Near Real-time Geolocation Prediction in Twitter Streams via Matrix Factorization Based Regression

Nghia Duong-Trung
Information Systems and Machine Learning Lab
Universitätsplatz 1
31141 Hildesheim, Germany
duongn@ismll.uni-hildesheim.de

Nicolas Schilling
Information Systems and Machine Learning Lab
Universitätsplatz 1
31141 Hildesheim, Germany
schilling@ismll.uni-hildesheim.de

Lars Schmidt-Thieme
Information Systems and Machine Learning Lab
Universitätsplatz 1
31141 Hildesheim, Germany
schmidt-thieme@ismll.uni-hildesheim.de

ABSTRACT

Previous research on content-based geolocation in general has developed prediction methods via conducting pre-partitioning and applying classification methods. The input of these methods is the concatenation of individual tweets during a period of time. But unfortunately, these methods have some drawbacks. They discard the natural real-values properties of latitude and longitude as well as fail to capture geolocation in near real-time. In this work, we develop a novel generative content-based regression model via a matrix factorization technique to tackle the near real-time geolocation prediction problem. With this model, we aim to address a couple of un-answered questions. First, we prove that near real-time geolocation prediction can be accomplished if we leave out the concatenation. Second, we account the real-values properties of physical coordinates within a regression solution. We apply our model on Twitter datasets as an example to prove the effectiveness and generality. Our experimental results show that the proposed model, in the best scenario, outperforms a set of state-of-the-art regression models including Support Vector Machines and Factorization Machines by a reduction of the median localization error up to 79%.

CCS Concepts

•**Information systems** → Blogs; Social networking sites; Geographic information systems; •**Theory of computation** → Models of learning;

Keywords

Geolocation; Matrix Factorization; Regression; Twitter

1. INTRODUCTION

One of the early pioneer papers about geolocation in Twitter streams was published back in 2010 [3]. In this work, the authors concatenate all user's tweets during a specified duration into one single representative document. A geographical variable was introduced to model the topic distribution across different geographical regions. Hence, the observed user locations are generated from geographical regions and the region variable in topic models is associated with the user's geographical location. Previous works have been developed to solve the classification task to predict user location. Based on these initial ideas, a wide range of follow-up research focuses on solving this type of prediction problem.

Prerequisites to these methods are how the earth's surface is represented. Geolocations can be visualized as points or clusters based on a pre-partitioning of regions into discrete sub-regions using city locations, location indicative words as well as vernacular expressions with the aid of comprehensive gazetteers. But unfortunately, the text of tweets are usually banal and ad hoc where named location words might not exist. To solve this problem, using an invisible grid to partition the earth's surface then got attention in many state-of-the-art models. One can develop the simplest uniform rectangular grid with cells of equal-sized degrees [11]. Furthermore, many research directions focus on more advanced grids by using either an adaptive grid based on *k*-d trees [9] or a hierarchical structure [10].

As being discussed, 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 work, in contrast to these aforementioned approaches and to the best of our knowledge, we firstly investigate a generative content-based regression model to predict exact latitude and longitude. Furthermore, we also analyze the near real-time geolocation in general.

2. PROBLEM STATEMENT

The problem we study in this paper is to predict the near real-time geolocation, e.g. the geolocation of a user at a particular tweet's posting time, using content information only. 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},

CIKM'16 , October 24-28, 2016, Indianapolis, IN, USA

DOI: http://dx.doi.org/10.1145/2983323.2983887

a test D^{test} and a validation D^{valid} set. The D^{valid} will be used for hyperparameter optimization later. We denote m, l and v as the number of tweets in D^{train}, D^{test} and D^{valid} 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 i 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. Furthermore, we denote the average geolocation of all users in the training set as $\bar{\mathbf{y}}_U = (\bar{y}^{lat}_U, \bar{y}^{lon}_U)$. 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 prediction model $f : \mathbb{R}^n \rightarrow \mathbb{R}^2$ which maps tweets to geolocations such that for some test data $X^{test} \in \mathbb{R}^{v \times n}$, the sum of localization errors

$$\sum_{i=1}^{v} d(f(X^{test}_i), Y^{test}_i) \tag{1}$$

is minimal. We denote the set of ground-truth labels for the test data as $Y^{test} \in \mathbb{R}^{v \times 2}$.

3. METHODOLOGY

In this section, we demonstrate our matrix factorization based regression model. We also present a learning algorithm for our approach which is based on stochastic gradient descent.

3.1 The Matrix Factorization Based Regression Model

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 words of each tweet to reconstruct X as:

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

where 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 combination to predict the geolocation of a user at a posting time:

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

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.

3.2 Learning from Training Data

We have to learn parameters T^{lat}, T^{lon}, W^{lat}, W^{lon}, θ, ϕ, where the W matrices are only used for reconstructing X and the T matrices are used to predict the actual geolocation. We optimize the prediction of the gelocation as well as the factorization of X for the absolute error. In order to avoid 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:

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

where the overall 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 $\left\| X^{train} - T^{lat} W^{lat} \right\|^2_F$ is the residual error of transforming X into T^{lat}, W^{lat}. The regularization terms $\lambda_\phi \left\| \phi \right\|^2$, $\lambda_T \left\| T^{lat} \right\|^2_F$ and $\lambda_W \left\| W^{lat} \right\|^2_F$ are multiplied by regularization parameters λ_ϕ, λ_T, and λ_W that control the amount of regularization. To solve the above optimization tasks, we apply the Stochastic Gradient Descent together with Adaptive Subgradient Method to control the learning rate.

3.3 Prediction of 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. We denote the latent tweet representations for the test tweets by T'^{lat} and T'^{lon} for latitude and longitude respectively and factorize X^{test} as

$$X^{test} \approx T'^{lat} W^{lat} \tag{5}$$

as well as the respective term for longitude.

As we can see in Equation 5, we reuse W^{lat} from the learning phase. Subsequently, in the fold-in, we define the objective function that we need to minimize for T'^{lat} as follows:

$$\begin{aligned} \mathcal{L}^{lat}\left(X^{test}, T'^{lat} W^{lat} \right) &= \frac{1}{|X^{test}|} \\ \left(\left\| X^{test} - T'^{lat} W^{lat} \right\|^2_F + \lambda_{test} \left\| T'^{lat} \right\|^2_F \right) \end{aligned} \tag{6}$$

where the objective function that we need to minimize for T'^{lon} is similar. Finally, we can 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. In this situation, we use the median geolocation of all users of the training data as:

$$\bar{\mathbf{y}}_{u_l} = \begin{cases} \bar{\mathbf{y}}_{u_l}, & \text{if } u_l \in D^{train} \\ \bar{\mathbf{y}}_U, & \text{otherwise} \end{cases} \tag{7}$$

Algorithm 1 illustrates how our proposed model works. In order to obtain a good predictive performance, we also need to carefully tune the hyperparameters in our model. We tune the hyperparameters by assessing the validation performance of our model and choosing the hyperparameter configuration which performs best. The hyperparameter configuration that achieves the smallest error on the validation set is then used by the model to evaluate on the test set.

Algorithm 1 The matrix factorization based regression model

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 $T^{lat}, T^{lon}, W^{lat}, W^{lon}, \phi, \theta, T'^{lat}, T'^{lon}$
// Learning
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
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
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 $d(\mathbf{y}, \hat{\mathbf{y}})$

4. EXPERIMENTS

In this section, we describe the datasets that we use as well as the data preprocessing procedure. Due to performing a regression task instead of a classification task we compare our model directly with other state-of-the-art regression models, namely linear regression model (LReg) [4], Support Vector Machines (SVM) [1] and Factorization Machines (FM) [8].

4.1 Dataset

We have worked with three publicly available tweet datasets containing geolocation information and compiled them to fit the near real-time scenario. The first dataset comprises the tweets posted within the United States, the second dataset contains tweets localized to north America and the last one comprises the tweets from all over the world. Through this, we evaluate our model's effectiveness and generality within different geographical scopes from a country to the whole world. We randomly split *all tweets of each user* by a 60/20/20 scheme, denoted as LocalRandom (LR). Moreover, we also investigate how our model works in case of 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 collected by [3], and was later also used in [2, 11, 7]. 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 resulting dataset contains 377,616 tweets posted by 9,475 users.

NA. The second dataset was collected by [9] and later used by [10, 6]. This dataset contains tweets within north America, including the United States, parts of Canada and Mexico from September 4th to November 29th, 2011. Because the NA dataset only contains user IDs and tweet IDs, 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 [5] and later used by [10, 6]. The dataset comprises tweets from all over the world. As being described above with the NA dataset, we also apply the same retrieving procedure. The resulting 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.

4.2 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.

4.3 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 $d : \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}$ is the great circle distance between two geographical coordinate pairs. The Haversine distance can be calculated by:

$$d(\mathbf{y}, \hat{\mathbf{y}}) = 2r \arcsin\left(\left(\sin^2\left(\frac{|\hat{y}^{lat} - y^{lat}|}{2}\right) + \cos(y^{lat})\cos(\hat{y}^{lat})\sin^2\left(\frac{|\hat{y}^{lon} - y^{lon}|}{2}\right)\right)^{\frac{1}{2}}\right) \tag{8}$$

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. 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 localization error d in kilometers between the ground-truth geolocation $\mathbf{y}$ and the predicted geolocation $\hat{\mathbf{y}}$.

4.4 Results and Comparison

We compare our models to the user average location (AL), and a couple of state-of-the-art regression models. We report the results both in mean and median localization errors in Table 1.

Table 1: The mean and median Haversine distance error in km of all models. The best distances are in bold.

Corpus	LR_US		LR_NA		GR_US		GR_NA		GR_WORLD	
Model	mean	median	mean	median	mean	median	mean	median	mean	median
User average location	29.64	0.67	164.45	7.20	27.07	0.66	177.51	7.24	3195.64	2668.71
Linear regression [4]	56.47	16.61	220.29	66.73	54.37	16.40	233.88	68.40	3196.94	2645.66
SVM with RBF kernel [1]	34.63	7.81	157.81	8.42	32.29	8.22	171.72	10.23	3179.57	2654.17
Factorization machines [8]	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**

Table 2: Statistics of the US, NA and WORLD datasets.

	US	NA	WORLD
#tweets	377,616	226,595	121,327
#users	9,475	10,950	80,179
$\|V\|$	5216	5292	5145
average tweet's length	12	5	5

In the AL model, the model discards all the content of tweets and only accounts for physical coordinates. If a user appears in both training and test sets, the predicted geolocation is the average of his geolocation in the training set. Otherwise, if a user only exists in the test set, the average geolocation of the whole training set is then used as prediction. For the LReg, SVM and FM, we run them separately to predict latitude and longitude. Subsequently, we combine the predicted latitude and longitude to conduct a final localization error. We also apply a grid-search mechanism applying on a validation set to find the best hyperparameter configurations for each prediction of latitude and longitude.

For our proposed model, we optimize it for the absolute error loss. On each dataset, we repeat running the models 10 times and take the average results. We can see that all other regression models on average do not perform that well, mainly because of using the extremely sparse $5,200$ TF.IDF scores. Our model, however, maps each tweet individually into an eight-dimensional latent feature space and uses those features for prediction.

The results show that our model outperforms all competitors with the smallest mean and median localization errors. Thus, we can reduce the error in comparison to the AL model on the LR_US dataset by 1.7% and 1.5%, whereas the reduction amounts to 2.3% and 1.5% on the GR_US dataset. For the LR_NA and GR_NA datasets, the reduction amounts account for 4.4%, 3.5%, 4.2% and 0.7% for the mean and median respectively. For the GR_WORLD dataset, we notice the significant improvement where the error is reduced by approximately 21% and 79% for the mean and median. We conclude that optimizing for the median localization error yields a bigger improvement as the region gets larger and therefore the average location does not work that well.

5. CONCLUSIONS

In our work, we propose an effective matrix factorization based regression approach to entail an early attempt to solve the task of near real-time geolocation prediction using publicly available datasets that were originally exploited for classification models. We apply our model on Twitter datasets to prove its potential of effectiveness and the generalization abilities. Starting from the model's significant performance especially when applying it in a wide geoscope, we can develop a generative regression approach for this type of geolocation prediction scenario where as the input might be generalized to textual descriptions of images or any types of text in social media. Through this, our model can further predict the user trajectory as he continues posting tweets along a path. Moreover, we can further investigate on the affection of tweet concatenation or the number of tweets needed to achieve an acceptable distance error and time span.

6. REFERENCES

[1] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

[2] J. Eisenstein, A. Ahmed, and E. P. Xing. Sparse additive generative models of text. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 1041–1048, 2011.

[3] J. Eisenstein, B. O'Connor, N. A. Smith, and E. P. Xing. A latent variable model for geographic lexical variation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1277–1287. Association for Computational Linguistics, 2010.

[4] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[5] B. Han, P. Cook, and T. Baldwin. Geolocation prediction in social media data by finding location indicative words. *Proceedings of COLING 2012: Technical Papers*, pages 1045–1062, 2012.

[6] B. Han, P. Cook, and T. Baldwin. Text-based twitter user geolocation prediction. *Journal of Artificial Intelligence Research*, pages 451–500, 2014.

[7] L. Hong, A. Ahmed, S. Gurumurthy, A. J. Smola, and K. Tsioutsiouliklis. Discovering geographical topics in the twitter stream. In *Proceedings of the 21st international conference on World Wide Web*, pages 769–778. ACM, 2012.

[8] S. Rendle. Factorization machines with libfm. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(3):57, 2012.

[9] S. Roller, M. Speriosu, S. Rallapalli, B. Wing, and J. Baldridge. 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*, pages 1500–1510. Association for Computational Linguistics, 2012.

[10] B. Wing and J. Baldridge. Hierarchical discriminative classification for text-based geolocation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 336–348, 2014.

[11] B. P. Wing and J. Baldridge. 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*, pages 955–964. Association for Computational Linguistics, 2011.