Trajectory User Linking With Self Organizing Trees

Eric Christensen
erichristensen@deloitte.com
Deloitte Consulting LLP
USA

Kade Shoemaker
kshoemaker@deloitte.com
Deloitte Consulting LLP
USA

Steve Hardy
sthardy@deloitte.com
Deloitte Consulting LLP
USA

ABSTRACT

The explosion of mobile devices, GPS technology, and IoT sensors has generated an unprecedented volume of spatiotemporal data, offering immense potential for insights into human behavior and spatial patterns. In this paper, we introduce a novel self-organizing tree algorithm designed to aggregate and organize human mobility data for downstream tasks. Our approach addresses the scalability issues faced by traditional Trajectory User Linking (TUL) techniques, enabling efficient handling of large numbers of users and dynamic addition of new users. This advancement paves the way for more personalized and innovative applications across various industries and. Our findings highlight the transformative potential of self-organizing trees in spatiotemporal data analysis, setting a new benchmark for future research.

CCS CONCEPTS

• **Computing methodologies** → **Modeling methodologies**; **Search with partial observations**; **Supervised learning by classification**.

KEYWORDS

spatiotemporal, geospatial, tree, trajectory, TUL, SOTA

ACM Reference Format:
Eric Christensen, Kade Shoemaker, and Steve Hardy. 2024. Trajectory User Linking With Self Organizing Trees. In *1st ACM SIGSPATIAL International Workshop on Geospatial Anomaly Detection (GeoAnomalies'24), October 29-November 1 2024, Atlanta, GA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3681765.3698450

1 INTRODUCTION

The proliferation of mobile devices, GPS technology, and Internet of Things (IoT) sensors has led to an unprecedented volume of spatiotemporal data being generated daily. At its core, spatiotemporal data consists of ID, latitude, longitude, and timestamp but can be enriched with additional features. Sequences of these points are called trajectories. Spatiotemporal data offers insights into human behavior, environmental changes, and spatial patterns, enabling informed decision-making for organizations that leverage it. There are myriad data sources that do not share a common globally unique identifier. As a result, correlating devices within and across datasets to their user unlocks greater possibilities for personalization.

GeoAnomalies'24, October 29-November 1 2024, Atlanta, GA, USA

ACM ISBN 979-8-4007-1144-2/24/10
https://doi.org/10.1145/3681765.3698450

The procedure of mapping a trajectory to an ID is called Trajectory User Linking (TUL). TUL was first formally addressed in 2017[7], although previous papers centered around similarity metrics for sequences indirectly touched on the topic. In this paper, we explore an unsupervised approach to aggregating collections of user trajectories to create representative patterns of life for individual users. We then use those representations to evaluate similarity to new unlabeled trajectories in order to assign an ID.

2 RELATED WORK

The majority of previous work on TUL is centered around deep learning approaches. Starting with the works of Gao et al.[7], the authors used recurrent neural networks (RNN) to map the trajectory to a latent space which is fed into a linear layer and softmax to generate probabilities for each user in the trained dataset. Since then, authors have added new ways of encoding the trajectory before passing it to the RNN. Ferrero breaks up trajectories into repeated sub-movements[6] that are then used as input features. TULVAE[17] use pretrained point of interest embeddings as input features and add variations to the deep learning architecture. MARC[12] converts all available trajectory semantic attributes into end-to-end learned embeddings including a geohash encoding for spatial data to use as inputs for a RNN. All of these deep learning approaches have challenges with scaling to larger populations - restricting performance to a few hundred users and requiring retraining to introduce new users.

Najjar and Mede[13] took a simpler approach that was focused on the heuristic that a small collection of locations is enough to identify a user. With this concept, they demonstrated the ability to predict for populations on the order of 100k. The paper, however, uses an arbitrary location index as the primary feature - taking the value of the three largest indices as the representative vector to perform Euclidean distance calculations. The reported scores in this paper are spectacular, but we believe the ordering of the indices allowed for data leakage between the train and test sets through a correlation between user ID and location index. While reproducing the results, we found that decoupling the user ID from the location index did not perform as well as cited in their research. We appreciate the simplicity of the approach and use it as inspiration while introducing a temporal component.

2.1 Trajectory-User Linking

In this section, we formally define TUL as demonstrated in previous works[7, 13]. Given a population U containing users $u \in U$, we define a trajectory $\mathbf{t}_u$ for user u with τ samples as

$$\mathbf{t}_u = \{(u, t_{u,1}, b_{u,1}), (u, t_{u,2}, b_{u,2}), ..., (u, t_{u,\tau}, b_{u,\tau})\}$$

where $(u, t_{u,i}, b_{u,i})$ is a tuple representing a record of user u checking-in at time $t_{u,i}$ and location $b_{u,i}$ from an enumerated set of check-in locations.

We define TUL as finding an onto (surjective) mapping $\mathbf{T} \rightarrow U$ where $\mathbf{T}$ is a set of unlabeled trajectories $\mathbf{t}$ belonging to users in the population U.

3 METHOD

The field of bioinformatics has had a rich history of using unsupervised techniques to cluster sequences. One such approach is through a Self-Organizing Tree Algorithm (SOTA) [5, 8, 11]. The authors all created variations of SOTA to create hierarchical representations of sequences. SOTA works similarly to self-organizing maps, but the number of nodes grows dynamically.

For each user $u \in U$, create a binary tree $\mathcal{T}_u$, containing a root node with two children. SOTA algorithms alternate between periods of learning and periods of growth. During learning periods, each sequence is passed in and evaluated against nodes $n \in \mathcal{T}_u$. The best node and its neighbors are identified and updated to more closely match the example sequence. During periods of growth, the network adds children to nodes with high heterogeneity.

We will use a variation of SOTA to cluster similar days for a user together to help overcome sparse data and create representative days that capture the pattern of life of those users. To do this we need to represent each day in the trajectory in a format that can be efficiently compared to a representative day, a way to aggregate the representation of similar days together, and a way to score unlabeled trajectories against a tree. We address these items in the following subsections.

3.1 Data Representation

While there are algorithms to score and accumulate different length trajectories[3, 14, 15] such as dynamic time warping and longest common subsequence, aggregation can be complex[1, 2] and computationally expensive. For this work, we chose to use a fixed sequence length T for each day, d, within a trajectory $\mathbf{t}$ by binning temporally into T time bins. For our experiment, we used 30-minute time bins, setting T to 48 for a 24-hour period.

For the datasets we use in evaluation, there is a pre-identified ID for each location, but any mapping from latitude and longitude to an integer could be used, such as S2 or H3 indexing. For each user u, we sort the locations visited by a user by visit frequency and enumerate. Mapping any given geospatial bin by a function $f_u : B \rightarrow \mathbb{Z}^+$

$$f_u(b) = \min(\text{rank}_u(b), L-1)$$

where $\text{rank}_u(b)$ returns the zero-indexed, without ties, rank of b based on the frequency of user u visits. The most visited location for a user is at the first index, and the second most is at the second index, and so on. We restrict the number of enumerated bins to $L-1$ where any bin that would have an index greater than $L-1$ is instead labeled $L-1$. This will represent a pool of locations that are visited less frequently. We set L to 10 in our experiment, capturing the 10 most frequent locations the user visits.

Once all locations are indexed according to the user, we perform a one-hot encoding adding two extra indices. One extra index will be used for when unlabeled data falls outside of the previously visited locations for the user and the last index is used for time bins with missing data. We average the one-hot encoded vectors across each time bin and create an array of dimension $(T, L+2)$ for each day with the $L+1$ bin filled with a 1 for time bins missing data to produce the sequence $\mathbf{x}$. Once data is in this form, we are able to train a SOTA for each user to use in evaluation.

3.2 Training

Each node n in the tree $\mathcal{T}_u$ carries with it a $(T, L+2)$ dimensional array $\mathbf{a}_n$ for all $n \in \mathcal{T}_u$ that captures an aggregate representation of a pattern of life for user u. Using an input training day for the user, $\mathbf{x^d}$, we score each leaf node in the tree using the mean squared error between the node array and the input array, $MSE(\mathbf{a}_n, \mathbf{x^d})$. The leaf with the lowest error is selected as the best match n^* and is updated by linearly interpolating between the node's representation $\mathbf{a}_{n^*}$ and the training example with a custom weight λ. For our training, we used a weight of 0.2 and iterated over all training examples 5 times before each period of growth. During tree growth, we create two child nodes for any leaf node that was updated with more than one training example. Each child is initialized with a copy of its parent's representation. For our experiment, we terminated after training at a tree depth of 2, yielding at most 4 trained leaf nodes.

At the end of training, we compute the standard deviation σ_n for each representation $\mathbf{a}_n$ using the training examples that best aligned with the leaf node. For each user u, we store a dictionary lookup to map a location to the user-specific geospatial encoding (f_u) and we store the corresponding generated tree $\mathcal{T}_u$. We also create a sparse array of the normalized visit frequency for each geospatial bin visited by the user. We aggregate the visit frequency across the trajectory.

3.3 Evaluation

We evaluate unlabeled trajectories using a multi-step process. First, as with the train data, we create a sparse array of the normalized geospatial bin visit frequency for each trajectory. We multiply this sparse array with the sparse train array to compute a cosine similarity between visited locations. We keep the top k highest cosine similarity scores and accumulate by user. We sort the users by count, then by cosine similarity - we refer to this as our Freq approach. The result of this scoring works as a filter to select a subset of users to score with the more computationally intensive tree.

For each user u identified in the subset, we use their precomputed tree $\mathcal{T}_u$ to score unlabeled test trajectories. We convert a given test trajectory $mathbf{t}_{test}$ into a $(T, L+2, D)$ sized array $\mathbf{x}_{test}$ using the user specific map (f_u) from a bin to an index where D is the number of distinct days covered by the trajectory. Any bin not in the map is set to index L. For each day in the test trajectory, we compute the score against each tree leaf representation $\mathbf{a}_n$ for all generated trees $\mathcal{T}_u$ using the following for each tree:

$$\text{score}(\mathbf{x}_{test}) = \sum_{d=0}^{D-1} \left\{ min_{n \in \mathcal{T}_u} \left(norm \left(\frac{\mathbf{x}_{test}^d - \mathbf{a}_n}{\epsilon + \sigma_n} \right) \right) \right\}$$

where ϵ is a small positive constant, we used 0.0001, $norm$ is the Frobenius norm, and $\mathbf{x}_{test}^d$ is a slice of $\mathbf{x}_{test}$ at day d. While we do use the missing data index in generating these representations, we observed better performance if we drop that index while scoring. We

Table 1: Summary of Datasets

Dataset	Check-Ins	Users
Gowalla	6,442,892	107,092
Brightkite	4,747,281	51,406
Weeplaces	7,369,712	15,793
Foursquare	33,263,631	266,909

aggregate those scores across all days using mean squared error and sort the users by the score with the lowest score representing users that align better to the patterns of life observed in the unlabeled trajectory.

4 EXPERIMENTS

In this section, we present the results of TUL for our proposed approach for four open source datasets.

4.1 Datasets

We used four open source datasets commonly in benchmarking TUL methodologies [7, 13]: Gowalla[10], Brightkite[4], Weeplaces[9], and Foursquare[16]. Each dataset contains records of user check-ins, defined as the arrival of a user u at location b at time t.

For each dataset, we group sequences of check-ins on a weekly timescale to generate user trajectories $\mathbf{t}_u$. Similar to previous works [13], we filter out any users with less than 5 weekly trajectories and any trajectories composed of less than 10 check-ins.

4.2 Metrics

We evaluate the performance of our models using a sequential 80-20 Train-Test Split of user-trajectories for a 400-user subset of each dataset.

For evaluation, we use commonly used TUL metrics: Accuracy at k (ACC@k), Macro F1 score (Macro-F1), Macro precision (Macro-P), and Macro recall (Macro-R).

ACC@k measures the accuracy of a set of predictions by its top k predicted labels.

$$ACC@k = \frac{\text{\# of correct classifications @ } k}{\text{\# of all observations}}$$

Macro-P and Macro-R are the mean precision and recall among all classes U

$$\text{Macro-P} = \frac{1}{|U|} \sum_{u \in U} \frac{TP_u}{TP_u + FP_u}$$

$$\text{Macro-R} = \frac{1}{|U|} \sum_{u \in U} \frac{TP_u}{TP_u + FN_u}$$

And Macro-F1 is the F1-Score computed using Macro-P and Macro-R

$$\text{Macro-F1} = \frac{2 * \text{Macro-P} * \text{Macro-R}}{\text{Macro-P} + \text{Macro-R}}$$

4.3 Results

We compute metrics outlined in 4.2 for state-of-the-art methods[12, 13] against and our proposed SOTA methodology.

We observe in Table 2 that our approach outperforms alternative models across a majority of evaluation metrics. In particular, we see a significant improvement for metrics that rely on the top predicted user. Our SOTA approach provides, on average, an 11% improvement in ACC@1 performance vs. our Freq classifications and achieves the highest Macro-F1 and Macro-P score for all datasets.

We also note that the reported performance in Table 2 for [13] is calculated after decoupling the mapping of venue and user IDs to avoid data leakage. In testing, when we did not perform this countermeasure, we observed performance metrics for their approach similar to the levels they presented in their original paper.

Similar to previous works, we calculate the performance of TUL at different scales of users in the dataset shown in Figure 1. We observe logarithmic scaling in performance across the user counts available in the data tested. The ability to perform TUL at large scale is crucial for real-world applications where the volume of data and number of users can be immense. With the exception of [13], analyses have remained limited to a few hundred users and have limited applicability as a result. Our scalability opens new possibilities for large-scale personalized services, urban planning, and targeted marketing, making our method a valuable tool for industries relying on extensive spatiotemporal data.

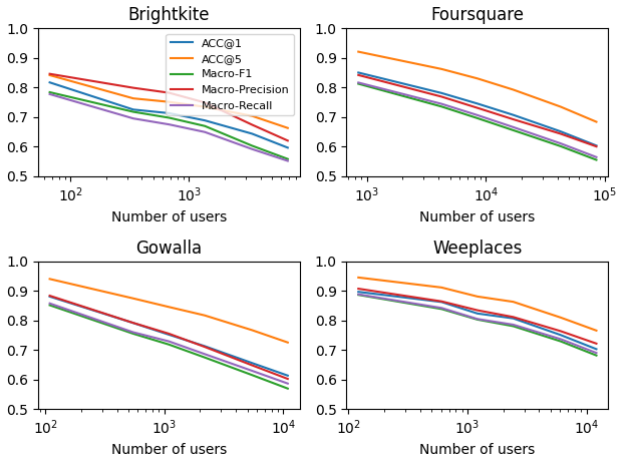

Figure 1: Measured TUL performance using our self-organizing tree algorithm at different numbers of users.

5 CONCLUSION

We introduced a novel approach to user trajectory data using a self-organizing tree algorithm. Our method not only provides human-interpretable intermediate results but also demonstrates superior performance compared to state-of-the-art techniques in Trajectory User Linking (TUL) when restricted to just latitude, longitude, and timestamp data.

Our findings underscore the potential of self-organizing trees to offer a scalable and efficient solution for TUL. This approach opens up new avenues for personalized services and applications across various industries such as urban planning, targeted marketing, and land use.

Moreover, our work lays the groundwork for future research. Potential extensions include incorporating additional data features

Table 2: Evaluation metrics across four public datasets for state-of-the-art approaches and two of our new approaches. We use weekly trajectories satisfying requirements outlined in section 4.1 and restrict the population to 400 randomly selected users with only space and time features. Ours (Freq) represents predictions from the cosine similarity between visit locations while Ours (Tree) represents results from ranking the top 10 results from Ours (Freq) using our self organizing tree approach.

Dataset	Method	ACC@1	ACC@5	Macro-F1	Macro-P	Macro-R
Brightkite[4]	[12]	**0.7463**	**0.8666**	0.6717	0.6932	**0.6956**
	[13]	0.2717	0.2795	0.2686	0.3505	0.2652
	Ours (Freq)	0.6047	0.7473	0.5785	0.6826	0.5492
	Ours (Tree)	0.7128	0.7478	**0.7071**	**0.7891**	0.6829
Foursquare[16]	[12]	0.8371	**0.9397**	0.8050	0.8364	0.8166
	[13]	0.1018	0.1049	0.1117	0.1994	0.0905
	Ours (Freq)	0.8089	0.9350	0.7793	0.8328	0.7690
	Ours (Tree)	**0.8810**	0.9366	**0.8520**	**0.8835**	**0.8531**
Gowalla[10]	[12]	0.7269	0.8665	0.6935	0.7275	0.7082
	[13]	0.0452	0.0467	0.0574	0.1147	0.0456
	Ours (Freq)	0.7210	**0.8922**	0.6933	0.7704	0.6747
	Ours (Tree)	**0.8160**	0.8914	**0.7778**	**0.8113**	**0.7844**
Weeplaces[9]	[12]	0.8074	0.9105	0.7829	0.8083	0.7892
	[13]	0.0658	0.0690	0.0952	0.1964	0.0780
	Ours (Freq)	0.8347	0.9173	0.8174	0.8717	0.8050
	Ours (Tree)	**0.8715**	**0.9205**	**0.8461**	**0.8746**	**0.8471**

to the node representation, such as points of interest, social interactions, and temporal patterns, to further enhance the accuracy and applicability of the model. Exploring hybrid models that combine the strengths of deep learning with self-organizing trees could also yield promising results.

Our self-organizing tree algorithm represents a significant advancement in the field of spatiotemporal data analysis, providing a robust and interpretable framework for trajectory user linking. We believe this approach will inspire further innovations and applications, driving forward spatiotemporal data analytics.

ACKNOWLEDGMENTS

This work was partially funded by IARPA contract 140D0423C0058.

This publication contains general information only and Deloitte is not, by means of this publication, rendering accounting, business, financial, investment, legal, tax, or other professional advice or services. This publication is not a substitute for such professional advice or services, nor should it be used as a basis for any decision or action that may affect your business. Before making any decision or taking any action that may affect your business, you should consult a qualified professional advisor. Deloitte shall not be responsible for any loss sustained by any person who relies on this publication.

REFERENCES

[1] Natalia Adrienko and Gennady Adrienko. 2010. Spatial generalization and aggregation of massive movement data. *IEEE Transactions on visualization and computer graphics* 17, 2 (2010), 205–219.

[2] Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. 2019. Approximating (k, l)-center clustering for curves. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM, 2922–2938.

[3] Lei Chen, M Tamer Özsu, and Vincent Oria. 2005. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*. 491–502.

[4] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.

[5] Joaquín Dopazo and José María Carazo. 1997. Phylogenetic reconstruction using an unsupervised growing neural network that adopts the topology of a phylogenetic tree. *Journal of molecular evolution* 44, 2 (1997), 226–233.

[6] Carlos Andres Ferrero, Luis Otavio Alvares, Willian Zalewski, and Vania Bogorny. 2018. Movelets: Exploring relevant subtrajectories for robust trajectory classification. In *Proceedings of the 33rd Annual ACM symposium on applied computing*. 849–856.

[7] Qiang Gao, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, Xucheng Luo, and Fengli Zhang. 2017. Identifying Human Mobility via Trajectory Embeddings.. In *IJCAI*, Vol. 17. 1689–1695.

[8] Javier Herrero, Alfonso Valencia, and Joaquın Dopazo. 2001. A hierarchical unsupervised growing neural network for clustering gene expression patterns. *Bioinformatics* 17, 2 (2001), 126–136.

[9] Young Liu. 2014. *Weeplaces dataset.* Retrieved Aug 5, 2024 from https://www.yongliu.org/datasets/

[10] Yong Liu, Wei Wei, Aixin Sun, and Chunyan Miao. 2014. Exploiting geographical neighborhood characteristics for location recommendation. In *Proceedings of the 23rd ACM international conference on conference on information and knowledge management*. 739–748.

[11] Feng Luo, Latifur Khan, Farokh Bastani, I-Ling Yen, and Jizhong Zhou. 2004. A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles. *Bioinformatics* 20, 16 (2004), 2605–2617.

[12] Lucas May Petry, Camila Leite Da Silva, Andrea Esuli, Chiara Renso, and Vania Bogorny. 2020. MARC: a robust method for multiple-aspect trajectory classification via space, time, and semantic embeddings. *International Journal of Geographical Information Science* 34, 7 (2020), 1428–1450.

[13] Alameen Najjar and Kyle Mede. 2022. Trajectory-user linking is easier than you think. In *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 4936–4943.

[14] Kevin Toohey and Matt Duckham. 2015. Trajectory similarity measures. *Sigspatial Special* 7, 1 (2015), 43–50.

[15] Michail Vlachos, George Kollios, and Dimitrios Gunopulos. 2002. Discovering similar multidimensional trajectories. In *Proceedings 18th international conference on data engineering*. IEEE, 673–684.

[16] Dingqi Yang, Daqing Zhang, and Bingqing Qu. 2016. Participatory cultural mapping based on collective behavior data in location-based social networks. *ACM Transactions on Intelligent Systems and Technology (TIST)* 7, 3 (2016), 1–23.

[17] Fan Zhou, Qiang Gao, Goce Trajcevski, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2018. Trajectory-User Linking via Variational AutoEncoder.. In *IJCAI*. 3212–3218.