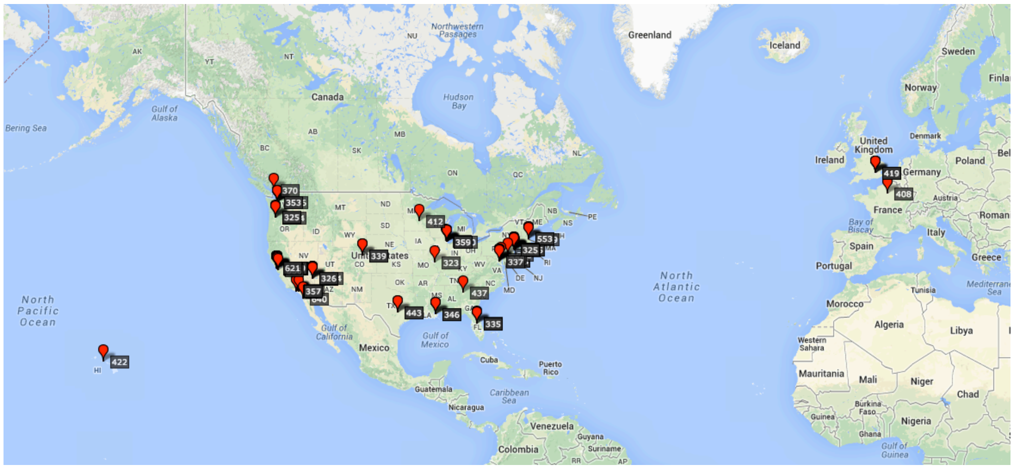


**Figure 3: Top-1000 photo locations ordered by number of unique users.**

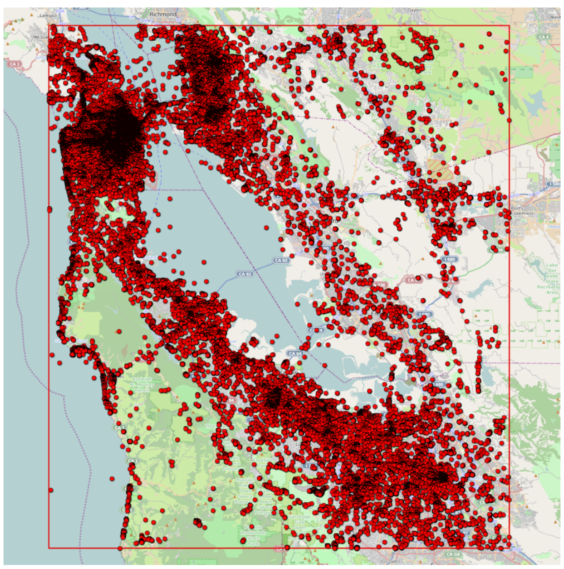

**Figure 2: 194K photos from the San Francisco Bay Area in California.**

uses collaborative filtering, personalized recommendations cannot be generated until the user has accumulated a sufficient number of his own geotagged photos (where we set a threshold of at least $l = 5$ locations, a value that can be changed). This "cold-start" problem is also shared by e-commerce systems that use collaborative filtering, and we follow best practices in this situation by offering generalized most-popular nearby geolocations (again within a distance $d$), where the most popular spots are found by summing unique users in specific bins.

## 3.2 Data Set

We produce these recommendations by applying collaborative filtering on a collection of photos across many users. To that end, we obtained a data set of publicly-available geotagged photos from the Flickr.com photo-sharing site using its public RESTful API. We searched for photos taken between September 1, 2009, and September 1, 2013, that were specifically taken with smartphones because such photos would typically have embedded geocoordinates produced by the smartphone when the photo was taken.

We started the photo search within the USA and found an initial set of users; we then expanded the search by obtaining all the photos taken by these users, even if they were taken in another country. The final set comprised 3,099,752 photos and 39,034 unique users across North America, Europe, and Asia. For example, Figure 2 shows the geolocations of 194,000 photos taken in the San Francisco Bay Area. Figure 3 shows the geographic distribution of the top-1000 photo locations sorted by the number of unique visitors.

For each photo, we collected: photo URL (string); user ID (string); epoch timestamp (long integer); latitude (float); and longitude (float). We made no assumption about the geocoordinates' accuracy other than that they were produced by the smartphone geolocation system (e.g. via GPS or Wi-Fi/cellular radio trilateration) at the time the photos were taken.

## 3.3 Geocoordinate Space Discretization

Because our goal is to make geolocation recommendations, we must represent real-world physical places in our system. Importantly, we wanted to remain in the geocoordinate space rather than align the data points to named POI physical locations such as names for landmarks or other buildings, where lists of available POIs have typically been assembled through a process of identifying significant locations within a city. As mentioned earlier, there are two advantages of staying in the geocoordinate space. First, because a physical location can be photographed from different spots, our system will be able to identify and recommend these spots around the location rather than be limited to that location's canonical address. Second, our system will be able to identify and recommend locations that are far away from any known named POI.

We thus need to uniquely identify locations, but the geocoordinate data is expressed as continuous floating-point values for the latitude and longitude. As a result, we discretized latitude and longitude pairs into virtual rectangular bins that are formed throughout the coordinate space. Each bin is cre-

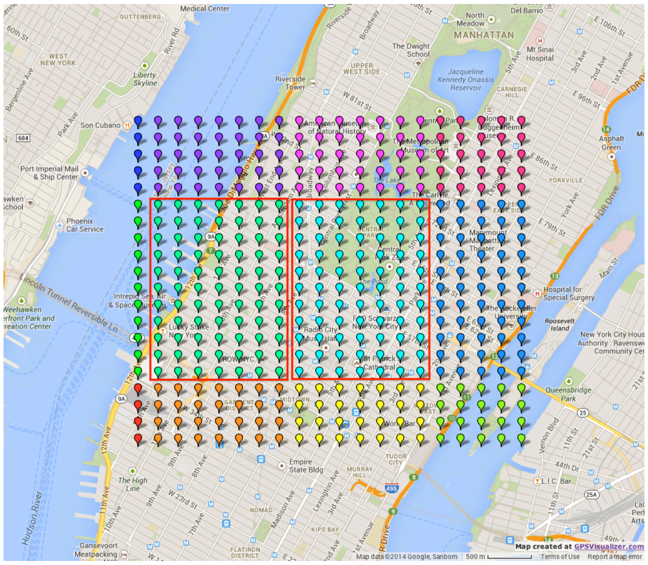


**Figure 4: The results of cartographic hashing, where data points within each differently-colored block correspond to a different virtual bin.**

ated through Cartographic Sparse Hashing, our $O(1)$ hash algorithm that takes (i) latitude and longitude and (ii) a resolution $r$ in meters (where in our experiments we varied $r$ from 100 meters to 2000 meters). The function then hashes the longitude and latitude into the high and low bits, respectively, of the resulting output 64-bit integer. This key then represents a virtual bin approximately $r$ meters per side, although the bin will be elongated north-to-south as one gets further away from the equator due to the Earth's curvature.

Figure 4 shows this hashing, where we plot many geocoordinates in New York City, hash each geocoordinate to a key using $r = 2000$m, and then assign a new color for each key. To aid visualizing distinctions, we outlined two unique bins in red. The end result of hashing is that any geocoordinates that fall within the same virtual bin will be assigned the same key. This key is the location identifier in the $user \times location$ co-occurrence matrix, described in the next subsection.

Our approach of using hashing has several advantages over clustering algorithms. We can precisely control the physical cartesian size of the virtual bin with the CASH algorithm's $r$ resolution parameter, which is important for identifying specific spots for photography. On the other hand, clustering algorithms like k-means, agglomerative clustering, or DBSCAN cannot be tuned to produce clusters of maximum cartesian size. Additionally, unlike k-means or k-medoid, our hashing technique does not need to specify the number of bins beforehand.

## 3.4 Recommendation Model

Recommender algorithms usually fall into one of two categories [15]: content-based and collaborative filtering. The former leverages product attributes, such as a movie's actors, to match against the same attributes for which a user has shown an interest. The latter generates recommendations using data from other users, where the recommendation stems from similarity among the users or the items. Our work uses collaborative filtering.

This approach's principal data structure is a co-occurrence matrix of $user \times item$, where an entry in the matrix at position $i, j$ is the rating given by user $i$ for item $j$. On some e-commerce sites like Netflix, users can explicitly rate an item on a scale of 1 to 5 stars. Note that since a single user is likely to evaluate only a small subset of available inventory, the overall matrix will tend to be extremely sparse.

For our data set, we likewise used a co-occurrence matrix, but here representing $user \times location$, where a column $location$ is a hashed virtual bin key. Sparsity is also a problem for our system: because users typically only visit a small subset of the possible locations, any given row will have only a few non-empty entries.

The ratings placed into the $user \times location$ matrix should represent the user's interest for a location, but unfortunately, in our data set users do not provide an explicit rating for the locations where they took their photos. We thus derived an implicit feedback [21] value normalized over $[0.0, 1.0]$ as the fraction of photos taken by a user at a location over the total count of photos taken by that user. More specifically, let $C_{ij}$ be the count of photos taken by user $i$ at location $j$, and let $L$ be the set of all locations. The implicit feedback of user $i$ for location $j$ is then

$$feedback_{ij} = \frac{C_{ij}}{\sum_{k \in L} C_{ik}}$$

We applied basic data filtering to reduce noise. We kept only locations that have feedback from at least $u$ users (in our experiments we set $u = 5$), and we kept users that have feedback for at least $l$ locations (in our experiments we set $l = 5$). In the future we will explore other settings.

## 3.5 Recommendation Algorithms

Given the $user \times location$ co-occurrence matrix parameterized with spatial bin sizes from the previous section, we looked to run our recommendation algorithms. For a given user, his row in the co-occurrence matrix is sparse with many missing rating values; the goal of the recommendation algorithm is then to impute the missing values based on observed values in the matrix [15, 20]. Once the matrix has been completed, a user can be given a recommendation list of previously-unvisited locations ordered by descending predicted ratings. In our work we applied two sets of algorithms, memory-based and model-based, both of which we implemented in Python.

For the memory-based approach [24], we impute the co-occurrence matrix using similarity between the locations. (Note that since we substitute $location$ for $item$, this algorithm is considered location-location similarity rather than item-item as is traditionally found in the literature [18]). Specifically, we first construct a similarity matrix consisting of all pairwise cosine similarity between the locations. For each user, we then impute all location ratings that we have not observed using the information from a neighborhood given by this similarity matrix. For each blank location, we find the top N similar locations where this user took photos before. The missing value is then estimated by the weighted average of these values.

We also applied three model-based methods: non-negative matrix factorization (NMF) [14, 27], probabilistic matrix factorization (PMF) [23], and a low- rank factorization model for matrix completion (LMaFit) [26]. In all these approaches, we learn two factors $U \in \mathbf{R}^{n\times k}$ and $V \in \mathbf{R}^{m\times k}$ from the partially observed co-occurrence matrix, where $n$ is the number of users, $m$ is the number of locations, and $k$ is the number of latent factors. The multiplication of these two matrices gives a dense approximation to the full matrix; i.e. we compute the multiplication $u_i v_j^T$ in order to obtain the prediction $\hat{X}_{ij}$. The three algorithms differ from each other in how these factors are obtained. The NMF obtains two non-negative factors by solving the following optimization objective:

$$\min_{U,V} \|X - UV^T\|_F^2, \quad \text{s.t.: } U \geq 0, V \geq 0$$

We use the efficient projected gradient [17] to solve this objective. PMF is motivated by introducing Gaussian priors on $U$ and $V$ and solves the following objective:

$$\min_{U,V} \sum_{ij} \|X_{ij} - U_i^T V_j\|_F^2 + \lambda_U \|U\|_F^2 + \lambda_V \|V\|_F^2$$

where $\lambda_U$ and $\lambda_V$ are regularization parameters which correspond to the variance of the Gaussian distribution. We use stochastic gradient descent to solve the objective. Finally, LMaFit solves a formulation that is similar to PMF:

$$\min_{U,V} \sum_{ij} \|X_{ij} - U_i^T V_j\|_F^2 + \lambda(\|U\|_F^2 + \|V\|_F^2)$$

where $\lambda$ is the regularization parameter that controls overfitting. The difference between LMaFit and PMF is mainly algorithmic, where in LMaFit we iteratively solve least squares problems [26], which is very efficient.

# 4. EXPERIMENTS AND RESULTS

We conducted a series of experiments on our collected data set of 3 million geotagged photos, where we evaluated the imputed user feedback for locations produced by the four recommendation algorithms that we described earlier (location-location, NMF, PMF, and LMaFit). We ran all experiments on a 64-bit CentOS Linux server with 64 GB of RAM. Each trial completed within a few hours, and in the future we will parallelize our algorithms to run on a compute cluster.

First, we looked to determine the number of unique discretized geographic bins that were generated from our hashing function. An input to this function is a value $r$ in meters that specifies the approximate length of one side of the bin. Intuitively, a smaller value of $r$ will increase the number of identified unique locations and reduce the number of geocoordinates that hash to the same bin. Figure 5 shows how many locations are found at varying hashing resolutions, where the highest count occurs at the smallest resolution of 100m, matching our understanding.

We then evaluated our algorithms using root-mean-squared-error (RMSE), a standard metric for recommender systems [10, 12, 11]. Recall that the goal of a recommendation algorithm is to predict missing rating values from observed values in the matrix. We conducted our evaluation on a test set by computing RMSE between the predicted values and the observed ones. More formally, we define $S$ to be the set of all *user* $\times$ *location* co-occurrence pairs, $r_{ij}$ to be the ground-truth feedback given by user $i$ for location $j$ (following our implicit feedback assumption previously mentioned), and $\hat{r}_{ij}$ to be the estimate for this feedback provided by the

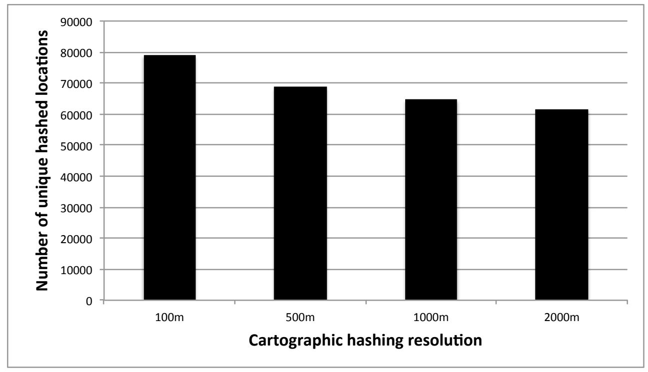


**Figure 5: The number of unique locations at varying hash resolutions.**

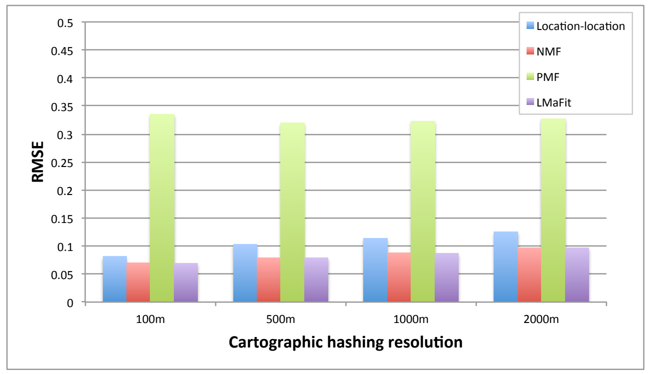


**Figure 6: RMSE results for all algorithms.**

algorithms. RMSE is then

$$RMSE = \sqrt{\frac{1}{|S|} \sum_{i,j \in S} (\hat{r}_{ij} - r_{ij})^2}$$

We tuned our algorithms beforehand using RMSE in order to get the best results. The parameters that produced the best average performance on hold-out validation data are used to build a model from training data, and the model is evaluated on test data. For example, for LMaFit, we tuned $\lambda$ by parameter-sweeping experiments with a bin hashing resolution 500m, resulting in the best RMSE at $\lambda = 1000$.

Figure 6 shows the final RMSE results for all algorithms at a hashing resolution of 500m. NMF and LMaFit produced the lowest RMSE values, while PMF produced the highest values.

We further note the overall trend across all algorithms where larger values for the hash resolution produce higher RMSE. We continue to investigate this phenomenon, but we conjecture that RMSE increases because a larger bin captures more geocoordinates, which in turn increases the likelihood that user-location combinations will share similar feedback (i.e. fraction of photos taken at a location as defined in Section 3.4) when in reality their similarity is low. A smaller bin, on the other hand, is finer-grained, so similar user-location combinations are less likely to be spurious.

## 5. CONCLUSION

We explored the viability of generating recommendations for photo-taking locations in the geocoordinate space using a data set of 3 million geotagged photos from the Flickr.com website. We produced these recommendations by first discretizing the continuous latitude and longitude geocoordinates into geographic virtual bins to serve as unique location identifiers; this step allows us to remain in the geocoordinate space rather than align our data points to a POI landmarks, a choice which is more helpful for taking photos. We then used four collaborative filtering algorithms (one memory-based and three model-based) and evaluated the approaches using RMSE. Our early results show that RMSE is sensitive to increasing hash bin size. In the future we will evaluate the system qualitatively with a user study, more completely determine the algorithms' impact on RMSE, and implement an end-to-end mobile application.

## 6. REFERENCES


[1] R. Bell, Y. Koren, and C. Volinsky. "All Together Now: a Perspective on the Netflix Prize," *Chance*, vol. 23, no. 1, 2010.

[2] S. Bourke, K. McCarthy, and B. Smyth. "The Social Camera: Recommending Photo Composition Using Contextual Features," In *Proceedings of CARS*, 2010.

[3] A.J.-Cheng, Y.-Y. Chen, Y.-T. Huang, W. Hsu, and H.-Y. Liao. "Personalized Travel Recommendations by Mining People Attributes from Community-Contributed Photos," In *Proceedings of ACM MM*, 2011.

[4] M. Clements, P. Serdyukov, A. de Vries, and M. Reinders. "Using Flickr Geotags to Predict User Travel Behaviour," In *Proceedings of ACM SIGIR*, 2010.

[5] D. Crandall, L. Backstrom, D. Huttenlocher, and J. Kleinberg. "Mapping the World's Photos," In *Proceedings of WWW*, 2009.

[6] M. De Choudhury, M. Feldman, S. Amer-Yahia, N. Golbandi, R. Lempel, and C. Yu. "Automatic Construction of Travel Itineraries using Social Breadcrumbs," In *Proceedings of ACM HT*, 2010.

[7] T. Donegan. "Smartphone cameras are taking over," *USA Today*, June 6, 2013. `www.usatoday.com/story/tech/2013/06/06/reviewed-smartphones-replace-point-and-shoots/2373375/`

[8] F. Girardin, J. Blat, F. Calabrese, F. Dal Fiore, and C. Ratti. "Digital Footprinting: Uncovering Tourists with User-Generated Content," *IEEE Pervasive Computing*, October-December 2008.

[9] D. Goldberg, D. Nichols, B. Oki, and D. Terry. "Using collaborative filtering to weave an information tapestry," *Communications of the ACM*, 35(12), 1992.

[10] A. Gunawardana and G. Shani. "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research*, vol. 10, 2009.

[11] M. Hahsler. "recommenderlab: Lab for Developing and Testing Recommender Algorithms." `cran.r-project.org/web/packages/recommenderlab/index.html`

[12] J. Herlocker, J. Konstan, L. Terveen, and J. Riedl. "Evaluating Collaborative Filtering Recommender Systems," *ACM Transactions on Information Systems*, 22(1), 2004.

[13] S. Horaczek. "The Most Photographed (and Uploaded) Places in the World According to Google," *PopPhoto.com*, December 27, 2013. `www.popphoto.com/news/2013/12/most-photographed-and-uploaded-places-world-according-to-google`

[14] D. Lee and H. Seung. "Learning the parts of objects by non-negative matrix factorization," *Nature*, 401(6755), 1999.

[15] J. Leskovec, A. Rajaraman, and J. Ullman *Mining of Massive Datasets*, Cambridge University Press, 2012.

[16] J. Levandoski, M. Sarwat, A. Eldawy, and M. Mokbel. "LARS: A Location-Aware Recommender System," In *Proceedings of ICDE*, 2012.

[17] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.

[18] G. Linden, B. Smith, and J. York. "Amazon.com Recommendations, Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, Jan.-Feb. 2003.

[19] B. Liu, Y. Fu, Z. Yao, and H. Xiong. "Learning Geographical Preferences for Point-of-Interest Recommendation," In *Proceedings of KDD*, 2013.

[20] P. Melville and V. Sindwani. "Recommender Systems," book chapter of *Encyclopedia of Machine Learning*, Springer-Verlag, 2010.

[21] D. Oard and J. Kim. "Implicit Feedback for Recommender Systems," In *Proceedings of DELOS Workshop on Filtering and Collaborative Filtering*, 1998.

[22] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Rield. "GroupLens: an open architecture for collaborative filtering of netnews," In *Proceedings of ACM CSCW*, 1994.

[23] R. Salakhutdinov and A. Mnih. "Probabilistic matrix factorization," In *NIPS*, volume 1, 2007.

[24] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. "Item-based collaborative filtering recommendation algorithms," In *Proceedings of ACM WWW*, 2001.

[25] Y. Shi, P. Serdyukov, A. Hanjalic, and M. Larson. "Nontrivial Landmark Recommendation Using Geotagged Photos," *ACM Transactions on Intelligent Systems and Technology*, 4(3), June 2013.

[26] Z. Wen, W. Yin, and Y. Zhang. "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, 4(4), 2012.

[27] S. Zhang, W. Wang, J. Ford, and F. Makedon. "Learning from incomplete ratings using non-negative matrix factorization" In *SDM*. SIAM, 2006.

[28] Y. Zheng, L. Zhang, X. Xie, and W.-Y. Ma. "Mining Interesting Locations and Travel Sequences from GPS Trajectories," In *Proceedings of WWW*, 2009.

[29] V. Zheng, Y. Zheng, X. Xie, and Q. Yang. "Collaborative Location and Activity Recommendations with GPS History Data," In *Proceedings of WWW*, 2010.