On Privacy in Spatio-Temporal Data: User Identification Using Geotagged Social Media

A thesis submitted in partial fulfillment of the requirements for the degree of
Master of Science at George Mason University

By

Erik Seglem
Bachelor of Arts
American Military University, 2014

Director: Dr. Andreas Züfle, Professor
Department of Geography and GeoInformation Science

Spring Semester 2017
George Mason University
Fairfax, VA
Dedication

I dedicate this thesis to my loving wife, for putting up with all of the time spent on this thesis. Her never ending support was key to its completion.
I would like to thank the following people who made this possible: Andreas Züfle, Jan Stutzki, Felix Borutta, Evgheniy Faerman, and Matthias Schubert.
# Table of Contents

List of Tables ................................................. vi
List of Figures ............................................... vii
Abstract ..................................................... viii

1 Introduction ............................................... 1

2 Related Work .............................................. 5
  2.1 Mobility Patterns ...................................... 5
  2.2 User Identification .................................. 6
  2.3 User Linkage .......................................... 7
  2.4 Spatial Privacy ....................................... 8

3 Problem Definition ......................................... 11

4 Trace based user identification ......................... 14
  4.1 Trace Profile Modeling ............................... 14
     4.1.1 Set Descriptors ................................ 14
     4.1.2 Transition Descriptors ......................... 17
    4.2 Classification ...................................... 19
    4.3 User Linkage ....................................... 20

5 Experimental Evaluation .................................. 21
  5.1 Accuracy using Set Descriptors .................... 24
  5.2 Accuracy using Frequent Transitions ............... 26
  5.3 Accuracy for Different Observation Counts ...... 27

6 Scalability ................................................ 29

7 User Linkage between different Social Networks .... 33

8 Conclusion ................................................ 36
   8.1 Additional Research Opportunities ................ 37

A Metric Accuracies ......................................... 38

B Source Code .............................................. 42

Bibliography ................................................ 43
### List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Grid Sizes and Total Cells</td>
<td>23</td>
</tr>
<tr>
<td>A.1 Accuracy for all Metrics ((k = 1))</td>
<td>39</td>
</tr>
<tr>
<td>A.2 Accuracy for all Metrics ((k = 3))</td>
<td>40</td>
</tr>
<tr>
<td>A.3 Accuracy for all Metrics ((k = 5))</td>
<td>41</td>
</tr>
</tbody>
</table>
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Illustration of Twitter Traces</td>
<td>2</td>
</tr>
<tr>
<td>4.1</td>
<td>Set descriptor trace creation</td>
<td>16</td>
</tr>
<tr>
<td>5.1</td>
<td>Distribution of the Top 500 users in the London-Twitter Dataset.</td>
<td>22</td>
</tr>
<tr>
<td>5.2</td>
<td>Classification Accuracy for varying grid-cell size and varying k.</td>
<td>25</td>
</tr>
<tr>
<td>5.3</td>
<td>Classification Accuracy using Frequent Transitions.</td>
<td>26</td>
</tr>
<tr>
<td>5.4</td>
<td>User identification accuracy for different observation counts.</td>
<td>28</td>
</tr>
<tr>
<td>6.1</td>
<td>Distribution of all 15,989 users in the London-Twitter Dataset.</td>
<td>30</td>
</tr>
<tr>
<td>6.2</td>
<td>Scalability: Scaling the number of Twitter users.</td>
<td>31</td>
</tr>
<tr>
<td>7.1</td>
<td>Classification Accuracy for different Social Networks.</td>
<td>34</td>
</tr>
</tbody>
</table>
Abstract

ON PRIVACY IN SPATIO-TEMPORAL DATA: USER IDENTIFICATION USING GEO-TAGGED SOCIAL MEDIA

Erik Seglem
George Mason University, 2017
Thesis Director: Dr. Andreas Züfle

Location data is among the most sensitive data regarding the privacy of the observed users. To collect location data, mobile phones and other mobile devices constantly track their positions. This work examines the question whether publicly available spatio-temporal user data can be used to link newly observed location data to known user profiles. For this study, publicly available location information about Twitter users is used to construct spatio-temporal user profiles describing a user’s movement in space and time. It shows how to use these profiles to match a new location trace to their user with high accuracy. Furthermore, it shows how to link users of two different trace data sets.

For this case study, 15,989 of the most prolific Twitter users in London in 2014 are considered. The experimental results show that the classification approach allows to correctly identify 98% of the most prolific 500 of these users. Furthermore, it can correctly identify more than 50% of any users by using three observations of these users, rather than their whole location trace. This alarming result shows that spatio-temporal data is highly discriminative, thus putting the privacy of hundreds of millions of geo-social network users at a risk. It further shows that it can correctly match most users of Instagram to users of Twitter.
Chapter 1: Introduction

It is estimated that a third of the 130 billion copies of applications distributed by Apple’s App Store® access a user’s geographic location [1, 2]. As an example, the recently launched augmented reality game “Pokémon Go”, which has been downloaded more than 100 million times on Android devices alone [3], constantly synchronizes the GPS location of users with a company server. While users trust that their location data will be used in sensitive fashion, Apple® recently updated its privacy policy to allow sharing the spatio-temporal location of their users with “partners and licensees”[4].

The mobility behavior of a person often reveals a large variety of sensitive information, which they may not be aware of. A list of potentially sensitive professional and personal information that could be inferred about an individual, knowing only their mobility trace, was published recently by the Electronic Frontier Foundation [5]. Such personal information could simply be marketing information, obtained from a user’s choice of restaurants, or a user’s religious beliefs, inferred through the proximity to a particular church. It can also indicate other, much more sensitive, information about an individual based on their presence in a motel or at a medical clinic.

In this work, the severity of privacy risks through publishing individual spatio-temporal data on the use case of Twitter data is investigated. In particular, it is shown that geotagged tweets might yield enough location information for building user specific trace profiles. Based on these profiles, Twitter accounts can be linked to additional trace data being observed from unknown users. Other location based services or mobile devices are also potential sources for traces. Additionally, face detection methods tag known persons in images in social networks. Thus, geotagged images can reveal a user’s whereabouts at certain points in time. Given that there are multiple such images, it might be possible to build a trace and link it to a known user. To conclude, freely available location data might
Figure 1.1: Illustration of Twitter Traces

(a) Weekly history of a single user.

(b) 12-week history of 10 users.
be used to link accounts and devices for the same user. Thus, the user reveals more of their movements and actions than might be intended.

To derive trace profiles for a given Twitter account, geotagged tweets containing an exact geolocation, a time, and a user ID were collected. Since this work focuses on the location aspect the content of the Tweet is completely ignored, even though it might add even more useful information to user profile. Using the Twitter API, or similar microblogging applications, users can publish a short text message, called a Tweet, together with their current geolocation, a current time-stamp, and their user ID.

The sequence of Tweets of a user is interpreted as a trace. For each user, all available Twitter data is used to build a trace profile to capture each user’s specific mobility patterns. Using these profiles, new trace, for which the originating user is unknown, can be linked to a known user with an alarmingly high accuracy. To illustrate this classification problem, a typical Twitter trace of a single user is depicted in Figure 1.1a. The figure shows a twelve week trace of a user’s tweets, in color-coded one-week intervals. For comparison, Figure 1.1b shows the same twelve week traces for ten users, using a different color per user. Note that the tweets of this user are voluntarily published by the user, such that Figure 1.1a and Figure 1.1b do not raise any privacy concerns.

The challenge of this work is to match a new trace, such as a one week trace corresponding to a single color in Figure 1.1a, to the correct user corresponding to one of the colors in Figure 1.1b. Note that the ten selected user profiles in the example are located in relatively distinct activity regions. Thus, finding the right profile is relatively simple. In a more realistic setting, distinguishing thousands of users in the same area, and user identification is significantly more challenging. In these experiments up to 15,989 users, within the same bounding box of London, are used leading to a much more challenging classification task.

Twitter data is comparatively sparse to other location tracking applications, as tweets are typically published at a frequency of less than one per hour. Despite this data sparsity, it is shown that a large quantity of low-quality location data can still be used to construct highly discriminative user models. To summarize the contributions of this work are as
follows:

• Trace models to capture user-specific movement profiles from sparse traces obtained from Twitter.

• Methods for mapping a newly observed traces of an unknown user to the most likely user in the database.

• An experimental evaluation showing that individual patterns are highly unique and allow for a user classification accuracy of up to 98%.

• A case study of linking users of Twitter to users of Instagram, with an accuracy of up to 81%.

The remainder of this paper is organized as follows. Chapter 2 describes related work of analyzing trace data and user identification. In Chapter 3, problem setting is formalized and the task of linking new trace to users is defined. Chapter 4 describes the trace models and the approach to user identification. The results of the experimental evaluation are described in Chapter 5. Scalability of this solution is address in Chapter 6 and further user linkage experiments are address in Chapter 7. It is concluded in Chapter 8, with additional research opportunities addressed in Chapter 8.1.
Chapter 2: Related Work

This section provides a survey of the state-of-the-art in mobility patterns of individuals, spatio-temporal user identification, user linkage, and spatial privacy. User identification is focused on identifying the same user again within the same database, while user linkage is focused on linking two users together across multiple sources of data. This work assumes that user trace data is fully available, without any notion of privacy preservation. This assumption is appropriate in the experimental evaluation, using publicly available Twitter data. However, other datasets may employ some form of privacy, thus it is important to understand privacy methods, which might be used on this data.

2.1 Mobility Patterns

There are a variety of ways to study mobility patterns of individuals. One source of this data would be travel diaries such as in Wong and Shaw [6]. GPS devices have also been used [7], and more recently cell phone data has been used to study these patterns [8]. Social media, such as Twitter, can also be used. Stefanidis et al. [9] are able to use it to harvest informations about groups. However there are some biases in the data, such as an individual only using social media when they are at certain locations or at certain times of the day it is still usable [10]. This can still provide frequently visited locations. Identifying these major locations provides a good basis for activity pattern analysis such as in Horton and Reynolds [11].

Day to day variability in activity patterns have been shown in Hanson and Huff [12]. Bayarma et al. [13] establish that a single day is not sufficient enough to capture a persons regular activity pattern, but their lives do revolve around a number of major locations. Thus, they are not completely random, and do contain a pattern [14]. If data is collected
only over a day or two, it can be thrown off by random activity, hence a longer period of
collection may be necessary [15]. Hornsby and Cole [16] establish that a coarser granularity
is better at general representation of mobility patterns. Huang and Wong [17] aggregate
longer periods of time, in order to gather mode samples to offset the sparsity of Twitter
data.

2.2 User Identification

A problem similar to the problem of trace based user-identification was considered in
de Montjoye et al. [18]. This work estimates the number of points needed to uniquely
identify an individual trace. The dataset contained 15 months of data on 1.5M users in a
small European country. Each time a user connected to a mobile phone tower to send or
receive a call or text message, a tower location and time, with a resolution of one hour, was
recorded. There are almost 6500 unique antennas in the dataset, and on average each user
has 114 interactions per month. Among this dataset, they found that four randomly chosen
points in a trace were enough to uniquely identify 95% of the trace, and two randomly
chosen points were enough to identify 50% of the traces. However, the question whether a
trace is unique, is different to the problem of user-identification tackled in this work.

The user-identification method in de Montjoye et al. assumes that a trace of the user to
be identified is already in the database. Thus, a new trace, which has not been seen before,
cannot be classified. Summarizing, the work in de Montjoye et al., aims at identifying
individual traces, rather than individual users. Their work provided an initial framework
to build this work on.

The work presented in Bettini et al. [19] investigates the problem of how to prevent
the identification of actual persons behind the users of location based services. Thus de-
anonymizing the user. Therefore, the authors employ so-called location-based quasi identi-
fiers, which are formed from historical spatio-temporal movement patterns that are gathered
from location-based service requests as a potential privacy concern. However, the stated
problem is slightly different from this work, as they make use of external sources to finally
get the real names behind the pseudonames.

2.3 User Linkage

There are a variety of publications considering the problem of user linkage or more general record linkage. In the database community, record linkage generally aims at detecting duplicate records within one or several databases. Records describing the same entity may not share a common key or contain faulty attribute values, which makes the detection of such duplicates non-trivial. A survey on the proposed approaches can be found in Elmagarmid et al. [20].

Considering networks, record linkage is widely understood as user linkage and is stated as the problem of linking corresponding identities from different communities appearing within one or many networks [21]. It is specifically tailored to the requirements of user identification in heterogeneous data by considering co-occurrences adjusted with a stimulus signal. The stimulus signal is derived from locations with frequent co-occurrences and decays with increasing distance to a trajectory. The stimulus signal allows this method to weight important locations, which helps to distinguish two users with very similar trajectories.

An important area of user linkage is social networks where the user linking problem aims at connecting user profiles from different platforms that are used by the same persons. Liu et al. [22] differentiate between three types of user linkage across social networks: user-profile-based methods, which use information provided by the profiles to connect corresponding profiles [23], user-generated-content-based approaches, which analyze the content published by the users to link profiles [22] and user-behavior-model-based methods that generate models based on the (temporal) user behaviors and finally link user profiles based on the similarity of these models [24].

Most related to this approach is the recent work of Cao et al. [25]. In this work, the authors use various sources for data for the trajectories and propose a MapReduce-based framework called Automatic User Identification (AUI). They identify sample rate,
temporal and spatial sparsity, and the fact that people with a close relationship provide similar trajectories as distinct features of the data. Sparsity of the data is corrected by using a long time frame. Signal Based Similarity (SIG) is introduced as a measurement of the similarity of two trajectories. In contrast to that approach, this work uses sparser trajectories. While the authors of Cao et al. do consider sparse social media data, they accumulate these trajectories during a long time interval of at least multiple months. In this work, a long term mobility history of user is not assumed to be available. Instead, it aims at identifying users with the fewest observations possible.

2.4 Spatial Privacy

The predominantly used measurement for privacy is k-anonymity [26], which works with a closed world assumption and assures that, for each query that could be used to identify the identity of a user, at least $k - 1$ other users are returned as possible results.

Common approaches to guarantee a defined degree of anonymity are suppression, obfuscation and generalization [27]. To achieve k-anonymity by suppression, every element that does not fit into an anonymity set is removed [28, 29]. For trajectories, suppression would require discarding observations in discriminative locations such as a user’s home. While this method is effective, the use of suppression alone can lead to a significant loss of information. Perturbation is another method used to obfuscate the data [30]. The goal is to generate a synthetic dataset with the same properties of the original dataset using a generative model. For generalization, $k$-groups of users could simply be unified into a single entity.

This work does not try to maintain privacy of users, and can be seen as an adversary approach of trying to breach the privacy of users. A highly relevant future piece of work is to investigate how existing privacy preservation methods for trajectories can be employed to suppress, obfuscate and generalize trajectories to minimize the user identification accuracy of this solutions, while further minimizing the loss of information in the data.

A more refined version of k-anonymity is l-diversity, which addresses some shortcomings of k-anonymity [31], mainly where properties of the data are homogeneous and allow
conclusions, which might violate the assured k-anonymity. Regarding trajectories, location l-diversity is required as introduced in Beresford and Stajano [32]. As an enhancement of l-diversity, t-closeness [33] is used on datasets where the distribution of attribute values allows conclusions to identities.

These measurements are typically applied when medical records are published or in regards to Location Based Services (LBS), which require personalized location information. As LBS are usually working with GPS coordinates and trajectories, the raw data is similar to the information used in this work. But there is a difference in quality and frequency. LBS usually work with the assumption that a user is willingly providing their location as precise as possible and/or performing measurements of the location with a high frequency. While work has been done on interpolating real trajectories from purposefully obfuscated ones [34], the data used is limited to one service and focusing on the k of k-anonymity instead on user identification.

The work of Abul et al. [35] applies k-anonymity on spatio-temporal objects introducing the \((k,\delta)\)-anonymity. The trajectories of a user are extended by the uncertainty of the location measurement \(\delta\). The authors claim that a series of trajectories and locations can be modeled as a series of cylinders, or a tube. k-anonymity is granted when \(k-1\) additional elements of the set can fit into a tube. The proposed method uses outlier detection and other forms of suppression in combination with space transformation of a maximum of \(\delta/2\) while \(\delta\) defining the circumference of the tube remains unchanged. The paper proposes a heuristic that succeeds to find anonymity sets as the problem is NP-hard.

The notion of \((k,\delta)\)-anonymity is also discussed in Trujillo-Rasua and Domingo-Ferrer [36]. The authors come to the conclusion that existing methods to create \((k,\delta)\)-anonymity as developed in Abul et al. are not sufficient if \(\delta > 0\). By defining every location in a spatio-temporal trajectory as a quasi-identifier and assuming that a potential adversary has knowledge about one sub trajectory they show that the probability to correctly identify a series of trajectories is larger than \(1/k\) thus violating the \(k\)-anonymity. This work will show that it is indeed possible to identify users with high probability by only knowing a
sub trajectory.
Chapter 3: Problem Definition

In this work, the question of to what extent a set of spatio-temporal observations, such as geotagged tweets, are sufficient to derive spatial user profiles for the observed users and reliably link location traces of unknown users to one of the known user profiles is answered. Therefore, this section will define terms and notations, and formally define the problem of user identification using trace data.

In this paper, spatio-temporal data is considered. That is data of users annotated with a geolocation and a timestamp, such as obtained from Twitter.

**Definition 1** (Spatio-Temporal Database). Let $U$ denote a set of unique user identifiers, let $S$ be a set of spatial regions, and let $T$ denote a time domain. A spatio-temporal database $D \subseteq U \times S \times T$ is a collection of triples $(id \in U, s \in S, t \in T)$. Each triple $(u, s, t) \in D$ is called an observation.

Furthermore, a trajectory is defined as a sequence of location and time pairs.

**Definition 2** (Trajectory). A trajectory $tr \subseteq S \times T$ is a collection of pairs $(s \in S, t \in T)$.

These trajectories are then partitioned temporally and spatially to distill the information down to a minimum set of components.

To build a user specific mobility pattern, the data is temporally partitioned into equal sized time intervals called epochs. Within an epoch the set of observations of a specific user is called a location trace, formally defined as follows.

**Definition 3** (Location Trace). Let $D$ be a spatio-temporal database. Let $E = \{e_1, ... e_n\}$ be a partitioning of $T$ into $n$ temporal intervals denoted as epochs. For each epoch $e \in E$, and each user $u$, the trace

$$D(u, e) := \{(u', s, t) \in D | u' = u, t \in e\},$$

(3.1)
is called the location trace of user id during epoch \( e \).

In the remainder of this paper, location trace models are introduced to capture the motion of a user in space and time. The models are derived from the set of all trace of a user, called their trace profile formally defined as follows.

**Definition 4 (Trace Profile).** Let \( D \) be a spatio-temporal database, let \( u \in U \) be a user and let \( E \) be a temporal partitioning of \( D \) into \( n \) epochs. The trace profile \( \mathcal{P}(u) \) is the set of all traces of \( u \), i.e.,

\[
\mathcal{P}(u) = \{ D(u', e) | u' = u, e \in E \}.
\]

(3.2)

This trace profile is used to establish a pattern between multiple traces over discrete epochs, in order to allow for more accurate user identification.

The main challenge of this paper is to map the trace of an unknown user to a user already in the database, thus identifying them.

**Definition 5 (User Identification).** Let \( D \) be a spatio-temporal database and let \( Q \subseteq S \times T \) be a trace of an unknown user \( u \). The task of user identification is to predict the identity of user \( u \) of \( Q \) given \( D \). The function

\[
I : \mathcal{P}(S \times T) \mapsto U,
\]

(3.3)

maps a trace \( Q \) to user \( x \) as a user identification function.

Thus, user identification is a classification task mapping a trace to its unknown user. This is not to be confused with a de-anonymization attack, such as in Bettini et al. where a user’s real world identity is uncovered. This task is only attempting to identify which user’s Trace Profile is most similar to the new trace. To train a user identification function \( I(Q) \), the next chapter presents the classification approach, which uses the traces in \( D \) as a training set, in order to predict the user of a new trace \( Q \not\in D \).

User linkage, takes this task further and attempts to map two users, from separate databases, together and is formally defined as follows.
**Definition 6 (User Linkage).** Let $\mathcal{D}_1$ and $\mathcal{D}_2$ be two trace feature databases, and let $\mathcal{U}_{\mathcal{D}_1}$ and $\mathcal{U}_{\mathcal{D}_1}$ be two user databases. Such that each entry $(T, u) \in \mathcal{D}_1$ corresponds to a trace feature vector $T$ and a user $u \in \mathcal{U}_{\mathcal{D}_1}$, and each $(T', u') \in \mathcal{D}_2$ corresponds to a trace feature vector $T'$ and a user $u' \in \mathcal{U}_{\mathcal{D}_2}$. The task of user linkage is to map a user in $\mathcal{U}_{\mathcal{D}_1}$ to a user in $\mathcal{U}_{\mathcal{D}_2}$.

This is achieved by using the user identification techniques discussed in the next chapter and will be discussed in Chapter 7.
Chapter 4: Trace based user identification

Trace models are introduced to capture the motion of a user \( u \in U \) in space and time by learning from their trace profile \( P(u) \) in Subsection 4.1. Note that this first approach does not consider the time component of observations of a user within an epoch. The time component is only used to divide the whole trajectory of a user into different epochs that can be used for learning and testing. For each model, a similarity measure to quantify similarity between different trace models is proposed. Based on these similarity measures, the user identification approach is presented in Subsection 4.2. As mentioned before, the prediction is based on the assumption that there exists a profile \( P(u_i) \) for each user \( u_i \in U \).

4.1 Trace Profile Modeling

Each trace \( D(u, e) \) of user \( u \) during epoch \( e \) is a sequence of observations, i.e., time-stamped geo-locations. A spatial grid to partition geo-space into equal sized regions \( S = \{ S_1, S_{|S|} \} \) is used, thus reducing a trace to a sequence of time-stamped grid-cells. To model such a sequence, two kinds of approaches are proposed:

- The first approach using set descriptors treats a trace as a set of grid-cell observations, thus ignoring the sequence, ordering, and time-stamps of these observations.
- The second approach using frequent transitions considers the transitions of users from one spatial region to another, thus explicitly modeling the order of observations.

4.1.1 Set Descriptors

Ignoring the temporal aspect, a trace \( D(u, e) \) of user \( u \) during epoch \( e \) can be described by a vector \( v(u, e) \) of all spatial regions in \( S \). In other words, each spatial region is represented
by a dimension of \( v(u, e) \).

Note that \( v(u, e) \) contains zero values in the majority of dimensions as each user usually only traverses a small fraction of space during an epoch. In other words, \( v(u, e) \) is sparse. Modeling trace using frequency descriptions has a strong resemblance to handling bag of words vectors known in text mining. To describe, if and how often a domain was visited within trace \( D(u, e) \), the following two approaches are examined.

Simple examples of these can be seen in Figure 4.1. Each color represents a different user geotagging in the area of interest. Then a simple grid was applied to the area, and set descriptors were generated from the data.

**Binary Descriptor** In this rather simple method, a trace \( D(u, e) \) is represented as a set of visited spatial regions. Thus, each feature value \( v_{\text{bit}} \) equals one if user \( u \) visited region \( S_i \) (at least once) during epoch \( e \), formally:

\[
v_{\text{bit}}^i(u, e) := \begin{cases} 
1, & \text{if } \exists (u', s, t) \in D : u' = u \land s \in S_i \land t \in e, \\
0, & \text{otherwise}
\end{cases}
\] (4.1)

A visual example of this is shown in 4.1a.

To compare binary vectors \( v, v' \in \{0, 1\}^n \), the Jaccard coefficient is employed [37], which is a standard similarity measure for sets:

**Definition 7 (Jaccard Coefficient).** Let \( v, v' \in \{0, 1\}^n \) be two bit vectors, then the Jaccard coefficient is defined as follows:

\[
\text{Jac}(v, v') = \frac{\sum_{i=1}^{n} v_i \wedge v'_i}{\sum_{i=1}^{n} v_i \vee v'_i}
\] (4.2)

**Frequency Descriptors** A frequency, or term weighted, vector [38] \( v^{\text{freq}} \) contains the number of visits of each spatial region of user \( u \) in epoch \( e \). This allows to distinguish
Figure 4.1: Set descriptor trace creation.
between users visiting a particular region more or less often than other users.

\[
v^{\text{freq}}(u,e)_i = |\{(u',s,t) \in D | u' = u \land s \in S_i \land t \in e\}|. \quad (4.3)
\]

A visual example of this is shown in 4.1b.

A common way to compute the similarity in sparse numerical vectors is the cosine coefficient:

**Definition 8 (Cosine Coefficient).** Let \( v, v' \in \mathbb{N}^n \) be two vectors, then the Cosine coefficient is defined as follows:

\[
\text{Cos}(v, v') = \frac{v \cdot v'}{||v|| \cdot ||v'||} \quad (4.4)
\]

Since the cosine coefficient can be strongly dominated by dimensions having high average frequency values, spatial regions are normalized by their total number of observations [38].

### 4.1.2 Transition Descriptors

All of the previous trace descriptors had in common that they treat a trace as an unordered set of locations, without considering any notion of sequence or time. In this section, a trace is treated as a sequence of regions. As a baseline to compute the similarity between two sequences, dynamic time-warping [39] (DTW), a state-of-the-art method for similarity search on sequences, is used. Since the experimental evaluation shows that using DTW without any adaption as a similarity measure yields a fairly low classification accuracy, this section presents two approaches to directly model the transitions of a trace. A transition is a pair \((s, s')\) of regions where \(s\) is called source and \(s'\) is called destination. Using a descriptor for each pair of spatial regions \(s_i, s_j\), describing the number of times the specific sequence \((s_i, s_j)\) has been observed in a trace \(D(u,e)\), is proposed.

**Definition 9 (Trace Transitions).** Let \(D(u,e) = \{(s_1,t_1),\ldots,(s_n,t_n)\}\) be a trace, the set of
transitions \( \uparrow \mathcal{D}(u, e) \) is defined as the multi-set (thus allowing duplicates)

\[
\uparrow \mathcal{D}(u, e) := \bigvee_{1 \leq i < n} (s_i, s_{i+1}). \tag{4.5}
\]

The number of occurrences of \((s, s')\) in trace \( \mathcal{D}(s, e) \) is denoted as \( \uparrow \mathcal{D}(u, e)(s, s') \).

Since modeling all observed transitions blows up the feature space quadratically, using only the \( k \) globally most frequent transitions as features is proposed.

- **Frequent Transitions:** The globally most frequent transitions are searched for and the number of occurrences of these transitions is used as a feature vector to describe a trace.

- **Transition Probabilities:** Common transitions of two traces are found, and their similarities are adapted by the global rarity of these transitions.

**Definition 10** (Top-\( k \) Most Frequent Transitions). Let \( k \) be a positive integer, then the set \( \text{FT} \) is a set of pairs of spatial regions defined as

\[
\text{FT}^k(\mathcal{D}) = \arg\max_{s_i, s_j \in \mathcal{S}} \{ \sum_{u \in \mathcal{U}, e \in \mathcal{E}} \uparrow \mathcal{D}(u, e)(s_i, s_j) \}, \tag{4.6}
\]

where \( \arg\max_X(\varphi) \) returns the set of \( k \) arguments \( x \in X \) yielding the maximum value substituted in term \( \varphi \).

Now the \( k \) most frequent transitions \( \text{FT}^k(\mathcal{D}) \) can be used as additional features. Similar to the set descriptors presented in Subsection 4.1.1, the features are described using

- **Bit vectors,** using the feature vector

\[
v_i^{\uparrow \text{bit}(u, e)} = \begin{cases} 
1 & \text{if } \text{FT}^k(\mathcal{D})_i \in \uparrow \mathcal{D}(u, e) \\
0 & \text{otherwise}
\end{cases} \tag{4.7}
\]
• Frequency vectors, using the binary feature vector

\[ v^{\text{freq}}(u, e)_i = \uparrow D(u, e)(FT^k(D)_i) \]  

(4.8)

For these vectors, the same similarity functions defined in Section 4.1.1 can be used.

4.2 Classification

Regardless of which of the modeling approaches presented in this section is employed, the result is a high-dimensional feature vector. To classify a new trace of an unknown user, the next section proposes the classification procedure, using the previously proposed user-specific trace models. To classify the user of a new trace, a \(k\)-nearest neighbor classification approach is employed. This choice is made due to the extremely high dimensional feature space, having one dimension per spatial grid-cell. Therefore, given a trace database \(D\), traces \(D(u, e)\) are extracted for each user \(u\) in each epoch \(e\). Since the user is known for each of these traces, the result is a labeled dataset \(P_{train}\) of feature vectors. Given a new trace \(Q\), map \(Q\) to its feature description \(v_{new}\) and search the \(k\)-nearest neighbors of \(v_{new}\) in \(P_{train}\) w.r.t. a corresponding similarity measure. To decide the final class decision, each queried neighbor is weighted by its similarity value and the class is predicted as the one having the largest cumulated similarity.

Formally, the \(k\)-nearest neighbors classification can be defined as follows. Let \(P_{train} = \{(v_i, y_i) \mid v_i \in \{0, 1\}^n \wedge y_i \in \mathcal{L}\}\) be the set of training instances consisting of pairs \((v_i, y_i)\) with \(v_i\) being the feature description of the user trace \(i\) and \(y_i\) being the label, i.e., identity of the user, assigned to trace \(i\). \(\mathcal{L}\) denotes the set of labels. Given the feature description \(v_{new}\) of a query trace, the identity, resp. label, \(y_{new}\) of \(v_{new}\) is determined by cumulating the similarities, i.e., \(d(., .)\), for each label \(l \in \mathcal{L}\) represented among the \(k\)-nearest neighbors of \(v_{new}\) and taking the most representative label.

\[ y_{new} = \arg\max_{l \in \mathcal{L}} \{ \sum d(v_{new}, v_k^l) \mid v_k^l \in k\text{NN}(v_{new}) \} \]  

(4.9)
Note that no index structure is used to support the kNN-search due to the high dimensionality of the feature space.

4.3 User Linkage

In addition to the identification of individual users, another application of the user trace profiling is to link users between two trace datasets. Therefore, let \( D \) and \( D' \) be two trace databases having the set of users \( \mathcal{U} \) and \( \mathcal{U}' \), respectively. The task of user linkage is to find pairs of database users \((u \in \mathcal{U}, u' \in \mathcal{U}')\) that correspond to the same individual in the real world, i.e., having \( u = u' \). As an example, the two datasets may correspond to Twitter and Instagram. The same individual may have different user names in both social networks. The task of user linkage is to find such individuals.

Clearly, using the approach presented in Section 4.2, the trace of each user are classified in \( D \), and the most similar user in \( D' \) is classified. The drawback of such approach is that multiple users in \( D \) may be matched to the same user in \( D' \), and some users in \( D' \) might not have any match. To avoid this drawback, the matching problem is formalized as a bipartite graph, containing for each \((u \in \mathcal{U}, u' \in \mathcal{U}')\) a weight of similarity. This similarity is chosen by performing a kNN search of each trace in \( D \) on the database \( D' \). Then, the score of \((u, u')\) corresponds to the number of occurrences of \( u' \) in kNN sets of all traces of user \( u \).

Given this bipartite graph, the Hopcroft-Karp algorithm [40] is used to find an optimal matching, i.e., mapping of each user in the smaller database to exactly one user in the other that maximizes the total score.
Chapter 5: Experimental Evaluation

The proposed approach is initially evaluated on a dataset mined from Twitter using their public API, feeding from a global 1%-sample using a $(51.25, 51.75)$ degrees longitude to $(-0.55, 0.30)$ degrees latitude window covering the London region shown in Figure 1.1b. London was chosen as a starting location for having a high population, while still being predominantly an English speaking location. Furthermore, London shows a high density of users and tweets, thus increasing the size of the trace database $\mathcal{D}$, and allowing more significant conclusions to be made.

It was also decided to use one-week periods for the temporal epoch. This choice was meant to minimize the daily variability in a user's locations. For example: a user may work Monday through Friday, but only go to the gym Monday, Wednesday and Friday, or night classes on Tuesday and Thursday. They may also go grocery shopping, or to a religious institution only on the weekends, thus going locations significantly different than were they would be on a week day. Additionally, Twitter data is extremely sparse for most users, thus more than a day was necessary to a reasonable number of tweets to create location traces from. This lead to the choice of a twelve week time interval from December 30, 2013 to March 24, 2014 being used. The choice of twelve weeks was to allow for multiple weeks of traces for each user, while not having too large of a dataset to use for initial testing purposes.

Out of these London-Tweets, the 500 users with the most Tweets during the study period were selected, excluding obvious spammer or bot users. This dataset was then split into temporal epochs of one-week. Thus, the database contains a total of $|\mathcal{U}| = 500$ users, and a total of $|\mathcal{E}| = 12$ epochs. Consequently, the database $\mathcal{D}$ contains a total of $\mathcal{U} \times \mathcal{E} = 6000$ location traces.
Figure 5.1: Distribution of the Top 500 users in the London-Twitter Dataset.

(a) Traces per user within the 12 epochs.

(b) Locations per one-week trace.
To discretize space, a spatial grid is applied on the aforementioned rectangle covering the London region, having an extent $\text{ext}$ in longitude and latitude ranging from 0.01' to 0.001'. The set of all resulting grid cells constitutes the set of spatial regions $\mathcal{S}$, having $|\mathcal{S}| = 4,250$ cells for $\text{ext} = 0.01'$ and 425,000 cells for $\text{ext} = 0.001'$. Table 5.1 shows all grid sizes as well as the dimensions of the related traces.

Consequently, for a user $u \in \mathcal{U}$ and an epoch $e \in \mathcal{E}$ a trace $\mathcal{D}(u, e)$ is a sequence of cells in $\mathcal{S}$. To give a more detailed intuition of the characteristics of the dataset, Figure 5.1 shows statistics about the traces of these 500 users. Figure 5.1a shows the number of traces having at least one observation in the corresponding epoch.

Of users, 42% have an observation have at least one observation in each of the twelve epochs, and 75% of the users have at least one observation in at least eight epochs. In addition, Figure 5.1b shows the number of observed cells for each trace. Most users only visited a small number of space cells each week, as half of the trace contain six or less cells. Note that any trace having zero observations were removed from the dataset.

The classification experiments in this work were performed using an eight-fold cross validation. Eight folds for optimal parallelization on an eight core processor. Thus, in each experiment a test set of trace $\mathcal{Q}(u, e) \subset \mathcal{D}(u, e)$ is selected, and user mobility profiles are built using the techniques of Section 4.1, without using the test traces, i.e.

<table>
<thead>
<tr>
<th>Degrees</th>
<th>X Size</th>
<th>Y Size</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.010</td>
<td>85</td>
<td>50</td>
<td>4,250</td>
</tr>
<tr>
<td>0.009</td>
<td>95</td>
<td>56</td>
<td>5,320</td>
</tr>
<tr>
<td>0.008</td>
<td>107</td>
<td>63</td>
<td>6,741</td>
</tr>
<tr>
<td>0.007</td>
<td>122</td>
<td>72</td>
<td>8,784</td>
</tr>
<tr>
<td>0.006</td>
<td>142</td>
<td>84</td>
<td>11,928</td>
</tr>
<tr>
<td>0.005</td>
<td>170</td>
<td>100</td>
<td>17,000</td>
</tr>
<tr>
<td>0.004</td>
<td>213</td>
<td>125</td>
<td>26,625</td>
</tr>
<tr>
<td>0.003</td>
<td>284</td>
<td>167</td>
<td>47,428</td>
</tr>
<tr>
<td>0.002</td>
<td>425</td>
<td>250</td>
<td>106,250</td>
</tr>
<tr>
<td>0.001</td>
<td>850</td>
<td>500</td>
<td>425,000</td>
</tr>
</tbody>
</table>
Note that this important avoidance of over-fitting is a main differentiation to the trace identification approach proposed in de Montjoye et al. [18]. By having the query trace in the training data, a $k = 1$-NN classification would always return a 100% classification accuracy, but defeating the purpose of user identification. Consequently, since the related work in de Montjoye et al., solves a different problem, a comparison would be unfair and non-explanatory. See Section 2 for more details on de Montjoye et al..

As a classifier, $k$-nearest neighbor classification was utilized, using a distance-weighting in case of ties, which is able to perform well despite an extremely large number of $|S|$ features. Classifications are performed using scikit-learn, a Python machine learning framework [41]. An exhaustive search of all combinations available in scikit-learn in order to determine the best possible settings to use. See Appendix A for raw results.

5.1 Accuracy using Set Descriptors

In the first set of experiments, the accuracy of the user identification is evaluated for different grid-resolutions $ext$, using binary descriptors for the Jaccard similarity measure (c.f. Definition 4.1.1). The results of this evaluation are shown in Figure 5.2a. In the basic setting having a relatively coarse spatial grid of $ext = 0.01'$, a simple distance weighted $k$NN classification is able to correctly identify (c.f. Definition 5) up to 85% of individuals for $k = 5$. This result improves even further as the grid-resolution $ext$ is increased. In the case of the most detailed grid having $ext = 0.001'$, the solution is able to break the 97% classification accuracy line. This result is quite concerning, as it shows that the motion of individual real-persons is quite characteristic, and that the motion model allows to capture this individuality and allows to discriminate different users very well.

The classification result are worse for $k = 1$ and $k = 3$. This result is contributed to chance, as another user may, by chance, have a trace very similar to the query trace $q_{u}^e \in Q(u, e)$ of user $u$. However, by using more neighbors, it is likely that the correct user
Figure 5.2: Classification Accuracy for varying grid-cell size and varying k.
u appears at least twice in the $k = 3$ or $k = 5$ set, thus out-weighting the erroneous user in the first rank. Yet, for $k > 5$ there is a drop in accuracy. This is contributed that the query user only has at most 11 traces in the training set. This number might be less than 11 if a user was not active in all epochs. This is the case for many users, shown by Figure 5.1. In the extreme case having $k = 21$, at least 10 trace of wrong users must be in the $k$NN result, allowing noise have a much greater effect, especially in the case where $u$ has few trace.

Furthermore, Figure 5.2b shows the results using frequency vectors as descriptors, and using the cosine coefficient as a similarity measure (c.f. Definition 4.1.1). The improvement in classification accuracy is relatively minor, but are able to hit the 98% accuracy mark. This result can be contributed to the fact that binary descriptors already perform so well. Summarizing, knowing the set of places that a user visited is descriptive enough, such that the frequency of visits does not yield much additional descriptiveness.

## 5.2 Accuracy using Frequent Transitions

In the next set of experiments, how the usage of transition descriptors (c.f. Section 4.1.2) instead of set descriptors affects the classification accuracy is evaluated. The results depicted
in Figure 5.3 indicate that using from-to-transitions, as opposed to just using sets of cells, further allows to improve the classification quality. An increase in classification accuracy of around 10% (absolute) is observed using transitions, achieving an classification accuracy of nearly 95%. This result indicates that the sequence, and thus the motion in space and time is more descriptive than just sets of regions, and thus the motion in space-only.

While this was method did allow for a slight increase in accuracy, this increase came at a cost. It causes the dimensionality of the data, and greatly increase complexity. And thus, it greatly increases the processing requirements. Because there was only a small increase in accuracy for this increase in complexity, transitions were not use used in the remainder of the experiments. Though, further research in the subject could be worthwhile.

5.3 Accuracy for Different Observation Counts

Next, the number of observations required to identify (c.f. Definition 5) a user accurately is evaluated. Therefore counts are created according to the observation distribution in Figure 5.1. Then tests are for each count. If a trace does not have the minimum number of observations for the corresponding group, it is not tested, and if a trace has more observations than the allowed maximum for the corresponding group, a random sample is taken and tested instead. Thus, instead of testing the accuracy on the original traces this tests the accuracy on controlled observation counts.

The classification results for each group can be seen in Figure 5.4. Surprisingly, in the case of having only one random observation for each trace, it is possible to identify over 70% of the users in this dataset. This is likely due to the fact that a random location from a trace is likely to pick a users most frequent grid cell, which is most discriminative. Increasing the number of observation samples to two, a significant increase in accuracy to 78% is seen, and a steady growth in accuracy from there is shown. Accuracy starts to level off after having 30 or more observations from a user. This is surprising, as the vast majority of trace has more than 60 observations. Thus, sampling down to 30 observations, yields a significant reduction in data, but as Figure 5.4 shows, yields almost no reduction in accuracy.
of discriminative information.

The leveled accuracy level is above 90%, which is extremely high for a classification task having 500 different classes. This positive result is also a consequence of large trace (i.e., traces having a large number of observations) generally having larger trace in the training set, as the frequency distribution of tweets among these 500 Twitter users in London is very skewed. Finally, the classification performs the best, if the parameter of the \( kNN \) classification is set to \( k = 1 \). This result is in line with Figure 5.2b, as Cosine-Similarity is used per default in this experiment.

Summarizing this experiment, very short trace having 10 or less observations in space and time are enough to unveil the identity of a user. This is a concerning result.
Chapter 6: Scalability

In all of the previous experiments, only the top 500 Twitter users in London were used. In the final experiment, this number of users is scaled up, by using 15,989 users that have at least two trace containing at least two observations each. This larger dataset contains over 2.7 million Tweets, including the original dataset. Statistics for this dataset are shown in Figure 6.1. The quality of the observed traces is much worse compared to the earlier 500 users explored in Figure 5.1: In Figure 6.1a more than half of the users have less than five traces within the twelve epochs, and only a small fraction of 6% of the users have maximum number of twelve traces. In addition, Figure 6.1b shows the quality of these trace is much lower, as nearly 50% of the traces have three or less observations. Due to the quality of this data a eight-fold split was no longer possible. A stratified shuffle split was used instead, taking 10 iterations of 20% samples.

The results on this dataset, in terms of classification accuracy as well as run-times are shown in Figure 6.2. In terms of accuracy, there is a vast decrease in accuracy observed, even for the default setting of 500 users. This is because the experiments are no longer using the top users, but just a random sample of users, and the data quality, in terms of number of observations per trace, as well as the number of trace per user, is much lower for these users.

Clearly, less frequent users are harder to classify, since there is less information. As the experiments are scaled up the number of users, there is a decrease in classification accuracy, as the classification problem becomes harder having more users. Still, the classification accuracy remains at almost 50%, despite the large number of 15,989 users, and the much lower trace quality.

Since a $k$NN classification is employed, and thus a lazy learning method is used, there is no model learning phase. The run-time results for the classification is shown in Figure
(a) Traces per user.

(b) Observations per one-week trace.

Figure 6.1: Distribution of all 15,989 users in the London-Twitter Dataset.
Figure 6.2: Scalability: Scaling the number of Twitter users.
6.2b. a linear run-time is observed, which is attributed to the extreme high dimensionality of the feature vectors, which cannot be beneficially supported by an index structure for the kNN search. But even at the full 15,989 users, the time to classify each trace is less than 1ms.

\footnote{Run-time tests were performed on AWS using a m4.2xlarge EC2 instance running Amazon Linux. This instance type has 8 CPU cores and 32GB of RAM.}
Chapter 7: User Linkage between different Social Networks

In all the previous experiments, a single user had to be identified based on a new trace. In this section, the next step is evaluated. Linking whole sets of users of two different social networks, based on their traces, as described in Section 4.3 and defined in Definition 6. For this purpose, two new datasets are employed, one generated synthetically by splitting the scalability (c.f. Chapter 6) dataset randomly, and one splitting the same dataset based on links between Twitter and Instagram.

**Synthetic Database Split:** For the synthetic database, a fraction of $p$ Tweets is uniformly sampled from the Twitter dataset $\mathcal{D}$, and pretend that this set belongs to a different social network $\mathcal{D}'$. In this sampled database $\mathcal{D}'$, the user-labels as ground-truth, which the algorithm tries to predict given the data in $\mathcal{D}$ can be used. For this experiment, only traces having at least 10 tweets to sample from are considered. If uniform sampling of a trace yields an empty set, it is re-sampled.

**Instagram Data:** Out of the 2.7 million tweets in the dataset, a significant portion of 204 thousand tweets is labelled as coming from the Instagram network. These Tweets were cross posted by the user, on both Instagram and Twitter. Thus, the Instagram database $\mathcal{D}^I$ consists of all these cross-linked posts. For the Twitter database, two cases are evaluated. In the first case, the full dataset $\mathcal{D}$ can simply be used, thus assuming that the Instagram observations were made in both datasets. In the second case, the database $\mathcal{D}^T = \mathcal{D} \setminus \mathcal{D}^I$ is used, thus assuming that the Instagram observations were made in the Instagram network only.

The results on the synthetic database split are shown in Figure 7.1a. For each value of $p$, 10 random samples of the database $\mathcal{D}$ are obtained, and results from each are averaged in order to avoid effects generated due to random sampled. In all ten runs, the depicted values showed almost no deviation, all being in a $\pm 0.5\%$ interval. An even 50/50 split yields
(a) User Linkage results for different fractions of user belonging to each database.

(b) User Linkage results for linking Twitter and Instagram.

Figure 7.1: Classification Accuracy for different Social Networks.
a correct linkage rate of almost 85%. Yet, this split becomes biased towards a smaller value $p$. This can be explained by having a larger sample in the training database $\mathcal{D}$, on which the traces of $\mathcal{D}'$ are queried on. However, for $p = 0.1$, this accuracy drops significantly. This can be explained by the previous experiments, showing that a sample of as little as three observations suffices for a high classification accuracy. However, since many of the traces only have $10 - 20$ observations, there is a high chance that a 10% random sample may only have one or two observations.

For the Instagram-Twitter matching, the results are shown in Figure 7.1b, for the two cases of using the data as is, thus having all Instagram observations also present in the Twitter database, and the case of splitting the dataset, thus removing the Instagram observations from the Twitter traces. Using the raw dataset a prediction accuracy of roughly 80% using $k = 1$ nearest neighbor classification to build the bi-partite graph is observed.

In contrast, the case of splitting Instagram off of Twitter, the accuracy drops to about 10%. These disappointing results can be explained by making the hypothesis that users use Instagram and Twitter in different ways, such as using Instagram when on a far-away vacation, while also using Twitter in locations where you don’t usually take a picture, such as work and home. Also, some of the users had all their tweets linked to Instagram, such that the algorithm had no training data left in the Twitter database, thus having to random guess the user. Thus, it appears that Twitter and Instagram are used differently by users, making the Instagram sample much harder to match than a uniform random sample taken from Twitter.
Chapter 8: Conclusion

In this work, the challenge of identifying users in a spatio-temporal database was approached. This approach uses historic traces of a user to learn their motion in space and time, by proposing various feature extraction and similarity search methods. Using a 12-week dataset of Tweets in the London region, the experimental results show that it is possible to map a trace to a ground-truth user with extremely high accuracy.

This raises various concerns and opportunities:

- **The Threat** of loss of privacy: Traces of real people are publicly available. Given only few observations of an individual. For example, one person inadvertently appearing in the background of another person’s Facebook images, then identifying this user in a trace database, and linking them to additional data, such as username or real name.

- **The Potential** through record linkage of trace databases. For example, joining the personal interests in locations (such as restaurants, bars, cafes) from a LBSN with textual thoughts of a user from a micro-blog. Thus, a micro-blog tweet from user $u$ such as “**W00T, I’m going to George Mason University!**”, might be used to recommend restaurants to $u$ by mining their restaurant preferences using their check-ins in the LBSN.

- **The Challenge** of privacy preservation by trace obfuscation and other means. By learning the characteristics that make a trace matchable to its user, techniques to hide particularly descriptive and discriminative observations from the public trace can be developed.
8.1 Additional Research Opportunities

There are many additional research opportunities that could follow on this work, in a wide area of topics. Some of those areas might include:

- Utilizing this research with addition sources of traces to further validate the work in it. A prime target for this would be using twitter and Instagram data to link users across multiple platforms.

- Additional research into frequent transitions. This research showed the sequence of locations a user visits can add additional uniqueness, allowing users to be identified more easily, however there is likely room for improvement. The transitions added additions dimensionality to an problem, which was already extremely dimensional, and was not a primary focus of this research.

- Dimension reduction, or some form of principle component analysis, could provide significant improvements to this research. The dimensionality of the data will likely cause issues when scaling up to a larger area of interest, and could become a hindrance in very large areas.

- A tiered approach could be useful for larger areas of interest, to include world wide analysis. One might be able to use a very coarse grid at a world scale, in order to group users together into smaller areas. Then they could keep drilling down into smaller and smaller areas. At each level, there may be a set of unique users, which are identifiable, and the rest can be drilled down on at a finer scale.
Appendix A: Metric Accuracies

The following tables include the accuracies for all metrics attempted, on the 500 user dataset, in order to discern which would be the best to start with in the experiments of this work. This was an exhaustive search of all possible combinations offered within scikit-learn [41].
Table A.1: Accuracy for all Metrics (k = 1)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Index</th>
<th>Accuracy</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>euclidean</td>
<td>kd tree</td>
<td>0.838188</td>
<td>0.044796</td>
</tr>
<tr>
<td>manhattan</td>
<td>kd tree</td>
<td>0.826809</td>
<td>0.057972</td>
</tr>
<tr>
<td>chebyshev</td>
<td>kd tree</td>
<td>0.820695</td>
<td>0.039398</td>
</tr>
<tr>
<td>euclidean</td>
<td>ball tree</td>
<td>0.837978</td>
<td>0.044938</td>
</tr>
<tr>
<td>manhattan</td>
<td>ball tree</td>
<td>0.827230</td>
<td>0.058734</td>
</tr>
<tr>
<td>chebyshev</td>
<td>ball tree</td>
<td>0.820695</td>
<td>0.038600</td>
</tr>
<tr>
<td>hamming</td>
<td>ball tree</td>
<td>0.466513</td>
<td>0.052677</td>
</tr>
<tr>
<td>canberra</td>
<td>ball tree</td>
<td>0.723581</td>
<td>0.086002</td>
</tr>
<tr>
<td>braycurtis</td>
<td>ball tree</td>
<td>0.891077</td>
<td>0.044141</td>
</tr>
<tr>
<td>jaccard</td>
<td>ball tree</td>
<td>0.840933</td>
<td>0.054737</td>
</tr>
<tr>
<td>dice</td>
<td>ball tree</td>
<td>0.840933</td>
<td>0.054737</td>
</tr>
<tr>
<td>kulsinski</td>
<td>ball tree</td>
<td>0.751595</td>
<td>0.063372</td>
</tr>
<tr>
<td>rogerstanimoto</td>
<td>ball tree</td>
<td>0.731370</td>
<td>0.061987</td>
</tr>
<tr>
<td>russellrao</td>
<td>ball tree</td>
<td>0.625803</td>
<td>0.074475</td>
</tr>
<tr>
<td>sokalmichener</td>
<td>ball tree</td>
<td>0.731370</td>
<td>0.061987</td>
</tr>
<tr>
<td>sokalsneath</td>
<td>ball tree</td>
<td>0.840933</td>
<td>0.054737</td>
</tr>
<tr>
<td>euclidean</td>
<td>brute</td>
<td>0.838188</td>
<td>0.044796</td>
</tr>
<tr>
<td>manhattan</td>
<td>brute</td>
<td>0.827019</td>
<td>0.058927</td>
</tr>
<tr>
<td>chebyshev</td>
<td>brute</td>
<td>0.820274</td>
<td>0.038897</td>
</tr>
<tr>
<td>hamming</td>
<td>brute</td>
<td>0.462506</td>
<td>0.045803</td>
</tr>
<tr>
<td>canberra</td>
<td>brute</td>
<td>0.723791</td>
<td>0.086601</td>
</tr>
<tr>
<td>braycurtis</td>
<td>brute</td>
<td>0.891077</td>
<td>0.044141</td>
</tr>
<tr>
<td>jaccard</td>
<td>brute</td>
<td>0.840933</td>
<td>0.054737</td>
</tr>
<tr>
<td>dice</td>
<td>brute</td>
<td>0.840933</td>
<td>0.054737</td>
</tr>
<tr>
<td>kulsinski</td>
<td>brute</td>
<td>0.751595</td>
<td>0.061987</td>
</tr>
<tr>
<td>rogerstanimoto</td>
<td>brute</td>
<td>0.729263</td>
<td>0.064473</td>
</tr>
<tr>
<td>russellrao</td>
<td>brute</td>
<td>0.622429</td>
<td>0.071010</td>
</tr>
<tr>
<td>sokalmichener</td>
<td>brute</td>
<td>0.729263</td>
<td>0.064473</td>
</tr>
<tr>
<td>sokalsneath</td>
<td>brute</td>
<td>0.840933</td>
<td>0.055508</td>
</tr>
<tr>
<td>cosine</td>
<td>brute</td>
<td>0.914248</td>
<td>0.029629</td>
</tr>
</tbody>
</table>
### Table A.2: Accuracy for all Metrics (k = 3)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Index</th>
<th>Accuracy</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>euclidean</td>
<td>kd tree</td>
<td>0.827866</td>
<td>0.053549</td>
</tr>
<tr>
<td>manhattan</td>
<td>kd tree</td>
<td>0.817327</td>
<td>0.062069</td>
</tr>
<tr>
<td>chebyshev</td>
<td>kd tree</td>
<td>0.813746</td>
<td>0.042208</td>
</tr>
<tr>
<td>euclidean</td>
<td>ball tree</td>
<td>0.827866</td>
<td>0.053549</td>
</tr>
<tr>
<td>manhattan</td>
<td>ball tree</td>
<td>0.818170</td>
<td>0.062555</td>
</tr>
<tr>
<td>chebyshev</td>
<td>ball tree</td>
<td>0.813957</td>
<td>0.041596</td>
</tr>
<tr>
<td>hamming</td>
<td>ball tree</td>
<td>0.464402</td>
<td>0.049754</td>
</tr>
<tr>
<td>canberra</td>
<td>ball tree</td>
<td>0.710095</td>
<td>0.081823</td>
</tr>
<tr>
<td>braycurtis</td>
<td>ball tree</td>
<td>0.885812</td>
<td>0.046944</td>
</tr>
<tr>
<td>jaccard</td>
<td>ball tree</td>
<td>0.849785</td>
<td>0.065483</td>
</tr>
<tr>
<td>dice</td>
<td>ball tree</td>
<td>0.850628</td>
<td>0.065584</td>
</tr>
<tr>
<td>kulsinski</td>
<td>ball tree</td>
<td>0.765713</td>
<td>0.059737</td>
</tr>
<tr>
<td>rogerstanimoto</td>
<td>ball tree</td>
<td>0.717679</td>
<td>0.072936</td>
</tr>
<tr>
<td>russellrao</td>
<td>ball tree</td>
<td>0.633812</td>
<td>0.067216</td>
</tr>
<tr>
<td>sokalmichener</td>
<td>ball tree</td>
<td>0.717679</td>
<td>0.072936</td>
</tr>
<tr>
<td>sokalsneath</td>
<td>ball tree</td>
<td>0.849363</td>
<td>0.064541</td>
</tr>
<tr>
<td>euclidean</td>
<td>brute</td>
<td>0.827866</td>
<td>0.053549</td>
</tr>
<tr>
<td>manhattan</td>
<td>brute</td>
<td>0.818170</td>
<td>0.061136</td>
</tr>
<tr>
<td>chebyshev</td>
<td>brute</td>
<td>0.814378</td>
<td>0.040797</td>
</tr>
<tr>
<td>hamming</td>
<td>brute</td>
<td>0.459136</td>
<td>0.055243</td>
</tr>
<tr>
<td>canberra</td>
<td>brute</td>
<td>0.709674</td>
<td>0.081047</td>
</tr>
<tr>
<td>braycurtis</td>
<td>brute</td>
<td>0.885812</td>
<td>0.046944</td>
</tr>
<tr>
<td>jaccard</td>
<td>brute</td>
<td>0.849995</td>
<td>0.065003</td>
</tr>
<tr>
<td>dice</td>
<td>brute</td>
<td>0.849995</td>
<td>0.107582</td>
</tr>
<tr>
<td>kulsinski</td>
<td>brute</td>
<td>0.765502</td>
<td>0.060497</td>
</tr>
<tr>
<td>rogerstanimoto</td>
<td>brute</td>
<td>0.717048</td>
<td>0.065846</td>
</tr>
<tr>
<td>russellrao</td>
<td>brute</td>
<td>0.630017</td>
<td>0.076714</td>
</tr>
<tr>
<td>sokalmichener</td>
<td>brute</td>
<td>0.717048</td>
<td>0.064078</td>
</tr>
<tr>
<td>sokalsneath</td>
<td>brute</td>
<td>0.848731</td>
<td>0.039755</td>
</tr>
<tr>
<td>cosine</td>
<td>brute</td>
<td>0.906244</td>
<td>0.039755</td>
</tr>
</tbody>
</table>
Table A.3: Accuracy for all Metrics (k = 5)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Index</th>
<th>Accuracy</th>
<th>Std. Dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>euclidean</td>
<td>kd tree</td>
<td>0.824074</td>
<td>0.055560</td>
</tr>
<tr>
<td>manhattan</td>
<td>kd tree</td>
<td>0.815642</td>
<td>0.061258</td>
</tr>
<tr>
<td>chebyshev</td>
<td>kd tree</td>
<td>0.811429</td>
<td>0.045801</td>
</tr>
<tr>
<td>euclidean</td>
<td>ball tree</td>
<td>0.824074</td>
<td>0.055560</td>
</tr>
<tr>
<td>manhattan</td>
<td>ball tree</td>
<td>0.816696</td>
<td>0.062219</td>
</tr>
<tr>
<td>chebyshev</td>
<td>ball tree</td>
<td>0.811219</td>
<td>0.046123</td>
</tr>
<tr>
<td>hamming</td>
<td>ball tree</td>
<td>0.460613</td>
<td>0.055359</td>
</tr>
<tr>
<td>canberra</td>
<td>ball tree</td>
<td>0.690076</td>
<td>0.067420</td>
</tr>
<tr>
<td>braycurtis</td>
<td>ball tree</td>
<td>0.887290</td>
<td>0.053699</td>
</tr>
<tr>
<td>jaccard</td>
<td>ball tree</td>
<td>0.852103</td>
<td>0.064630</td>
</tr>
<tr>
<td>dice</td>
<td>ball tree</td>
<td>0.854210</td>
<td>0.061707</td>
</tr>
<tr>
<td>kulsinski</td>
<td>ball tree</td>
<td>0.764875</td>
<td>0.073723</td>
</tr>
<tr>
<td>rogerstanimoto</td>
<td>ball tree</td>
<td>0.702083</td>
<td>0.064854</td>
</tr>
<tr>
<td>russellrao</td>
<td>ball tree</td>
<td>0.660578</td>
<td>0.072486</td>
</tr>
<tr>
<td>sokalmichener</td>
<td>ball tree</td>
<td>0.702083</td>
<td>0.064854</td>
</tr>
<tr>
<td>sokalsneath</td>
<td>ball tree</td>
<td>0.851261</td>
<td>0.064230</td>
</tr>
<tr>
<td>euclidean</td>
<td>brute</td>
<td>0.823231</td>
<td>0.054043</td>
</tr>
<tr>
<td>manhattan</td>
<td>brute</td>
<td>0.814588</td>
<td>0.060960</td>
</tr>
<tr>
<td>chebyshev</td>
<td>brute</td>
<td>0.810586</td>
<td>0.045200</td>
</tr>
<tr>
<td>hamming</td>
<td>brute</td>
<td>0.463354</td>
<td>0.063333</td>
</tr>
<tr>
<td>canberra</td>
<td>brute</td>
<td>0.691129</td>
<td>0.065973</td>
</tr>
<tr>
<td>braycurtis</td>
<td>brute</td>
<td>0.887079</td>
<td>0.053926</td>
</tr>
<tr>
<td>jaccard</td>
<td>brute</td>
<td>0.350008</td>
<td>0.099094</td>
</tr>
<tr>
<td>dice</td>
<td>brute</td>
<td>0.858484</td>
<td>0.062671</td>
</tr>
<tr>
<td>kulsinski</td>
<td>brute</td>
<td>0.765295</td>
<td>0.071434</td>
</tr>
<tr>
<td>rogerstanimoto</td>
<td>brute</td>
<td>0.701664</td>
<td>0.069232</td>
</tr>
<tr>
<td>russellrao</td>
<td>brute</td>
<td>0.636137</td>
<td>0.072479</td>
</tr>
<tr>
<td>sokalmichener</td>
<td>brute</td>
<td>0.701664</td>
<td>0.069232</td>
</tr>
<tr>
<td>sokalsneath</td>
<td>brute</td>
<td>0.852525</td>
<td>0.066063</td>
</tr>
<tr>
<td>cosine</td>
<td>brute</td>
<td>0.901398</td>
<td>0.034491</td>
</tr>
</tbody>
</table>
Appendix B: Source Code

All code for this document, and the experiments described in it, is available at:

https://github.com/eseglem/thesis
Bibliography


http://kdd.isti.cnr.it/$\sim$nanni/papers/NWA_ICDE08.pdf


Biography

Erik L. Seglem will soon to be a PhD. student in the Earth Systems and Geoinformation Sciences program. He received a Bachelors of Arts in Intelligence Studies from American Military University in August of 2014, and is graduating from George Mason University with a Master’s of Science in Geoinformatics and Geospatial Intelligence. In November of 2016 he won First Place in the Graduate category of the Student Research Poster Competition at GMU’s GIS Day. With over 9 years of experience in GIS, including 5 years in the Marine Corps as a Geospatial Analyst, and 4 years as a Geospatial Software Engineer at DigitalGlobe Inc. He has taken a new position at BigBear Inc, where he will be focusing on cloud technology and big data as it applies to geospatial problems.