Geolocation with Subsampled Microblog Social Media

Miriam Cha
Harvard University

Youngjune Gwon*
Harvard University

H. T. Kung
Harvard University

ABSTRACT

We propose a data-driven geolocation method on microblog text. Key idea underlying our approach is sparse coding, an unsupervised learning algorithm. Unlike conventional positioning algorithms, we geolocate a user by identifying features extracted from her social media text. We also present an enhancement robust to erasure of words in the text and report our experimental results with uniformly or randomly subsampled microblog text. Our solution features a novel two-step procedure consisting of upconversion and iterative refinement by joint sparse coding. As a result, we can reduce the amount of input data required by geolocation while preserving good prediction accuracy. In the light of information preservation and privacy, we remark potential applications of these results.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Clustering*; I.2.6 [**Artificial Intelligence**]: Learning—*Unsupervised feature learning*

Keywords

Geolocation; joint sparse coding; text subsampling; Twitter

1. INTRODUCTION

Traditionally, geolocation involves the detection and related computational processing of beacon signals used in a positioning system such as GPS. We consider a data-driven framework for geolocation that leverages the increased availability of geotagged social media data. In particular, we aim to develop an algorithm for estimating geocoordinates of a social network user by learning features from the user's social media text. We also extend the algorithm to take lossy, *subsampled* text data as input to geolocation prediction while preserving accurate geolocation estimates.

Geolocation information serves valuable context for social media. Recent research [1, 2] has experimented with latent variable models trained on the Twitter microblog text ("tweets") for geolocation. To achieve decent geolocation accuracy, a geographic topic model must be trained with a sufficient amount of labeled training examples. The fact that only 2.2% of tweets are geotagged [3] indicates the difficulty of supervised learning.

In this paper, we describe a model-free, geolocation method driven by a relatively small labeled dataset. Our key component is sparse coding, an unsupervised algorithm that learns a feature mapping for the raw text input. It is advantageous for prediction algorithms to operate on *sparse* (feature) representations of the raw data. We exploit the mapping to build a lookup table of reference geocoordinates and search it using the sparse feature vector computed on the text input. We apply k-Nearest Neighbor (k-NN) similarity matching in the feature domain for geolocation.

This paper also studies the preservation of geographic information after discarding words in the text. While the cause of such discarding could be unknown or randomly introduced, a purposeful usage would be to enhance privacy of a user, especially if the computing for geolocation is done remotely (*e.g.*, in the cloud). We demonstrate robustness of our approach by experimenting with uniformly and randomly subsampled tweet text. While subsampling is a simple method of dimensionality reduction, we face the difficulty unique for text data where information erasure is highly irregular depending on which word gets discarded.

To address this challenge, we propose a novel two-step procedure consisting of upconversion and refinement inspired by joint sparse coding for image super-resolution [4]. We will show that the refinement step can be iteratively repeated to produce better geolocation estimates.

Geolocation based on comprehending social media text is an ongoing, hard research problem [1, 2, 5]. Prediction with heavily subsampled text (*e.g.*, 50% of words in the text) is even more challenging. Our work here assesses the feasibility of a learning approach robust to subsampling. The results of this paper may be useful to develop secure applications for devices with limited computing power.

Rest of this paper is organized as follows. In Section 2, we describe our data-driven framework for text-based geolocation. Section 3 will present our approach based on subsampled text data. In Section 4, we discuss the results from an experimental evaluation on the CMU GeoText dataset [6], and Section 5 concludes the paper.

2. TEXT-BASED GEOLOCATION VIA SPARSE CODING

To alleviate the scarcity of labeled training examples in practice, we focus on unsupervised feature learning based on sparse coding and dictionary training.

*The author was a Ph.D. student at Harvard while the work was done. He now works for MIT Lincoln Laboratory in Lexington, MA.

MM'15, October 26–30, 2015, Brisbane, Australia.

DOI: http://dx.doi.org/10.1145/2733373.2806357.

2.1 Sparse coding for text data

We use sparse coding as the basic means to extract features from text. A text document, however, cannot be directly applied to sparse coding. Instead, we convert text to a numeric form in a procedure called "embedding." Let `vocab` denote a collection of unique words appearing in documents with size $V = |\texttt{vocab}|$. In *binary-bag-of-words* (BW) embedding scheme, a text document containing W words is represented as a bit vector $\mathbf{w}_{BW} \in \{0,1\}^V$. The ith element in $\mathbf{w}_{BW}$ is 1 if the word `vocab[`i`]` has appeared in the text. We also use *word-sequence* (WS) embedding $\mathbf{w}_{WS} \in \{1,\dots,V\}^W$, where w_i, the ith element in $\mathbf{w}_{WS}$, represents `vocab[`w_i`]`. Sparse coding takes in an unit input vector called patch drawn from data. We denote patch $\mathbf{x} \in \mathbb{R}^N$, a consecutive subvector taken from $\mathbf{w}_{BW}$ or $\mathbf{w}_{WS}$ for an input to sparse coding.

Given an input $\mathbf{x} \in \mathbb{R}^N$, sparse coding solves for a representation $\mathbf{y} \in \mathbb{R}^K$ in the following optimization problem:

$$\min_{\mathbf{D},\mathbf{y}} \|\mathbf{x} - \mathbf{D}\mathbf{y}\|_2^2 + \lambda\psi(\mathbf{y}) \tag{1}$$

where input $\mathbf{x}$ is represented as a sparse linear combination of basis vectors in an overcomplete dictionary $\mathbf{D} \in \mathbb{R}^{N\times K}$ ($K > N$). The solution $\mathbf{y}$ is the feature representation for $\mathbf{x}$. The system $\mathbf{x} = \mathbf{D}\mathbf{y}$ is underdetermined (*i.e.*, more unknowns than equations) and needs an extra constraint for assuring unique solution. As in the second term of Eq. (1), sparse coding regularizes on the ℓ_0- or ℓ_1-norm of $\mathbf{y}$ for $\psi(.)$ with $\lambda > 0$. Because the ℓ_0-norm of a vector is the number of nonzero elements, it can precisely serve the regularization purpose. Finding the sparsest ℓ_0-minimum solution in general, however, is known to be NP-hard. The ℓ_1-minimization with LASSO [7] or Basis Pursuit [8] is often preferred. Recently, it is known that the ℓ_0-based greedy algorithms such as Orthogonal Matching Pursuit (OMP) [9] can run fast.

Dictionary learning for sparse coding is done by an unsupervised, data-driven process incorporating two separate optimizations. It first computes sparse code for each training example using the current dictionary. Then, the reconstruction error from the computed sparse codes is used to update each basis vector in the dictionary. We use K-SVD algorithm [10] for dictionary learning.

2.2 Preprocessing patches

We can enhance the quality of unsupervised learning by preprocessing patches. For example, binary-bag-of-words embedding produces highly sparse bit vectors that are difficult for sparse coding to learn meaningful features. We can preprocess the patches of embedded text by removing the mean value and scaling with standard deviation. Another technique called whitening makes input data less redundant such that dictionary learning can be more effective. Combining these into one integrated procedure, we preprocess a batch of n input patches taken from embedded text vectors by whitening.

1. Remove mean $\mathbf{x}^{(i)} := \mathbf{x}^{(i)} - \frac{1}{n}\sum_{i=1}^{n}\mathbf{x}^{(i)}$;
2. Compute covariance matrix $\mathbf{C} = \frac{1}{n-1}\sum_{i=1}^{n}\mathbf{x}^{(i)}\mathbf{x}^{(i)\top}$;
3. Do eigendecomposition `[U,Λ] = eig(C)`;
4. Compute $\mathbf{x}_{\text{white}} = (\mathbf{\Lambda}+\epsilon\mathbf{I})^{-1/2}\mathbf{U}^\top\mathbf{x}$, where ϵ is a small positive value for regularization.

2.3 Baseline geolocation method

Our baseline geolocation method consists of the following steps in the training phase.

1. (Text embedding) perform binary or word-sequence embedding of text data using `vocab`;
2. (Unsupervised learning) feed patches drawn from unlabeled embedded text vectors to sparse coding and learn basis vectors for dictionary $\mathbf{D}$;
3. (Feature extraction) using the dictionary $\mathbf{D}$ learned during unsupervised learning, for given labeled training patches $\{(\mathbf{x}^{(1)}, l^{(1)}), (\mathbf{x}^{(2)}, l^{(2)}), \dots\}$, perform sparse coding on $\{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots\}$ and obtain sparse codes $\{\mathbf{y}^{(1)}, \mathbf{y}^{(2)}, \dots\}$ with their associated labels;
4. (Feature pooling) aggregate features by max pooling over a group of M sparse codes extracted from the same document and obtain pooled sparse code $\mathbf{z}$ such that the jth element in $\mathbf{z}$, $z_j = \max(y_{1,j}, y_{2,j}, \dots, y_{M,j})$ where $y_{i,j}$ is the jth element from $\mathbf{y}_i$, the ith sparse code in the pooling group;
5. (Tabularization of reference geocoordinates) build a lookup table of geocoordinates associated with pooled sparse codes $\mathbf{z}$ from labeled data patches.

Note that each label contains geocoordinates in the form $l^{(i)} = \{lat, lon\}$.

The baseline method works in the following manner for the testing phase. When text data (tweets) of unknown geocoordinates arrive, we perform preprocessing, feature extraction via sparse coding, and max pooling. Using the max-pooled sparse code of the tweets, we find the k pooled sparse codes from the lookup table that are closest in *cosine similarity*. We take the average geocoordinates of the k-NNs.

2.4 Grid-based voting scheme for k-NN

We accompany a simple voting scheme for k-NN. We lay out a grid over the all k-NN geocoordinates. Each k-NN casts a vote to its corresponding grid. We identify the grid that receives the most votes and take the average of geocoordinates in the selected grid as the final geolocation estimate.

3. GEOLOCATION FROM SUBSAMPLED TEXT

Using a linear subsampling matrix (*i.e.*, uniform or random), we consider two blind subsampling strategies. First, we subsample after binary-bag-of-words embedding. Secondly, we can subsample the raw text before embedding. Blind text subsampling concerns a tradeoff between amount of input data required for geolocation and prediction accuracy. This section proposes an enhancement robust to the effect of blind text subsampling on degradation of geolocation accuracy.

3.1 Subsampling binary-bag-of-words

Given $\mathbf{w}_{BW}$, we perform either uniform or random subsampling. For uniform subsampling, every αth component of $\mathbf{w}_{BW}$ is discarded for some integer α. Similarly, random subsampling discards the equivalent number of components randomly from $\mathbf{w}_{BW}$. As a result, the subsampled embedded vector has a smaller dimension than the original binary bag-of-words vector. Subsampling $\mathbf{w}_{BW}$ has an identical effect as embedding the raw text based on subsampled `vocab`.

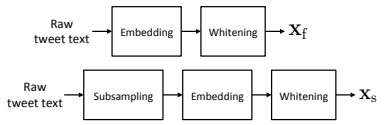

Figure 1: Patches $\{\mathbf{x}_\mathrm{f}, \mathbf{x}_\mathrm{s}\}$ from full and subsampled text

3.2 Subsampling raw text

Unlike $\mathbf{w}_{BW}$, $\mathbf{w}_{WS}$ converts text to a numeric form while retaining the ordering of words. Thus subsampling $\mathbf{w}_{WS}$ is subsampling raw text vector. Uniform subsampling discards every αth words, and random subsampling discards the same number of words randomly. The subsampled raw texts are embedded based on the binary-bag-of-words scheme, resulting the final subsampled vector of dimension V.

3.3 Joint sparse coding for upconversion and refinement

The effect of subsampling raw text is more devastating than subsampling binary-bag-of-words vectors. In this section, we propose a variation of joint sparse coding that can recover the original text and refine the recovery from the subsampled text for better geolocation performance.

Let us denote the pair $\{\mathbf{x}_\mathrm{f}, \mathbf{x}_\mathrm{s}\}$ patches drawn from the binary-bag-of-words vectors embedded on full and subsampled text, as seen in Figure 1. The sparse coding problems for $\mathbf{x}_\mathrm{f} \in \mathbb{R}^N$ and its subsampled counterpart $\mathbf{x}_\mathrm{s} \in \mathbb{R}^N$ are

$$\min_{\mathbf{D}_\mathrm{f}, \mathbf{y}_\mathrm{f}} \|\mathbf{x}_\mathrm{f} - \mathbf{D}_\mathrm{f}\mathbf{y}_\mathrm{f}\|_2^2 + \lambda_\mathrm{f}\|\mathbf{y}_\mathrm{f}\|_1 \tag{2}$$

and

$$\min_{\mathbf{D}_\mathrm{s}, \mathbf{y}_\mathrm{s}} \|\mathbf{x}_\mathrm{s} - \mathbf{D}_\mathrm{s}\mathbf{y}_\mathrm{s}\|_2^2 + \lambda_\mathrm{s}\|\mathbf{y}_\mathrm{s}\|_1 \tag{3}$$

where $\mathbf{D}_\mathrm{f}$ and $\mathbf{D}_\mathrm{s}$ are dictionaries from full and subsampled text, respectively. Image super-resolution [4] takes advantage of the shared sparse code between the high- and low-resolution pair of patches for the same image such that one can recover a high-resolution image from the low-resolution version. Similarly, if the sparse code for the full and subsampled is shared (*i.e.*, $\mathbf{y}_\mathrm{f} = \mathbf{y}_\mathrm{s}$), we can attempt to recover the full by using the subsampled. We formulate a new joint optimization that forces the sharing of the sparse code $\mathbf{v} = \mathbf{y}_\mathrm{f} = \mathbf{y}_\mathrm{s}$ between the full and subsampled pair of patches

$$\min_{\mathbf{D}_\mathrm{u}, \mathbf{D}_\mathrm{d}, \mathbf{v}} \|\mathbf{x}_\mathrm{f} - \mathbf{D}_\mathrm{u}\mathbf{v}\|_2^2 + \|\mathbf{x}_\mathrm{s} - \mathbf{D}_\mathrm{d}\mathbf{v}\|_2^2 + \lambda_\mathrm{v}\|\mathbf{v}\|_1 \tag{4}$$

where $\mathbf{D}_\mathrm{u}$ is the upconversion dictionary, and $\mathbf{D}_\mathrm{d}$ for downconversion.

In the unsupervised learning stage, we first solve for $\mathbf{D}_\mathrm{u}$, $\mathbf{D}_\mathrm{d}$, and $\mathbf{v}$ via joint sparse coding of Eq. (4). Using the learned $\mathbf{D}_\mathrm{u}$, we obtain an upconversion estimate $\hat{\mathbf{x}}_\mathrm{f} = \mathbf{D}_\mathrm{u}\mathbf{v}$. We can refine $\hat{\mathbf{x}}_\mathrm{f}$ in another joint optimization

$$\min_{\mathbf{D}_\mathrm{r}, \mathbf{D}_\mathrm{q}, \mathbf{w}} \|\mathbf{x}_\mathrm{f} - \mathbf{D}_\mathrm{r}\mathbf{w}\|_2^2 + \|\hat{\mathbf{x}}_\mathrm{f} - \mathbf{D}_\mathrm{q}\mathbf{w}\|_2^2 + \lambda_\mathrm{w}\|\mathbf{w}\|_1. \tag{5}$$

Here, we explicitly look for $\mathbf{D}_\mathrm{r}$ and $\mathbf{D}_\mathrm{q}$ that make the refinement of $\hat{\mathbf{x}}_\mathrm{f}$ possible. In the supervised learning stage, we perform sparse coding with (labeled) subsampled data patches using the learned $\mathbf{D}_\mathrm{d}$ and yield joint sparse codes $\mathbf{v}$. The joint sparse codes are applied to feature pooling and tabularization of reference geocoordinates in steps 4 and 5 of baseline geolocation method.

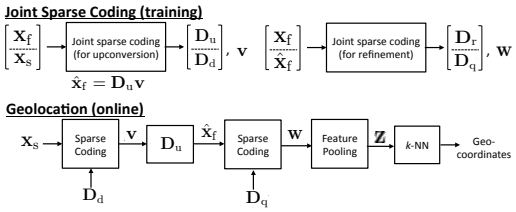

Figure 2: Full system pipelines

When subsampled tweets from unknown geocoordinates arrive, the geolocation pipeline consists of upconversion and refinement. We can iteratively repeat the refinement step. For enhanced geolocation with subsampled text, we use the refined feature vector $\mathbf{w}$. Figure 2 illustrates our full pipelines.

4. EVALUATION

In this section, we empirically evaluate the proposed approaches using the CMU geo-tagged microblog corpus [6]. We train our baseline geolocation method using the full data samples. We also train our method with uniformly and randomly subsampled raw text and binary-bag-of-words (embedded) text vectors to analyze the effect of subsampling on the accuracy degradation. We will discuss the improved geolocation performance by our joint sparse coding method for upconversion and refinement on subsampled text data.

4.1 Data

GeoText is a Twitter text dataset comprising 377,616 tweets by 9,475 users from 48 contiguous US states and Washington D.C. Each document in the dataset is concatenation of the entire tweets by a single user collected over one week. All documents include the user location information provided as GPS-assigned latitude and longitude values. The document is a sequence of integer numbers ranging 1 to 5,216, where each number represents the position in `vocab`.

4.2 Experimental methodology

For all our experiments, we have cut the dataset into five folds such that `fold=user_id%5`, following Eisenstein *et al.* [1]. Folds 1–4 are used for training, and fold 5 for testing. We have embedded the entire text data from each user to binary-bag-of-words and word-sequence vectors and applied uniform or random subsampling by $\alpha = 2, 3, 4$, and 6.

We precondition patches with PCA whitening before sparse coding. After numerous experiments, we have determined to use patch size $N = 64$. We have used OMP, a greedy-ℓ_0 sparse encoder, and K-SVD for dictionary learning. Other sparse coding parameters are also determined experimentally. We have used a dictionary size $K \approx 10N$, sparsity level S for $0.1N \leq S \leq 0.4N$ (S is number of nonzero elements in $\mathbf{y}$). We use max pooling factors M in 10s.

In supervised learning, we have built the table of reference geocoordinates for k-NN, using the max-pooled sparse code as the feature for lookup. This results in the lookup table of more than 7,500 entries. We have found good geolocation accuracy with $10 \leq k \leq 40$.

For the metric of performance evaluation, we use the median distance error between the predicted and ground-truth geocoordinates measured in kilometers. We note that related previous research has regarded median distance error more important than the mean. It is required to approximate the

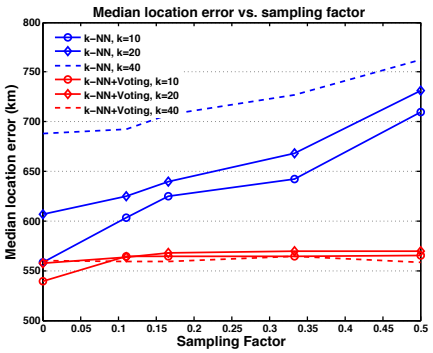

Figure 3: Median geolocation errors of subsampling binary-bag-of-words against various sampling factors

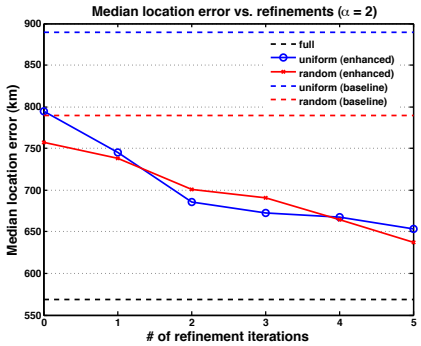

Figure 4: Median geolocation errors of uniform and random subsampling the raw text with $\alpha = 2$.

great-circle distance between any two locations on earth because of its round surface. We use the Haversine formula [11].

4.3 Results and discussion

In Figure 3, we present the effect of grid-based voting scheme on the geolocation errors using uniformly subsampled binary-bag-of-words text. Notice that the grid-based voting scheme is robust to subsampling binary-bag-of-words text. However, the effect of subsampling raw text is stronger than subsampling binary-bag-of-words vectors, and we have experienced that the grid-based voting scheme is not as effective when applied to subsampled raw text. Therefore we use our proposed upconversion and iterative refinement scheme by joint sparse coding that is robust to subsampling raw text. We present the median geolocation errors of uniform and random 2x subsampling (*i.e.*, $\alpha = 2$) on the raw text in Figure 4. We gradually increase the number of refinement steps to observe changes in the geolocation error. Applying our baseline geolocation method on full text gives 568 km error. As expected, discarding 50% of words significantly increases geolocation errors, 794 km for uniform subsampling and 757 km for random subsampling. Remarkably, multiple iterations of refinement step can help mitigate the geolocation error caused by heavy subsampling. After five iterations of refinement step, geolocation error for uniform subsampling decreases to 652 km and random subsampling to 636 km. Notice that these geolocation errors are only 84 km and 68 km higher than using the full text.

Figure 5 depicts the median geolocation errors against various uniform subsampling factors (in $1/\alpha$) for subsampling only, subsampling with enhancement by upconversion, and

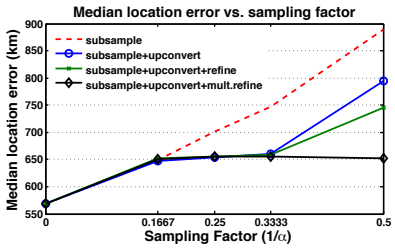

Figure 5: Median geolocation errors of uniform subsampling and enhancements against various sampling factors

subsampling with enhancements by both upconversion and single or multiple (5) iterations of refinement. As uniform and random subsampling results are comparable, we only report uniform subsampling results. Compared to subsampling only, upconversion and multiple iterations of refinement after subsampling is more robust to increased sampling factor.

5. CONCLUSION

We have presented a geolocation method based on sparse coding of microblog text data. We achieve the median geolocation errors of 568 km for full and 636 km even under 2x subsampling on the GeoText dataset. The proposed upconversion and iterative refinement scheme by joint sparse coding proves to be successful in drawing out the correlation between the full and subsampled text pair. The geolocation accuracy degradation is only by 12%, even if we have halved words in the text. The main contributions of this paper are the refinement scheme and its application to geolocation for subsampled microblog text. For future work, we plan to improve feature learning schemes by introducing hierarchy and evaluate our methods using both US and worldwide data.

Acknowledgments

This work is supported by the National Science Foundation Graduate Research Fellowship under Grant No. DGE1144152, the Naval Supply Systems Command award under the Naval Postgraduate School Agreement No. N00244-15-0050, and gifts from the Intel Corporation.

6. REFERENCES

[1] J. Eisenstein, B. O'Connor, N.A. Smith, and E.P. Xing. A Latent Variable Model for Geographic Lexical Variation. In *EMNLP*, 2010.

[2] L. Hong, A. Ahmed, S. Gurumurthy, A.J. Smola, and K. Tsioutsiouliklis. Discovering Geographical Topics in the Twitter Stream. In *WWW*, 2012.

[3] C. Weidemann. Social Media Location Intelligence: The Next Privacy Battle—An ArcGIS andd-in and Analysis of Geospatial Data Collected from Twitter.com. *Journal of Geoinfo.*, 2013.

[4] J. Yang, J. Wright, T.S. Huang, and Y. Ma. Image Super-Resolution via Sparse Representation. *IEEE Trans. on Image Processing*, 19(11):2861–2873, 2010.

[5] M. Cha, Y. Gwon, and H. T. Kung. Twitter Geolocation and Regional Classification via Sparse Coding. In *ICWSM*, 2015.

[6] GeoText. CMU Geo-tagged Microblog Corpus. `http://www.ark.cs.cmu.edu/GeoText/`, 2010.

[7] R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of Royal Statistical Society, Series B*, 1994.

[8] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Rev.*, 2001.

[9] J.A. Tropp and A.C. Gilbert. Signal Recovery From Random Measurements via Orthogonal Matching Pursuit. *IEEE Trans. on Information Theory*, 2007.

[10] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Trans. on Sig. Proc.*, 54(11), 2006.

[11] R.W. Sinnott. Virtues of Haversine. *Sky and Telescope*, 1984.