Matrix Factorization for Near Real-time Geolocation Prediction in Twitter Stream

Nghia Duong-Trung, Nicolas Schilling, Lucas Rego Drumond, and Lars Schmidt-Thieme

Information Systems and Machine Learning Lab (ISMLL)
Universitätsplatz 1, 31141 Hildesheim, Germany
{duongn,schilling,ldrumond,schmidt-thieme}@ismll.uni-hildesheim.de
http://www.ismll.uni-hildesheim.de

Abstract. The geographical location is vital to geospatial applications such as event detection, geo-aware recommendation and local search. Previous research on this topic has investigated geolocation prediction framework via conducting pre-partitioning and applying classification methods. These existing approaches target user's geolocation all at once via concatenation of tweets. In this paper, we study a novel problem in geolocation. We aim to predict user's geolocation at a given tweet's posting time. We propose a geo matrix factorization model to address this problem. First, we map tweets into a latent space using a matrix factorization technique. Second, we use a linear combination in the latent space to predict exact latitude and longitude. However, we only use one individual tweet as the input instead of using a concatenation of all tweets of a user. Our experimental results show that the proposed model has outperformed a set of regression models and state-of-the-art classification approaches.

Keywords: Twitter, Near real-time Geolocation, Matrix Factorization

1 Introduction

In the past years, online social networking and social media sites, e.g. Twitter in general, have become an ubiquitous and constant mechanism for sharing and seeking information. Although a tweet's length is limited to 140 characters, there is still a huge amount of information to explore. Its contents are inherently multifaceted and dynamic; consequently, representing people's thoughts and public announcement at a temporal currency and vicinity. This causes Twitter data to become specifically interesting for multi-purpose investigations as they are tweeted in near real-time fashion. Understanding the near real-time user's geographical location, e.g. latitude and longitude pairs or physical coordinates, enables providing policies and intervention aid strategies in a particular region such as localized aid [31,9], disaster response [27,21], event detection [28] and disease surveillance [3].

One of the early pioneer papers about geolocation in Twitter streams was published in 2010 [12]. In that work, the authors concatenated all user's tweets

during a specified duration into one single representative document. The geolocation of the first tweet or the first available geo-tagged tweet in the collection was then the geolocation of the representative document. Using a concatenation provides circumstantial contents to develop a wide variety of techniques used in geo-locating such as content analysis with terms in a gazetteer [19], content analysis with probabilistic language models [11,16,1], metadata of various sorts such as follow-following relationships [17,22], behavior-based time zone [20]. Furthermore, the research conducted in [17] exploits the idea of geolocation prediction as label propagation by interpreting location labels spatially. Additionally, the work of [6] extends [17] by taking into account edge weights as a function reflecting user interactions.

Prerequisites to these directions are the representation of the earth's surface. Geolocations can be captured as points, or clusters based on a pre-partitioning of regions into discrete sub-regions using city locations [5,18,26], named entities and location indicative words [14] as well as vernacular expressions with the aid of comprehensive gazetteers [15]. Another approach of partitioning the earth's surface is to use a grid. While the simplest grid is a uniform rectangular one with cells of equal-sized degrees [30], more advanced grids are either an adaptive grid based on k-d trees [25], an equal-area quaternary triangular mesh [8] or a hierarchical structure [29].

However, these approaches have some drawbacks due to some reasons. First of all, as being classification methods, they heavily depend on pre-partitioning or a framing architecture that is used to split the regions into discrete sub-regions. Thus, they discard the natural properties of real physical coordinates. Moreover, concatenating tweets into one representative document requires a time-consuming collection as well as data abundance. In addition, concatenation of tweets during a particular duration, e.g. a month, leads to failure of capturing geolocation in near real-time situations. Effective geolocation of a user while posting a single short tweet based purely on its content is a direction worth-investigating and also constitutes a more difficult task.

In this paper, we address a novel geolocation prediction scenario via regression within indicative latent feature space. By working on the latent feature space, we have proved that regression models can be utilized to solve this prediction problem. We aim to predict the exact user geolocation at a given posting's time, simply based on the textual content of tweets, ignoring their metadata.

2 Proposed Method

In this section, we present the general notation used in this paper as well as our approach. It is based on a matrix factorization of the individual tweets where we then learn a latent representation of tweets and words. This latent representation will then be used to predict the final geolocation. We also present a learning algorithm for our approach which is optimized by stochastic gradient descent.

2.1 Notation

Consider a dataset D containing a set of tweets where each tweet is described by n many features. The dataset will be split into a training D^{train}, a test D^{test} and a validation D^{valid} set, which will be used for hyperparameter optimization later. We have m, l and v tweets in the training D^{train}, test D^{test} and validation D^{valid} sets respectively. The tweet features are mapped from a dictionary that comprises all words/tokens/unigrams in the dataset. We denote the vocabulary size by $|V| = n$.

Each tweet is annotated with a ground-truth coordinate pair $\mathbf{y} \in \mathbb{R}^2$, $\mathbf{y} = (y^{lat}, y^{lon})$ where $y^{lat} \in \mathbb{R}$ is the latitude and $y^{lon} \in \mathbb{R}$ is the longitude of the associated tweet. By $\bar{\mathbf{y}}_{u_i} = (\bar{y}^{lat}_{u_i}, \bar{y}^{lon}_{u_i})$ we denote the average geolocation of a user in the training set, where $\bar{y}^{lat}_{u_i} \in \mathbb{R}$ is the average latitude and $\bar{y}^{lon}_{u_i} \in \mathbb{R}$ is the average longitude. Using $\bar{\mathbf{y}}_U = (\bar{y}^{lat}_U, \bar{y}^{lon}_U)$, we denote the average geolocation of all users in the training set. Given some training data $X^{train} \in \mathbb{R}^{m \times n}$, and the respective labels $Y^{train} \in \mathbb{R}^{m \times 2}$, we seek to learn a machine learning model $f : \mathbb{R}^n \to \mathbb{R}^2$ which maps tweets to geolocations such that for some test data $X^{test} \in \mathbb{R}^{v \times n}$, the sum of distances

$$\sum_{i=1}^{v} d(f(X_i^{test}), Y_i^{test}) \tag{1}$$

is minimal. By $Y^{test} \in \mathbb{R}^{v \times 2}$ we denote the set of ground-truth labels for the test data. Note that, d is a distance metric where in our learning algorithm we use the Haversine distance.

2.2 The Geo Matrix Factorization Model

Over the last decade, Matrix Factorization (MF) models have gained much attention by the Netflix Prize competition where they have shown very good predictive performance as well as decent run-time complexity in terms of dealing with very sparse matrices. Based on the vanilla MF, we develop a more multi-relational-oriented factorization model for the geolocation regression task: the Geo Matrix Factorization (GMF) model. We approach the user geolocation problem as a text regression task where we aim to predict the exact latitude and longitude values using an individual tweet. However, instead of using the highly sparse word counts as features in a linear regression, we firstly factorize the input space by learning a matrix $T \in \mathbb{R}^{m \times k}$ for tweets and $W \in \mathbb{R}^{k \times n}$ for individual words of each tweet to reconstruct X as:

$$X \approx TW \tag{2}$$

As in the usual setting, the number of latent features k is usually much smaller than the number of words n, such that through this approach, tweets are projected into a lower dimensional latent feature space. This latent representation of a tweet is then used within a linear model to predict the geolocation of the user at the posting time of the tweet:

$$\hat{y}_l^{lat} = \bar{y}_{u_l}^{lat} + \phi_0 + \sum_{k=1}^{K} \phi_k T_{lk}^{lat}$$
$$\hat{y}_l^{lon} = \bar{y}_{u_l}^{lon} + \theta_0 + \sum_{k=1}^{K} \theta_k T_{lk}^{lon} \tag{3}$$

where $\phi \in \mathbb{R}^{k+1}$ and $\theta \in \mathbb{R}^{k+1}$ are weight coefficients vectors for learning latitude and longitude respectively. Notice that we also actually perform two factorizations of X, one for latitude which yields T^{lat}, this is done for longitude as well. Our model then actually predicts the average training location of a user, plus a regression term on the latent feature space obtained by the factorization of X.

2.3 Model Fitting

Given the model, we have to learn parameters $T^{lat}, T^{lon}, W^{lat}, W^{lon}, \theta, \phi$, where the W matrices are only used for reconstructing X and not for predicting the actual geolocation. We optimize the prediction of the gelocation as well as the factorization of X for the least-squares error. In order to prevent the model from overfitting to the training data we apply a Tikhonov regularization on the regression parameters θ and ϕ, the latent feature matrices are regularized using the Frobenius norm. The overall loss term for learning the parameters associated to predicting latitude then looks like

$$\mathcal{L}^{lat}(\hat{y}^{lat}, y^{lat}) = \frac{1}{\mid X^{train} \mid} \left\| \hat{y}^{lat} - y^{lat} \right\|^2 + \lambda_\phi \left\| \phi \right\|^2$$
$$+ \left\| X^{train} - T^{lat} W^{lat} \right\|_F^2 + \lambda_T \left\| T^{lat} \right\|_F^2 + \lambda_W \left\| W^{lat} \right\|_F^2 \quad , \tag{4}$$

where the loss term associated to longitude $\mathcal{L}^{lon}(\hat{y}^{lon}, y^{lon})$ is similar. The only difference is that it involves θ, T^{lon} and W^{lon}. In Equation 4, the term $\left\| X^{train} - T^{lat} W^{lat} \right\|_F^2$ is the residual error of transforming X into T^{lat}, W^{lat}. The regularization terms $\lambda_\phi \left\| \phi \right\|^2$, $\lambda_T \left\| T^{lat} \right\|_F^2$, and $\lambda_W \left\| W^{lat} \right\|_F^2$ are multiplied by regularization parameters λ_ϕ, λ_T, and λ_W that control the amount of regularization.

These terms penalize parameters with high magnitudes, that typically lead to overly complex models with very small training errors but bad generalization performance. Certainly, these hyperparameters can not be learned from the data and will be optimized using a grid-search on the validation partition of the data. To solve the above optimization tasks, we apply Stochastic Gradient Descent (SGD) [2,13] where the learning rate is estimated using the Adaptive Subgradient Method (AdaGRAD) [10] which helps yielding a better run-time performance. The basic idea of SGD is that, instead of expensively calculating the gradient

of Equation 4 and its latitude counterpart, it randomly selects a tweet and calculates the corresponding gradient. Suppose we have chosen a tweet indexed by m, the partial derivatives of Equation 4 with the respect to T^{lat} can be computed as:

$$\begin{aligned}\frac{\partial \mathcal{L}^{lat}(\hat{y}_m^{lat}, y_m^{lat})}{\partial T_{ml}^{lat}} = &- \left(y_m^{lat} - \bar{y}_{u_m}^{lat} - \sum_{k=1}^{K} \phi_k T_{mk}^{lat} - \phi_0\right)\phi_l \\ &- \sum_{n=1}^{N} \left(\left(X_{mn} - \sum_{k=1}^{K} T_{mk}^{lat} W_{kn}^{lat}\right) W_{ln}^{lat}\right) + \lambda_T T_{ml}^{lat}\end{aligned} \tag{5}$$

The partial derivatives with respect to the latent feature matrix W^{lat} of the tokens is obtained by

$$\frac{\partial \mathcal{L}^{lat}(\hat{y}_m^{lat}, y_m^{lat})}{\partial W_{lj}^{lat}} = -\left(X_{mj} - \sum_{k=1}^{K} T_{mk}^{lat} W_{kj}^{lat}\right) T_{ml}^{lat} + \lambda_W W_{lj}^{lat} \tag{6}$$

Finally, the partial derivative of the regression parameters has the form:

$$\begin{aligned}\frac{\partial \mathcal{L}^{lat}(\hat{y}_m^{lat}, y_m^{lat})}{\partial \phi_j} &= - \left(y_m^{lat} - \bar{y}_{u_m}^{lat} - \sum_{k=1}^{K} \phi_k T_{mk}^{lat} - \phi_0\right) T_{mj}^{lat} + \lambda_\phi \phi_j \\ \frac{\partial \mathcal{L}^{lat}(\hat{y}_m^{lat}, y_m^{lat})}{\partial \phi_0} &= - \left(y_m^{lat} - \bar{y}_{u_m}^{lat} - \sum_{k=1}^{K} \phi_k T_{mk}^{lat} - \phi_0\right)\end{aligned} \tag{7}$$

The partial derivatives of the longitude loss with the respect to T^{lon}, W^{lon} and θ can be calculated in the exact same manner as Equations 5, 6 and 7.

2.4 Inference for Test Data

By optimizing the respective loss terms for the training data, we learn the latent representation T of all training tweets as well as the linear regression parameters θ and ϕ for predicting the final geolocation. However, as we want to predict geolocations of unseen test tweets, the latent representations T for the individual training tweets cannot be employed. Out of this reason, we perform a fold-in, where we factorize the feature matrix X^{test} of the test data, using the latent representation W of the word tokens that was learned on the training data. To avoid confusion, we denote the latent tweet representations for the test tweets by T'^{lat} and T'^{lon} and factorize X^{test} as

$$X^{test} \approx T'^{lat} W^{lat} \tag{8}$$

as well as the respective term for longitude.

As we can see, W^{lat} and W^{lon} are reused from the learning phase. Subsequently, in the fold-in phase, we define the objective function that we need to minimize for T'^{lat} as follows:

$$\mathcal{L}^{lat}\left(X^{test}, T'^{lat}W^{lat}\right) = \frac{1}{|X^{test}|}\left\|X^{test} - T'^{lat}W^{lat}\right\|_F^2 + \lambda_{test}\left\|T'^{lat}\right\|_F^2\Bigg) \tag{9}$$

The partial derivatives of Equation 9 with the respect to T'^{lat} can be computed by:

$$\frac{\partial\mathcal{L}^{lat}\left(X_{jn}^{test}, T'_{jk}{}^{lat}W_{kn}^{lat}\right)}{\partial T'^{lat}_{jk}} = -\left(X_{jn}^{test} - \sum_{k=1}^{K} T'_{jk}{}^{lat}W_{kn}^{lat}\right)W_{kn}^{lat} + \lambda_{test}T'^{lat}_{jk} \tag{10}$$

The partial derivatives with the respect to T'^{lon} can be also computed in the same manner as for Equation 10. Having learned the latent representation of the test tweets using the fold-in procedure, we can then perform predictions for the test users using Equation 3. However, not all users that appear in the test data necessarily have to appear in the training data, hence we cannot use their average geolocation for the final prediction. For those users, we then use the median geolocation of all users of the training data as:

$$\bar{\mathbf{y}}_{u_l} = \begin{cases} \bar{\mathbf{y}}_{u_l}, & \text{if } u_l \in D^{train} \\ \bar{\mathbf{y}}_U, & \text{otherwise} \end{cases} \tag{11}$$

Algorithm 1 illustrates how the overall GMF works.

3 Experiments

In this section, we first describe the datasets that we use as well as their preprocessing. Additionally, we describe how we optimized the hyperparameters of our model. Finally, we compare our approach to a set of competing methods.

3.1 Dataset

We have worked with three publicly available tweet datasets containing geolocation information and compiled them to fit the user geolocation prediction within the near real-time scenario. One dataset comprises the tweets posted within the United States, whereas the other dataset contains all tweets localized to north America and the world. Through this, we evaluate our model's effectiveness and generality within different geographical scopes from a country to the whole world. A splitting protocol is then designed for these datasets. We randomly

Algorithm 1 GMF

Require: $X^{train} \in \mathbb{R}^{m \times n}$, $X^{test} \in \mathbb{R}^{l \times n}$, $Y \in \mathbb{R}^{m \times 2}$
Ensure: $T \in \mathbb{R}^{m \times k}$, $T' \in \mathbb{R}^{l \times k}$, $W \in \mathbb{R}^{k \times n}$, $\phi \in \mathbb{R}^{k+1}$, $\theta \in \mathbb{R}^{k+1}$

Initialize $T^{lat} \leftarrow \mathcal{N}(0,1)$, $T^{lon} \leftarrow \mathcal{N}(0,1)$, $W^{lat} \leftarrow \mathcal{N}(0,1)$, $W^{lon} \leftarrow \mathcal{N}(0,1)$, $\phi \leftarrow \mathcal{N}(0,1)$, $\theta \leftarrow \mathcal{N}(0,1)$, $T'^{lat} \leftarrow \mathcal{N}(0,1)$, $T'^{lon} \leftarrow \mathcal{N}(0,1)$
// Learning phase
for $epoch \in 1, \ldots, max\_epoch$ **do**
  for $iteration \in 1, \ldots, M$ **do**
    Pick m randomly
    Pick X^{train}_{mn} randomly
    for $k \in 1, \ldots, K$ **do**
      Learning T^{lat}_{mk}, T^{lon}_{mk}, W^{lat}_{kn}, W^{lon}_{kn}, $\phi_{T\,mk}$, $\theta_{T\,mk}$
    end for
    Update ϕ_0, θ_0
  end for
end for
// Fold-in phase
for $epoch \in 1, \ldots, max\_epoch'$ **do**
  for $iteration \in 1, \ldots, L$ **do**
    Pick l randomly
    if X^{test}_{ln} exists **then**
      for $k \in 1, \ldots, K$ **do**
        Learning T'^{lat}_{lk} , T'^{lon}_{lk}
      end for
    end if
  end for
end for
// Prediction
for $l \in 1, \ldots, L$ **do**
  $\hat{y}^{lat}_l \leftarrow \bar{y}^{lat}_{u_l} + \phi_0 + \phi_{lk} T'^{lat}_{lk}$
  $\hat{y}^{lon}_l \leftarrow \bar{y}^{lon}_{u_l} + \theta_0 + \theta_{lk} T'^{lon}_{lk}$
end for
return $d_H(\mathbf{y}, \hat{\mathbf{y}})$

split *all tweets of each user* by a 60/20/20 scheme, denoted as LocalRandom (LR). Secondly, we also investigate how our model works with a user appearing in the test set might not exist in the training data by splitting *all tweets* using the 60/20/20 scheme, called GlobalRandom (GR).

US. This dataset is originally implemented by [12], and was later also used in [11,30,16]. The dataset comprises tweets gathered from the "Gardenhose" sample stream in the first week of March, 2010. In this dataset, the authors already provide geotagged tweets that we simply reuse. The implementing dataset contains 377,616 tweets posted by 9,475 users.

NA. The second dataset was collected by [25] and later implemented by [29,15]. This dataset contains tweets within north America, including the United

States, parts of Canada and Mexico from September 4th to November 29th, 2011. Because Twitter does not allow the distribution of complete tweets at that time, the NA dataset only contains user IDs and tweet IDs. Subsequently, we have to fetch the tweets from Twitter using its official API to check whether the tweets are available as well as their availability of embedded coordinates. Only 226,595 tweets out of 38 million posted by 10,950 users have geotags available and therefore are considered for the final dataset.

WORLD. The last dataset was compiled by [14] and later implemented by [29,15]. The dataset comprises tweets from all over the world. As being described in NA dataset, we also apply the same retrieving procedure. The implementing dataset then contains 121,327 tweets posted by 80,179 users. In the WORLD dataset, 70% of users has only one tweet. So that we only apply the GR 60/20/20 splitting scheme to it.

3.2 Data Preprocessing

In addition to length restriction, tweets are also characterized by the use of terms that are not found in natural language, including hashtags, abbreviations, emoticons and URLs. Through this, we propose a data preprocessing procedure as follows.

Tokenization. We apply a uni-gram tokenization procedure that preserves hashtags, @-replies, abbreviations, blocks of punctuation, emoticons and unicode glyphs and other symbols as tokens. We remove URL tokens to prevent the tweets where bots are posting information such as advertisement to enter our dataset.

Bag-of-words representation. After all tweets are tokenized, they are converted from sparse vectors of token counts into sparse vectors of bag-of-words representations using term frequency - inverse document frequency (TF.IDF) scores. By using the TF.IDF scores, we discard language and grammar structure, the token's order, semantics and meaning as well as part-of-speech. The TF.IDF weights reflect how important a token is to an instance. The more common a token is to many instances, the more penalization it gets. The tokens with the highest TF.IDF weight are often the tokens that best characterize the instance.

3.3 Evaluation Metrics

Given the ellipsoidal shape of the earth's surface, we apply the Haversine distance to calculate the distance of two points represented by their latitude in range of $\{-90, 90\}$ and longitude in range of $\{-180, 180\}$. The Haversine distance $d_H : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ is the great circle distance between two geographical coordinate pairs. We compute the distance between two points by the Haversine formula [24]. The formula of the central angle α between them is given by:

$$\alpha = \left(\sin^2 \left(\frac{|\hat{y}^{lat} - y^{lat}|}{2} \right) + \cos(y^{lat}) \cos(\hat{y}^{lat}) \sin^2 \left(\frac{|\hat{y}^{lon} - y^{lon}|}{2} \right) \right)^{\frac{1}{2}} \quad (12)$$

Then, the Haversine distance of the two points can be calculated by:

$$d_H(\mathbf{y}, \hat{\mathbf{y}}) = 2r \arcsin(\alpha) \tag{13}$$

where r is the radius of the earth. Because of the ellipsoidal shape of the earth, its radius varies from the equator to the poles. According to [7], we take the mean of the earth's radius which amounts to $r = 6371$ km. Finally, the evaluation metrics are the mean and median Haversine distances d_H in kilometers between the ground-truth geolocation $\mathbf{y}$ and the predicted geolocation $\hat{\mathbf{y}}$.

3.4 Hyperparameter Setup

In order to obtain good predictive performance, we also need to carefully tune the hyperparameters in our model. By $k \in \mathbb{N}^+$ we denote the number of latent features used within the factorization of X. By $\lambda_T, \lambda_W, \lambda_\phi, \lambda_\theta$ and $\lambda_{T'}$ we denote the regularization hyperparameters used when learning the latent feature matrices, latent vocabulary matrices, the linear regression parameters for predicting latitude and longitude and the latent features matrices for the test tweets respectively. With $\alpha_T, \alpha_W, \alpha_\phi, \alpha_\theta$ and $\alpha_{T'}$ we denote the respective learning rates. We tune the hyperparameters by assessing the validation performance of our model and choosing the hyperparameter configuration which performs best. The number of latent dimensions is selected among the range of $k \in \{2, 4, 8, 16\}$, while the value of all other hyperparameters are selected among the range of $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. The preprocessed datasets used in the paper are publicly available unconditionally[1].

3.5 Results and Comparison

For the Support Vector Machine (SVM) and Factorization Machines (FM), we run them separately to predict latitude and longitude. To allow for a fair comparison, all these regression models also include the user bias in their estimation. Finally, we combine the predicted latitude and longitude to conduct a final distance calculation. For these models, we also apply a grid-search mechanism to find the best hyperparameter configurations for each prediction of latitude and longitude. On each dataset, we repeat running the models 10 times and take the average results. The final results can be observed in Table 1. We can see that all other regression models on average do not perform that well, mainly due to them using the extremely sparse $5,200$ TF.IDF features. Our model, however, maps each tweet individually into an eight-dimensional latent feature space and uses those features for prediction. The number of k latent feature is found by grid-search mechanism. The results show that GMF outperforms all competitors with large margins.

We also report the state-of-the-art results by classification approaches (see Table 2). One might notice that there are significant differences in term of accuracy prediction in two geolocation prediction scenarios. By targeting user's geolocation at a given posting's time, the results show that our model significantly

[1] Available online at: http://fs.ismll.de/publicspace/GMF/

Table 1. The results by regression approaches targeting the user's geolocation in a given posting's time scenario using only textual information. The mean and median Haversine distance error are in *km*. The best distances are in **bold**.

Corpus	LR_US		LR_NA		GR_US		GR_NA		WORLD	
Model	mean	median	mean	median	mean	median	mean	median	mean	median
SVM (RBF kernel) [4]	34.63	7.81	157.81	8.42	32.29	8.22	171.72	10.23	3179.57	2654.17
FM [23]	29.67	0.68	164.51	7.27	27.09	0.66	177.53	7.26	3219.16	2650.48
Our model	**29.15**	**0.66**	**157.22**	**6.95**	**26.44**	**0.65**	**170.08**	**7.19**	**2524.66**	**553.24**

reduces the localization error on the US and NA datasets. For the WORLD dataset, the average individual tweet's length is 5 tokens while being 49 tokens for the concatenation of tweets, our model still achieves reasonable results.

Table 2. The state-of-the-art results by classification approaches targeting the user's geolocation in all-at-once scenario using only textual information. The mean and median Haversine distance error are in *km*. ("-" signifies no implemented results for the given dataset, and "?" signifies that no result was reported for the given metric).

Corpus	US		NA		WORLD	
Model	mean	median	mean	median	mean	median
Hierarchical clustering [29]	-	-	686.6	171.5	1669.6	490.0
Hierarchical topic model [1]	?	298	-	-	-	-

4 Conclusion and Future Work

We have investigated the geo matrix factorization model for the task of near real-time text-based geolocation in Twitter. In our work, we tackle the user geolocation prediction task in a regression perspective. We analyze a single tweet as the model's input without any concatenation. Through this, we can further predict the user trajectory and achieve geolocation at a given posting's time. This is a starting point for further investigation on the affection of tweet concatenation or the number of tweets needed to achieve an acceptable distance error. Furthermore, We also address the sparsity and imbalance of online conversational texts by a matrix factorization technique. Based on the experiment results, our model outperforms all the competitors including SVM and FM within the regression task using dedicated latent feature spaces. In comparison with current state-of-the-art results by classification approaches, our model still outperforms and/or achieve reasonable results. Our further improvement broadly falls into various directions: optimization or applying the model over different datasets. In the optimization direction, we will analyze direct optimization of the Haversine formula. We also expand our model to predict near real-time geolocation of another types of datasets such as Wikipedia articles and Flickr images.

Acknowledgments. Nghia Duong-Trung gratefully acknowledges the funding of his work by the Ministry of Education and Training of Vietnam under the national project no. 911.

References

1. Ahmed, A., Hong, L., Smola, A.J.: Hierarchical geographical modeling of user locations from social media posts. In: Proceedings of the 22nd international conference on World Wide Web. pp. 25–36. International World Wide Web Conferences Steering Committee (2013)
2. Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, pp. 177–186. Springer (2010)
3. Burton, S.H., Tanner, K.W., Giraud-Carrier, C.G., West, J.H., Barnes, M.D.: " right time, right place" health communication on twitter: value and accuracy of location information. Journal of medical Internet research 14(6) (2012)
4. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011)
5. Cheng, Z., Caverlee, J., Lee, K.: You are where you tweet: a content-based approach to geo-locating twitter users. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 759–768. ACM (2010)
6. Compton, R., Jurgens, D., Allen, D.: Geotagging one hundred million twitter accounts with total variation minimization. In: Big Data (Big Data), 2014 IEEE International Conference on. pp. 393–401. IEEE (2014)
7. Decker, B.L.: World geodetic system 1984. Tech. rep., DTIC Document (1986)
8. Dias, D., Anastácio, I., Martins, B.: A language modeling approach for georeferencing textual documents. In: Actas del Congreso Español de Recuperación de Información (2012)
9. Dredze, M.: How social media will change public health. Intelligent Systems, IEEE 27(4), 81–84 (2012)
10. Duchi, J., Hazan, E., Singer, Y.: Adaptive subgradient methods for online learning and stochastic optimization. The Journal of Machine Learning Research 12, 2121–2159 (2011)
11. Eisenstein, J., Ahmed, A., Xing, E.P.: Sparse additive generative models of text. In: Proceedings of the 28th International Conference on Machine Learning (ICML-11). pp. 1041–1048 (2011)
12. Eisenstein, J., O'Connor, B., Smith, N.A., Xing, E.P.: A latent variable model for geographic lexical variation. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. pp. 1277–1287. Association for Computational Linguistics (2010)
13. Gemulla, R., Nijkamp, E., Haas, P.J., Sismanis, Y.: Large-scale matrix factorization with distributed stochastic gradient descent. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 69–77. ACM (2011)
14. Han, B., Cook, P., Baldwin, T.: Geolocation prediction in social media data by finding location indicative words. Proceedings of COLING 2012: Technical Papers pp. 1045–1062 (2012)
15. Han, B., Cook, P., Baldwin, T.: Text-based twitter user geolocation prediction. Journal of Artificial Intelligence Research pp. 451–500 (2014)

16. Hong, L., Ahmed, A., Gurumurthy, S., Smola, A.J., Tsioutsiouliklis, K.: Discovering geographical topics in the twitter stream. In: Proceedings of the 21st international conference on World Wide Web. pp. 769–778. ACM (2012)
17. Jurgens, D.: That's what friends are for: Inferring location in online social media platforms based on social relationships. In: ICWSM (2013)
18. Kinsella, S., Murdock, V., O'Hare, N.: I'm eating a sandwich in glasgow: modeling locations with tweets. In: Proceedings of the 3rd international workshop on Search and mining user-generated contents. pp. 61–68. ACM (2011)
19. Li, R., Wang, S., Deng, H., Wang, R., Chang, K.C.C.: Towards social user profiling: unified and discriminative influence model for inferring home locations. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1023–1031. ACM (2012)
20. Mahmud, J., Nichols, J., Drews, C.: Home location identification of twitter users. ACM Transactions on Intelligent Systems and Technology (TIST) 5(3), 47 (2014)
21. McClendon, S., Robinson, A.C.: Leveraging geospatially-oriented social media communications in disaster response. International Journal of Information Systems for Crisis Response and Management (IJISCRAM) 5(1), 22–40 (2013)
22. McGee, J., Caverlee, J., Cheng, Z.: Location prediction in social media based on tie strength. In: Proceedings of the 22nd ACM international conference on Conference on information & knowledge management. pp. 459–468. ACM (2013)
23. Rendle, S.: Factorization machines with libfm. ACM Transactions on Intelligent Systems and Technology (TIST) 3(3), 57 (2012)
24. Robusto, C.: The cosine-haversine formula. American Mathematical Monthly pp. 38–40 (1957)
25. Roller, S., Speriosu, M., Rallapalli, S., Wing, B., Baldridge, J.: Supervised text-based geolocation using language models on an adaptive grid. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 1500–1510. Association for Computational Linguistics (2012)
26. Rout, D., Bontcheva, K., Preoţiuc-Pietro, D., Cohn, T.: Where's@ wally?: a classification approach to geolocating users based on their social ties. In: Proceedings of the 24th ACM Conference on Hypertext and Social Media. pp. 11–20. ACM (2013)
27. Sakaki, T., Okazaki, M., Matsuo, Y.: Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on World wide web. pp. 851–860. ACM (2010)
28. Weng, J., Lee, B.S.: Event detection in twitter. ICWSM 11, 401–408 (2011)
29. Wing, B., Baldridge, J.: Hierarchical discriminative classification for text-based geolocation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. pp. 336–348 (2014)
30. Wing, B.P., Baldridge, J.: Simple supervised document geolocation with geodesic grids. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 955–964. Association for Computational Linguistics (2011)
31. Yin, J., Lampert, A., Cameron, M., Robinson, B., Power, R.: Using social media to enhance emergency situation awareness. IEEE Intelligent Systems (6), 52–59 (2012)