REPORTS OF
THE MACHINE LEARNING AND INFERENCE LABORATORY

LOCATION PREDICTION USING GPS TRACKERS

TOWARDS PREDICTING WANDERING IN PEOPLE WITH DEMENTIA

JANUSZ WOJTUSIAK
REYHANEH MOGHARABNIA

MLI 18-1
JUNE 2018

RESEARCH AND EDUCATION IN MACHINE LEARNING

Location Prediction Using GPS Trackers

Towards Predicting Wandering in People with Dementia

Janusz Wojtusiak, Reyhaneh Mogharabnia
{jwojtusi, rmoghara}@gmu.edu
Machine Learning and Inference Laboratory
Center for Discovery Science and Health Informatics
George Mason University

Abstract

Significant number of people with dementia are at risk of wandering and getting lost. These individuals may get hurt, cause distress to families and caregivers, and require costly search parties. This study explores the possibility of using machine learning methods applied to data from GPS trackers to create individualized models that describe patterns of movement. These patterns can be used to predict typical locations of individuals with dementia, and to detect movements that do not follow these patterns and may correspond to wandering. Data from a sample of 338 GPS trackers were used. After pre-processing the data are used for two-stage clustering, followed by classification learning. The number of clusters ranged between one (devices that always stayed "home") and seven for devices with maximum mobility. The average number of clusters was 2.33. Models for predicting location achieved varying accuracy, depending on regularity of wearer's schedule. The mean accuracy of 76% was achieved in predicting exact location of a device. Unusual locations that potentially correspond to wandering incidents can be identified as outliers from normal movement patterns.

Acknowledgements

The project would not be possible without help of GTX Corp. that provided access to GPS data and valuable expertise. The authors thank Hedyeh Mobahi, Katherine Irvin-Owen, Katherine Tompkins, Beverly Middle, Nalla Dural, and Andy Carle for collaboration and input at various stages of the project.

1. Introduction

Alzheimer's Disease (AD) and other forms of dementia constitute important public health concern. There are currently about 5.4 million individuals with dementia in the United States (Alzheimer's Association, 2017a), with 70-80% of all people with dementia in the US being cared for at home by a family member (Mayo Clinic, 2017) and 15 million caregivers provided annually an estimated 18.2 billion hours of care. In Virginia alone, there were approximately 447,000 family caregivers in 2013 and an estimated 455,000 in 2015 (Alzheimer's Association, 2017b). It is estimated that 60% of people with dementia will wander (Alzheimer's Association, 2017c). Wandering is a broad term that can be defined as "a syndrome of dementia-related locomotion behavior having a frequent, repetitive, temporally-disordered and/or spatially-disoriented nature that is manifested in lapping, random and/or pacing patterns, some of which are associated with eloping, eloping attempts or getting lost unless accompanied," (Algase et al., 2007). The wandering may be a result of a person with a dementia type such as Alzheimer's not being able to remember his or her name or address, and becoming disoriented even in familiar places. In the presented research we focus on a specific aspect of wandering, that is being lost outside of home. Wandering can occur during the mild, moderate or severe stages of AD and is potentially dangerous (leading to falls and fractures, institutionalization and death) and may cause significant stress for families and caregivers (Alzheimer's Association, 2017c; Rowe & Bennett, 2003). Characteristics and behaviors associated with wandering include having dementia for a longer duration, severity of dementia (though wandering can occur at any stage), presence of a sleep disorder, impairment in day-to-day functioning, and behavioral disturbances such as anxiety and depression (Ali et al., 2016). Recent research on the management of wandering behavior focuses on promoting safe walking which often includes electronic tagging of a person who wanders. GPS tracking of people with AD is often seen as unethical because it decreases a person's autonomy and the individual's right to privacy (Ali, et al., 2016; Yang and Kells, 2017), yet there are no alternatives except of constant supervision.

Ubiquitous presence of health trackers, GPS devices, smartwatches and other wearable technologies open new possibilities for improving safety and care for individuals with AD. The purpose, function and technology offered by these devices vary, and ranges from gait and movement analysis to assess physical activity, to alert systems that detect falls, to GPS trackers that help locate the missing. There are multiple trackers available on the market now, some of which are advertised specifically for the elderly or individuals with AD, including MX-LOCare, Mindme, Pocketfinder, GPS Shoe and its successor GPS SmartSole, to mention just few. The latter technology is used in this research as it provides real time monitoring of wearers. There are also other technologies such as one used by Project Lifesaver that uses radio location to help find the missing individuals (projectlifesaver.org). There are a number of previous research projects related to the approach presented here. The closest available research is by Shoval et al. (2008, 2011) in which the authors identified differences in movement patterns between people with mild dementia, MCI (Mild Cognitive Impairment) and no cognitive impairment. They found out that participants who suffer from mild dementia have much less varied mobility patterns than healthy participants and those with MCI and they usually stay close to their homes. They go out

at routine times, although it varies from person to person: some stay in the familiar surroundings while others move farther. However, in their approach, the authors did not consider prediction and detection of wandering patterns. According to a video-based observational study conducted by Saltzman et al., (1991) patients follow four basic travel patterns: direct trajectory, pacing, random trajectory and lapping. The latter 3 patterns are categorized as wandering. Vuong et al. (2011) used this rule to implement a classification algorithm for detecting these trajectories. In another experiment, the same team defined a feature vector for each trajectory including displacement, path length, total travel time, average velocity, straightness index, directional mean, and circular variance, and used existing supervised learning algorithms to classify trajectories into the 4 categories. They achieved the best accuracy of 72% using a Random Forest classifier. In another related research, Delaunay et al. (2017) detected wandering behaviors from movement patterns using GPS data collected by GPS watch trackers. Their algorithm uses the metrics coming from the GPS and sends an alert only when all the wandering behavior patterns are detected. Lin et al. (2012) proposed a real-time method to detect wandering behavior by finding adjacent turning points (points in a trace with a vector angle equal to or more than 90 degrees) within a distance range which form loop-like movements. Sposaro et al. (2010) built an android application to detect wandering patterns which uses two-phase approach proposed by Yin et al. (2007) in which a one-class Support Vector Machine (SVM) model filters out normal data, then the abnormal activities are detected from a normal activity model using kernel nonlinear regression (KNLR). The data collected is then evaluated by Bayesian networks which determine the probability of wandering behavior. Kearns et al. (2011) found a link between dementia diagnosis and path tortuosity (change of movement path) recorded in an indoor assisted living facility using Fractal Dimension (Fractal D) approach. Tung et al. (2014) used GPS data to measure life space of individuals with AD. There are many other examples of applications of machine learning methods to GPS data and prediction of location is relatively well established (i.e., Ashbrook & Starner, 2003; Zheng et al., 2008, 2010; Hightower, 2008; Feher & Forstner, 2011; Lin and Hsu, 2014).

The approach presented here has many similarities to research previously done by others, but is distinct in many ways. Our main objective was to create models capable of predicting person's location. This is achieved by a multi-stage process that involves unsupervised and supervised learning.

The remainder of this report has two main components, description of data and methods (Section 2) and analysis of obtained results (Section 3), followed by conclusion and research directions in GPS tracker data analysis.

2. Method

Machine learning techniques are used to identify frequency of wandering and detect spatiotemporal patterns that may allow for prediction of future incidents. Machine Learning (ML) is an interdisciplinary field that intersects artificial intelligence, statistics, cognitive science, computer science, and other disciplines. The general principle behind ML is that for very complex problems it is not possible to program computers to perform given tasks, instead computers learn

how to perform them. There is a wide range of applications areas that would not have been possible without ML methods, from self-driving cars, to automated image recognition and online shopping recommendations, to prediction of functional decline of the elderly (Wojtusiak et al., 2016). Although behind other disciplines, these methods are gaining popularity in medicine and health applications. There is often a confusion of why ML methods should be used rather than traditional statistics, including regression modeling. In short, ML methods allow for automated prediction of the unknown thus providing wide selection of techniques, while inferential statistics and traditional statistical modeling are about detecting regularities and trends in data, as well as studying asymptotic properties of created models.

In the reported research two types of ML methods are used. Unsupervised learning algorithms are used to find patterns in the data, that correspond to frequent locations in which GPS tracking devices are typically located. Supervised learning methods are used to construct models for predicting likely locations as well as unusual places. Because the data analyzed here did not include any information about the device users and dementia, in the presented work we cannot claim any relationship between obtained results and confirmed incidents of wandering, thus we assume that any non-typical movement is potentially related to wandering.

GPS SmartSole: SmartSoles by GTX Corp. are tracking devices that include GPS and GSM (cell phone) units embedded in shoe insoles to provide real-time geolocation data for wearers. The company also offers monitoring service, and notification of events in which wearers cross pre-defined geozones. The SmartSoles (Figure 1) fit many types of shoes and can be adjusted to wearers providing additional comfort (gpssmartsole.com). Because these devices are “hidden” inside shoes, they reduce chances of being discarded by individuals suffering from dementia known to remove foreign objects. The data used in this project has been obtained from GTX Corp. as part of data use agreement between the company and our research group. The team also received help and expertise in understanding the data and its limitations.

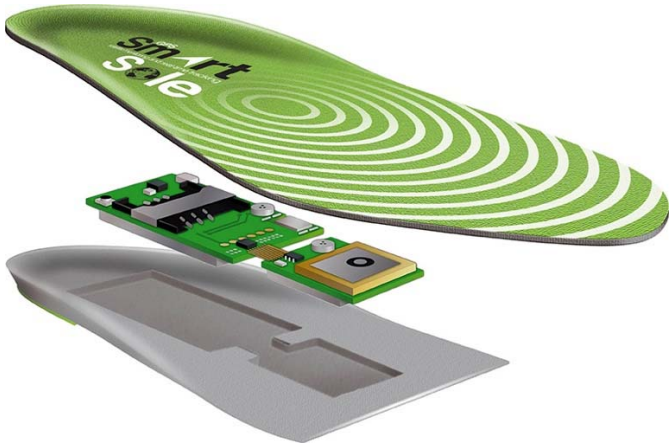


Figure 1: SmartSole design (source: gpssmartsole.com)

Spatiotemporal GPS Data: The obtained data consist of flat data tables that include the following fields: device ID, GPS timestamp, server timestamp, longitude, latitude, and device status. From the obtained data, a sample of 338 devices with at least 14 days of data was selected. ML methods were applied individually for each device, independent of other devices, thus creating individualized models. In the experiments, at least 7 days of data were used to test models and the reminder of data for training. The rationale for using time-based split is that models are intended to predict user behaviors in the future, and this is not guaranteed with random split between training and testing. This is illustrated in Figure 2. Summary of data is shown in table 1 and Figure 3.

Figure 2: Device data timeline.

Table 1 shows distribution of status codes for the devices and figure 3 shows the percentage of time spent in and outside during daytime and nighttime.

Sample size : 338	Mean	SD
Device status		
Percentage of time in motion	9%	11%
Percentage of time not moving	86%	15%
Other device status		
Percentage of time arrived at geozone	1%	7%
Percentage of time departed geozone	0.6%	2%
Percentage of time battery is low	3%	9%
Percentage of time spent at home	55%	30%
Number of data points	5667.56	7542.08
Number of days	99.63	125.21
	Grand Mean	**Grand SD**
Mean distance of device from home	31.24 km	198.56 km

Table 1: Device status statistics.

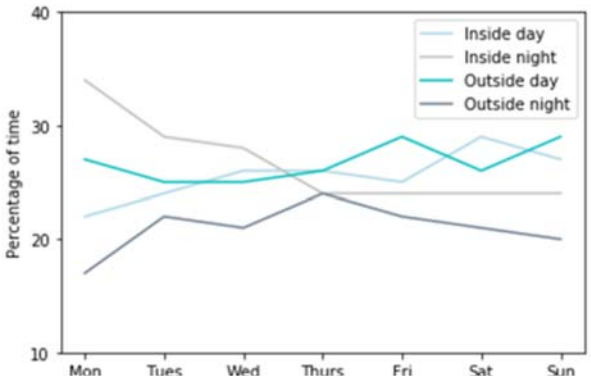


Figure 3: Percentage of time spent in and outside during the daytime and nighttime.

Data Analysis: The approach described below has been created to handle spatiotemporal nature of GPS data, detect patterns of typical movement, and unusual locations. To create patterns of movement for individuals with AD, raw GPS data needs to be normalized and transformed before machine learning algorithms can be applied (steps 1-4 below). Frequent locations are discovered by applying clustering algorithms (steps 5-7), followed by classification learning needed to predict frequent locations (step 9), and labeling the noise as routes between clusters and subsequently binarizing the labeling of data into typical and suspected wandering incidents (steps 10, 11). Finally, the unusual locations can be confirmed by secondary classification (step 12).
The steps are illustrated below with parts of Python 3 code used to implement the method. Machine learning library ScienceKit-Learn (sklearn) and data analysis library Pandas were also used.

1.	Select device and data: - Remove data for devices that: (a) were never activated outside of manufacturer's facility; (b) have less than 14 days of data. The minimum of 14 weeks of data is required later in the prediction step, 7 days for training and 7 days for testing. - Remove erogenous data points: (c) dataset data that correspond to the manufacturer location (initiation of each device and in some cases towards the end). These are typically first and last few minutes of the device's recorded activity. For SmartSole devices these are at the company location in Los Angeles, CA. (d) remove incorrect data, such as (0,0) coordinates, and those with dates in the past. The resulting project data is referred to as *D*.
2.	Calculate *home* location for each device as the most frequent *100x100 feet* rectangle present in the data *D*.
3.	Normalize and transform data D_{Norm}, so that the home location is in *(0,0)* coordinates, and other locations are randomly rotated. While not strictly needed for the following steps, this normalization and transformation ensures privacy and removes possible real location identification in the data.
4.	Convert timestamped geolocation data to *weighted frequency domain* in which weight of each data point is assigned as $$w_f(x_i) = \begin{cases} \frac{t_{i+1} - t_i}{c} & if\ dist(x_i, x_{i+1}) <= \lambda\ \wedge\ \neg s(x_{i+1}) \\ 1\ otherwise \end{cases}$$ where t_i and $t_{i+1}$is time measured at points x_i, x_{i+1}. In the data, by default the time is measured in seconds. We considered a constant c to be 600 seconds (10 minutes), that correspond to typical data frequency in GPS trackers used. A data point is weighed if its distance to the next point is not more than λ. Here, we used $\lambda = 0.05km$ after conducting experiments with different possible values. This approach weights only data when the device is stationary. $s(x_{i+1})$ indicates status of the device being initialized or switched off, thus preventing very high weights assigned to periods in which the device was off. The data do not indicate on/off status of devices; therefore, after experimental testing we decided that if the next point is more than τ = 3900 seconds far, the device is considered off and the point is not weighed.
5.	Apply weighted DBSCAN clustering algorithm with minimum support of $MinPts = \rho * \sum_i w(x_i)$ to identify top clusters $C_1, \ldots C_k$ with very high frequency in the data D^w_{Norm}. ρ is a parameter. After optimization, the values ρ = 0.1, 0.25, 0.3 were used in final experiments. Also, applying the same approach, the values $\varepsilon = 0.2km, 0.3km$ were used as the maximum distance between two points in order to be considered as neighbors. Training of DBSCAN in Python code is shown below. epsilon = ε min_sample = int(df['w'].sum() * ρ) dbscan_1 = DBSCAN(eps = epsilon / kms_per_degree, min_samples = min_sample, algorithm = 'ball_tree', metric = 'haversine') .fit(df[['x','y']], sample_weight = df['w']) labels_ = dbscan_1.labels_ unique_labels_ = set(labels_) n_clusters_1 = len(set(labels_)) - (1 if -1 in labels_ else 0) df = pd.DataFrame() df['id'] = deviceid

```
    df['t'] = T[:,0] # timestamps
    df['x'] = X[:,0] # latitudes
    df['y'] = X[:,1] # longitudes
    df['w'] = Y # weights
    df['l'] = labels_ # labels

    # Find home (label with the most sum of weights) & label it the max label, in order to distinguish it
later
    df_no_nois = df[df['l'] != -1]
    if not df_no_nois.empty:
        m = df_no_nois[['w','l']].groupby(['l']).sum().max()
        df_2 = df_no_nois[['w','l']].groupby(['l']).sum() == m
        lab = df_2[df_2['w']].index.values[0]

        new_max = max(labels_) + 1
        df.ix[df['l'] == lab, 'l'] = new_max

        # Relabel points starting from 0, in order to fill in empty space of former label of home
        j = 0
        dic = {}
        for i in sorted(set(df[df['l'] != -1]['l'])):
            dic[i] = j
            j += 1

        # Reassign new negative labels to clusters, because we will have same labels again in the second
dbscan
        # Later, once we have labels of the second dbscan we will reassign the positive numbers to these
labels
        labels = []
        for i in df['l']:
            if i != -1:
                label = -dic[i]-2
                labels.append(label)
            else:
                labels.append(i)

        df['l'] = labels
```

6. Remove from D^{w}_{Norm} all points belonging to clusters C_1, .. C_k. that is $\overline{D^{w}_{Norm}} = D^{w}_{Norm} - \bigcup C_l$

```
dff = df[df['label']==-1]
```

7. Apply the second round weighted DBSCAN clustering algorithm to points from $\overline{D^{w}_{Norm}}$ with minimum support of $MinPts = \begin{cases} \alpha * \sum_i w(x_i) \ where\ x_i \in \overline{D^{w}_{Norm}} if \ \alpha * \sum_i w(x_i) > \beta \\ \beta \ otherwise \end{cases}$
to obtain clusters C_{k+1}, … C_K. α and β are parameters. Using the pareto optimization technique, we chose values 0.01, 0.02, 0.03, 0.05 for α. In the initial experiments β is considered to be 30.

```
min_sample = int(dff['w'].sum() * α)
if min_sample > β:
    dbscan_2 = DBSCAN(eps = epsilon / kms_per_degree, min_samples = min_sample, algorithm =
                'ball_tree', metric = 'haversine').fit(dff[['x','y']], sample_weight = dff['w'])
else:
    dbscan_2 = DBSCAN(eps = epsilon / kms_per_degree, min_samples = β, algorithm = 'ball_tree',
                metric = 'haversine').fit(dff[['x','y']], sample_weight = dff['w'])

labels = dbscan_2.labels_
unique_labels = set(labels)
n_clusters_2 = len(set(labels)) - (1 if -1 in labels else 0)

df.ix[df['l']==-1, 'l'] = list(labels)
# Convert negative weights to positive
df.ix[df['l']<=-2, 'l'] = - df['l'] + max(list(db2.labels_)) - 1

n_clusters_ = n_clusters_1 + n_clusters_2
```

8. Output clusters $C_1, \ldots C_K$ labeled as the most frequent locations.

```
df_clusters = pd.DataFrame()
df_clusters = df[df['l'] != -1][['id','x','y','l']]
```

9. Construct models for predicting typical locations belonging to clusters $C_1, \ldots C_K$ given day and time.

- Remove the points between locations which are marked as noise in the clustering step, also remove the data related to the most frequent location (home) since it comprises most of the points and would make the prediction biased.
- Order data by timestamp, and use 7 last days of data or 25% of it towards the end whichever includes more days for testing and apply several classification methods.
- Calculate the accuracy and AUC. We chose Random Forest (Liaw, 2002) in our initial experiments because of the higher accuracy.

```
sql = "select tmp.deviceid, latitude, longitude, timestamp, day, hour, weight, label, \"# of days\" from
(select deviceid, (max(timestamp)- min(timestamp))/(3600*24) as \"# of days\" from data_30  group
by deviceid)tmp, data_30 where tmp.deviceid = data_30.deviceid and \"# of days\" >= 14;"
dff = pd.read_sql(sql, conn)
devices = dff['deviceid'].unique()
for device in devices:
    # Find home label
    m = dff[(dff['deviceid'] == device) & (dff['label'] != 1)][['weight','label']]
.groupby(['label']).sum().max()
    df_2 = dff[(dff['deviceid'] == device) & (dff['label'] != -1)][['weight','label']]
.groupby(['label']).sum() == m
    lab = df_2[df_2['weight']].index.values[0]
```

```
        # Remove noise and home
        df = dff[(dff['deviceid'] == device) & (dff['label'] != -1) & (dff['label'] != lab)]
        # Order data by timestamp
        df = df.sort_values('timestamp')
        sz = df['day'].size
        tr = df[:int(sz*0.75)]
        ts = df[int(sz*0.75):]

        min = ts['timestamp'].min()
        max = ts['timestamp'].max()
        if (max - min)+1 < 604800 : # a week
            cf = max – 604800
            # training set
            tr = df[df['timestamp'] <   cf]
            # test set
            ts = df[df['timestamp'] >=  cf]
        # Apply classification algorithm
        rf = RandomForestClassifier(n_estimators = 100)
        rf.fit(tr[cols], tr['label'])
        pred_rf = rf.predict(ts[cols])
        probs_rf = rf.predict_proba(ts[cols])

        result = pd.DataFrame()
        result['class'] = pred_rf
        result['probs'] = list(probs_rf)
        result['true'] = ts['label']
        # Calculate accuracy
        accuracy.append(float(result[result['true']==result['class']]['class'].count())/float(result['class'].
count()))

        # Calculate AUC
        probs=result['probs']
        j=0
        for i in np.sort(tr1.label.unique()):
           j+=1
           fpr, tpr, threshold = roc_curve(result['true'],probs[j],pos_label=i)
           if(np.isnan(fpr).any()): # if we don't have any Negative
              auc_list.append(1)
           else:
              if(np.isnan(tpr).any()): # if we don't have any Positive
                 auc_list.append(0)
              else:
                 auc_list.append(auc(fpr, tpr))

       mean_auc = np.mean(auc_list)
       auc.append(mean_auc)
```

10. Label the noise acquired in step 7; the routes are all labeled as $R_{Cx\text{-}Cy}$.

```
for device in devices:

    # Order data by timestamp
    cursor.execute("SELECT deviceid, timestamp, latitude, longitude, weight, cast(label as text) FROM
labels where deviceid='" + ''.join(device)+"' order by timestamp")
    X = cursor.fetchall()
    X = np.asarray(X)

    index = 0
    start = 0
    # If data starts with noise, pass it until you reach a cluster
    while X[index,5] == '-1':
        index = index + 1

    start = index
    while index < len(X[:,5]) - 1:
        # Pass the start cluster
        while index < len(X[:,5]) - 1 and X[index,5] == X[start,5]:
            index = index +1

        start = index - 1
        noise = index
        # Pass the noise between 2 clusters if any
        while index < len(X[:,5]) - 1 and X[index,5] == '-1':
            index = index + 1

        # Save cluster
        cluster = index
        if noise != index : # we had noise between clusters;
            end = index
            # Pass the end cluster
            while index < len(X[:,5]) - 1 and X[index,5] == X[end,5]:
                index = index + 1

            # Label noises between start and end clusters
            for i in range(start + 1, end):
                X[i,5] = 'R' + str(X[start,5]) + ',' + str(X[end,5])

            start = index - 1
        else: # we didn't have noise between clusters, we go to the next iteration;
            start = cluster
            index = cluster

    for i, j, k, l, m, n in zip(X[:,0],X[:,1],X[:,2],X[:,3],X[:,4],X[:,5]):
        cursor.execute("insert into \"labels_routes\"(deviceid, timestamp, latitude, longitude, weight,
label) values (%s, %s, %s, %s, %s, %s);",(i, j, k, l, m, n))
```

11. Label data that do not belong to clusters or regular routes between clusters as potential wandering incidents (anomaly detection), m_1, m_2, … m_p, and link that data to reported wandering incidents.

```
# Binarize labels for the next step's second classification
cursor.execute("select deviceid, timestamp, latitude, longitude, weight, label, label as label_2 from \"labels_routes\";")
X = cursor.fetchall()
df = pd.DataFrame(X)
arr = df.values
for elem in arr:
    if elem[5].find('R') != -1:
        # R_Cx_Cx
        if elem[5][elem[5].index('R') + 1: elem[5].index(',')] == elem[5][elem[5].index(',') + 1 : ]:
            elem[6] = 1
        # R_Cx_Cy
        else:
            elem[6]  = 0
    # cluster
    else:
        elem[6] = 0
```

12. Perform secondary classification to check for existence in patterns in unusual locations. This is done as binary classification in which points m_1, m_2, … m_p are classified against those from clusters C_1, … C_K and routes between them. Note that in this step, we expect AUC close to 0.5 that corresponds to unpredictable movements.

```
sql= "SELECT tmp.deviceid, latitude, longitude, timestamp, day, hour, weight, \"label_2\" as label, \"# of days\"  from \"labels_routes_binary\",(select deviceid, (max(timestamp)-min(timestamp))/(3600*24) as \"# of days\" from \"labels_routes_binary\"  group by deviceid)tmp where tmp.deviceid= \"labels_routes_binary\".deviceid and \"# of days\" >= 14;"
dff=pd.read_sql(sql, conn)
devices=dff['deviceid'].unique()
for device in devices:
    deviceids.append(device)
    df= dff[(dff['deviceid']==device)]
    # Order data by timestamp
    df = df.sort_values('timestamp')
    sz = df['day'].size
    tr = df[:int(sz*0.75)]
    ts = df[int(sz*0.75):]
    min = ts['timestamp'].min()
    max = ts['timestamp'].max()
    if (max - min)+1 < 604800 : # a week
        cf = max – 604800
        # training set
        tr = df[df['timestamp'] < cf]
        # test set
        ts = df[df['timestamp'] >= cf]
```

```
    # Apply classification algorithm
    if (len(tr) != 0) and (len(ts) != 0): # if training and test sets are not empty
      rf = RandomForestClassifier(n_estimators = 100)
      rf.fit(tr[cols], tr['label'])
      pred_rf = rf.predict(ts[cols])
      probs_rf = rf.predict_proba(ts[cols])

      result = pd.DataFrame()
      result[cols] = ts[cols]
      result['class'] = pred_rf
      result['probs'] = list(probs_rf[:,0])
      result['probs_1'] = 1
      result['probs_1'] = result['probs_1'].values - result['probs'].values
      result['true'] = ts['label']

      # Calculate accuracy
      Acc.append(float(result[result['true']==result['class']]['class'].count())/float(result['class'].count()))

      # Calculate AUC
      fpr, tpr, threshold = roc_curve(result['true'], result['probs_1'])
      AUC.append(auc(fpr, tpr))
    else: # if training and test sets are empty, go to the next iteration for the next device
      Acc.append(math.nan)
      AUC.append(math.nan)
```

Step 5 is needed because of large disproportion of weights between the most common (home) location and other locations, which also varies between devices. Without the step, it is practically impossible to select *MinPts* (parameter that defines minimum cluster size) that result in reasonable clustering. The above method relies on DBSCAN (Density-Based Spatial Clustering of Applications with Noise), which is a popular and very powerful clustering algorithm, particularly applicable to geospatial data (Sander et al., 1998).

In step 5-7, we have experimented with other clustering methods, including OPTICS (Ankerst et al., 1999), k-means, and hierarchical clustering, but decided on using BDSCAN because of its superior accuracy.

After spatiotemporal clustering, the data are ready for location prediction (step 9) in which ML is applied to create classification models for predicting normal location (given as a cluster/normal location $C_1..C_K$) given day and time. In our initial experiments, we have applied several methods from which Random Forest (Liaw, 2002) shows the best results. Random Forest models allow for automatically detecting geo-zones and can be used to raise alarms when the individual is not present at an expected location. In the future, we plan to collect sufficient longitudinal data to test the applicability of temporal deep learning techniques (i.e., Zhang et al., 2017), which are recently gaining popularity.

In steps 10-12, anomaly detection is applied to identify suspected wandering incidents, and then confirming them by linking them to survey data. In the future, we will be able to label specific parts of data as those in which individuals were lost. To do so, the data D^{w}_{Norm} are filtered to

remove all points belonging to normal locations $C_1..C_K$ as well as points in between (i.e., driving or walking between $C_1..C_K$). The resulting dataset D_{Anom} represents anomalies in the data, some of which may be indicative of wandering events will be matched against reported wandering. We believe that Gaussian Mixture Models (i.e., McLachlan & Peel, 2004), can be then applied to identify patterns of areas in which individuals with AD go missing. These models will be constructed from combination of GPS, survey, and landmark data. The quality of patterns will be measured using standard methods available in ML. Cross-validated Area under ROC Curve (AUC; C-statistic), precision and recall will be reported. In the initial set of experiments, we have explored large number of possible algorithm parameters, arriving at a number of satisfactory solutions that provide a compromise between precision accuracy and number of clusters.

Parameter Tuning: In order to choose the best possible parameters needed for our algorithms, we executed them with different combination of parameters needed in step 4 for calculating weights and in steps 5 and 7 for clustering.

To obtain threshold value for time gaps up to which the device could be assumed continuously on, indicted as $s(x_{i+1})$ in step 4, we analyzed frequencies of temporal distance between consecutive points reported by GPS trackers. These data are illustrated in histograms (Figure 4) that show averages for all devices as well as two randomly selected ones. The analysis indicated that 3900 seconds is an optimal value to be used as a threshold. Note that the work can be extended by adaptively selecting a device specific threshold, that would account for difference in usage patterns between different users.

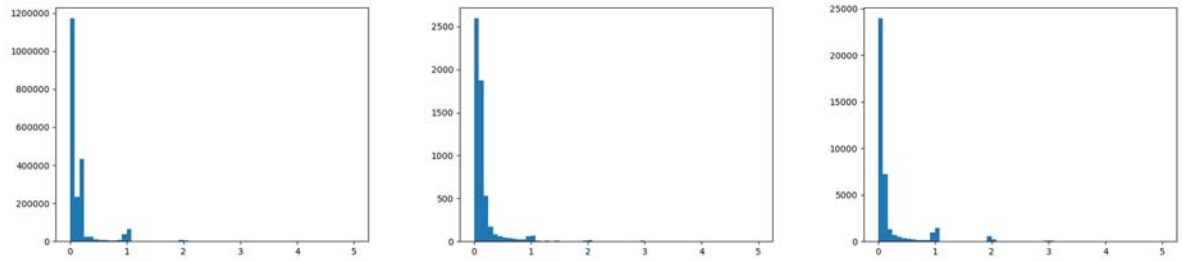


Figure 4: Frequencies of time gaps less than 5 hours for all the devices (left) one randomly selected device with approximately 5,000 data points (middle); and one device with approximately 40,000 data points (right).

In order to select the optimal parameters used by clustering algorithms, we performed optimization that searched over large number of combinations of possible values as follows:

'**λ**': maximum distance between two data points in order to be considered in the same region. The potential values are [0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9].
'**ε**': maximum distance between two points in order to be considered as neighbors in the DBSCAN clustering algorithm with possible values of [0.05, 0.1, 0.2, 0.3, 0.5, 0.7, 0.9].
'**ρ**': minimum support for the first DBSCAN with potential values of [0.1, 0.15, 0.2, 0.25, 0.3].

'**α**': minimum support for the second DBSCAN with possible values of [0.01, 0.02, 0.03, 0.05, 0.08, 0.1].

First, experimented with values of [0.3, 0.5, 0.7, 0.9], [0.1, 0.15, 0.2, 0.25] and [0.01, 0.05, 0.08, 0.1] for ε, ρ and α. Smaller values of ε and α resulted in higher accuracy whereas in the first DBSCAN, minimum sample support (ρ) with the bigger values had better results. Therefore, we chose [0.05, 0.1, 0.2], [0.2, 0.25, 0.3] and [0.01, 0.02, 0.03] for the second set of experiments. After running the clustering step with the above values and building our predictive model in the next step, we ran the pareto frontier optimization algorithm and arrived at a number of satisfactory solutions that provide a compromise between AUC and number of clusters (that allows for more precise location detection). The pareto optimal solutions chosen can be seen in Figure 5.

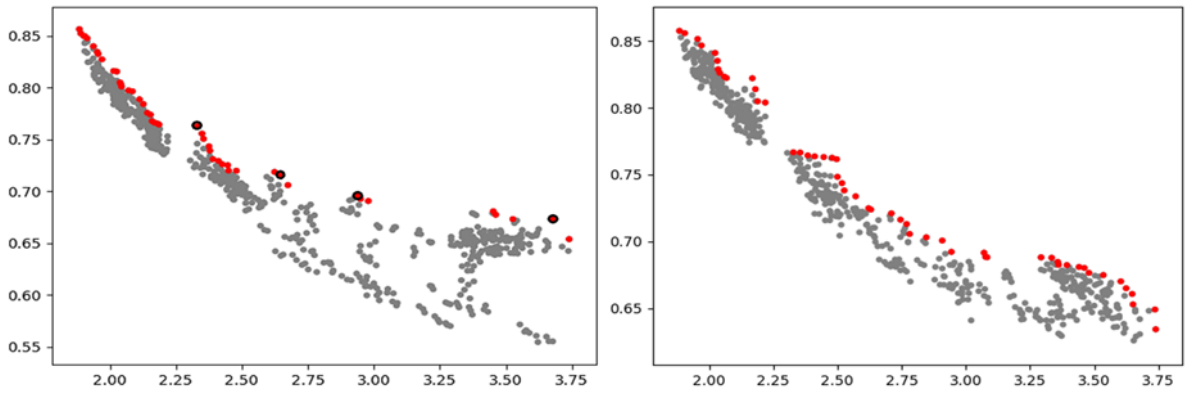


Figure 5: Average number of locations vs. average AUC (left) and accuracy (right). Pareto solutions are marked in red. The solutions circled are our selected choices.

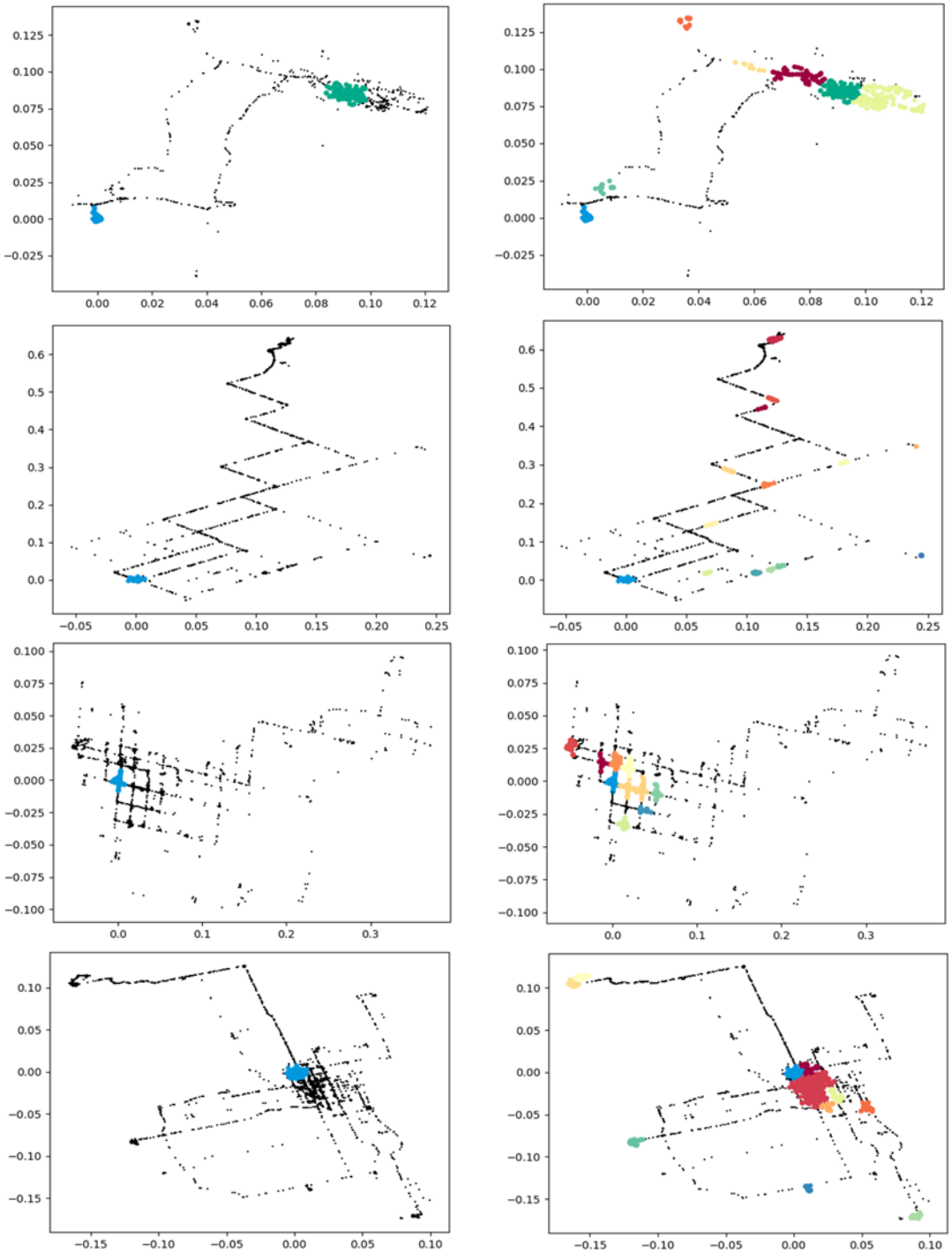
In the final set of experiments, we checked feasibility of tracking change of movement patterns, as identified by clusters over time. To do so, data were split into monthly periods and clusters generated within each month. The clusters were compared using Hotelling's_T-Squared test to indicate significant changes in movement.

3. Results

The method described in the previous section has been applied to a sample data obtained from GTX Corp. In this section, the obtained results are reported as follows. First, results of clustering are presented, followed by prediction of usual movement. Then, selected results that indicate possibility of detecting change of patterns over time are described. Finally, results of classification of unusual locations is shown.

Clustering: A typical data of location with marked (in color) frequently visited regions is presented in the Figure 6 below. Note that the location coordinates are recalculated so that home location is at (0,0) coordinates. The plots on the left represent clusters obtained in step 5 of the above method and plots on the right represent complete set of clusters (step 5 & step7). Table 2 shows

overall statistics of the clusters constructed with parameters of λ=0.05, ε=0.2, ρ=0.3 and α=0.02 (one of selected Pareto solutions).



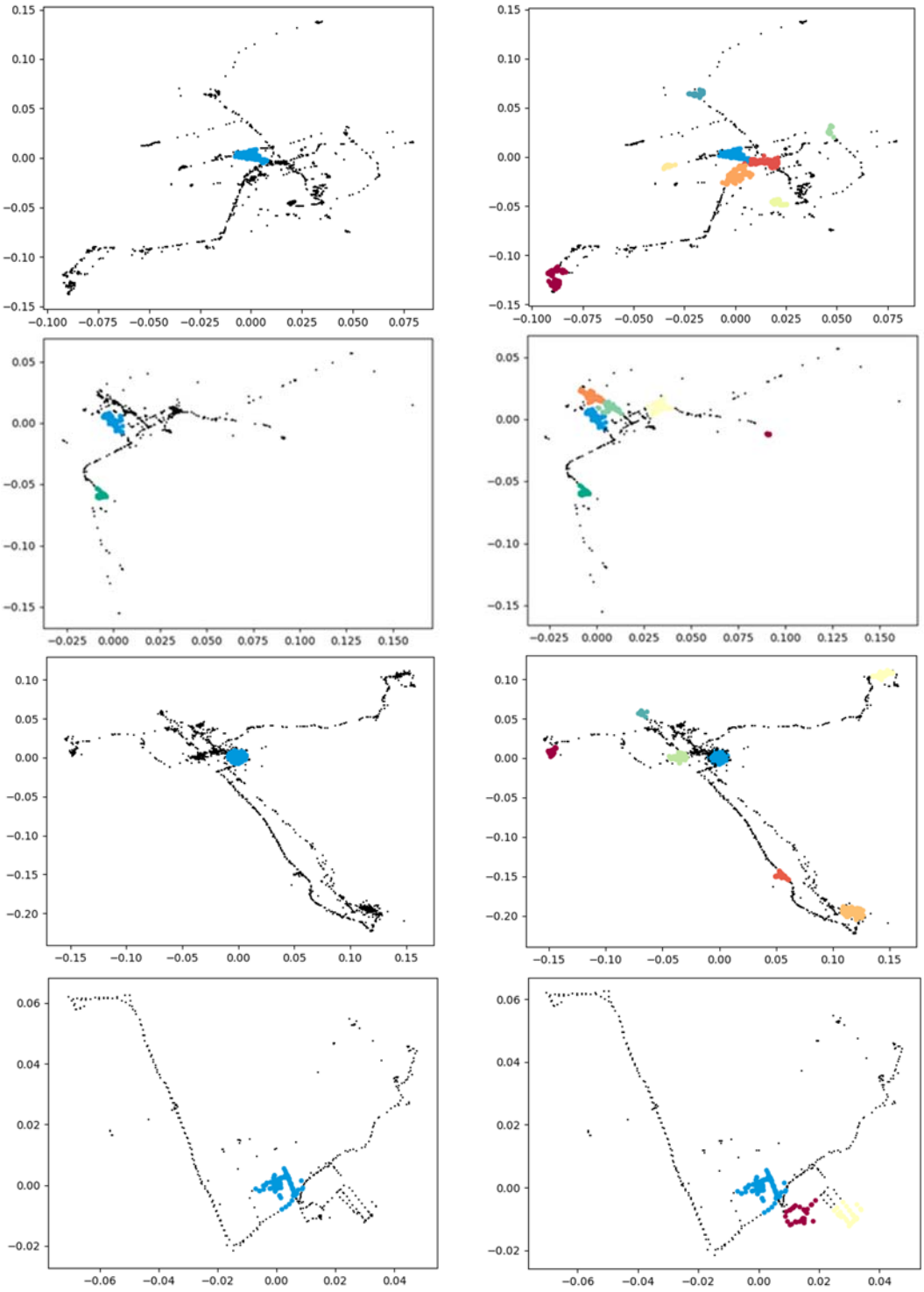


Figure 6: Frequently visited locations for a SmartSole device.
Left: clusters created in step 5; Right: complete set of clusters

	Mean	SD
Number of clusters	2.94	2.20
Number of clusters excluding home	1.94	2.20
	Grand Mean	**Grand SD**
Mean number of days spent in clusters	43.20	55.58
Mean number of days spent in clusters excluding home	6.48	19.59
Mean number of data points in clusters	2114.89	3411.80
Mean number of data points in clusters excluding home	217.77	316.14

Table 2: Cluster statistics based on 338 devices.

Prediction of Usual Movement: Next step after clustering is to apply supervised learning to construct models for predicting typical locations. According to initial results, the performance of the method depends on specific individual being followed. In the first approach, for some individuals, based just on day of week and time the method is able to correctly predict 100% of outside home locations, indicating a very regular lifestyle. For others, the accuracy can be as low as 0%. The four Pareto-chosen parameter sets (Figure 5) lead to average AUC of 76%, 72%, 70% and 67% (Figure 7). Further investigation why some devices have AUC of 0, indicates that one of the limitations of the project is availability of enough data: as it can be seen in figure 8, the clusters related to the testing data for one device (recorded in June) are completely different from the clusters in the training data (April & May). Table 3 reports average accuracies and statistics for prediction of locations.

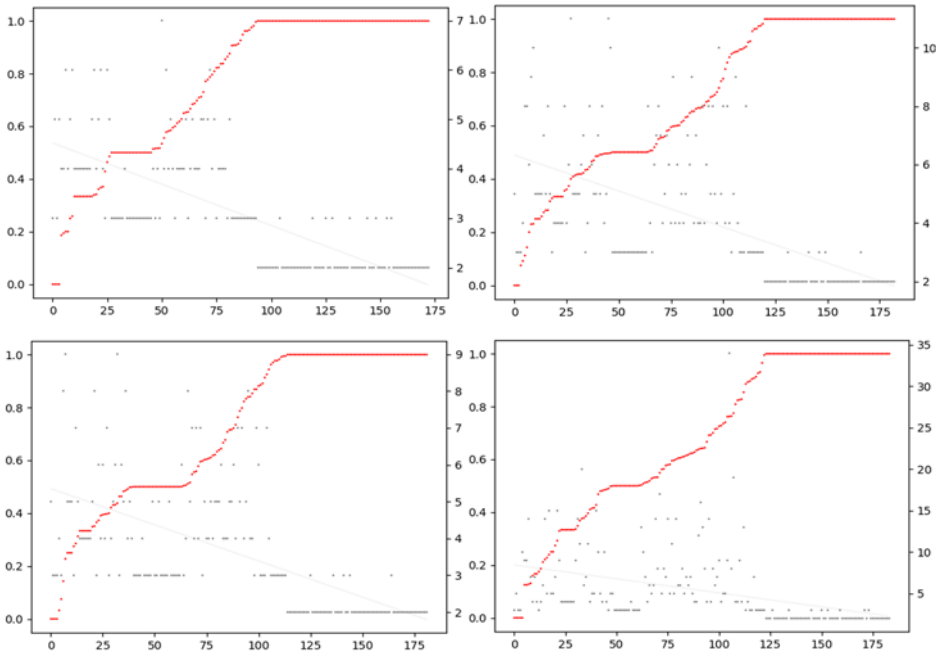


Figure 7: AUC & # of frequent locations of each device for the chosen pareto-optimal solutions. $\boldsymbol{\lambda = 0.05,\ \varepsilon = 0.3,\ \rho = 0.25,\ \alpha = 0.05}$; mean AUC = 76% (left top); $\boldsymbol{\lambda = 0.05,\ \varepsilon = 0.2,\ \rho = 0.3,\ \alpha = 0.02}$; mean AUC = 70% (right top)
$\boldsymbol{\lambda = 0.05,\ \varepsilon = 0.2,\ \rho = 0.25,\ \alpha = 0.03}$; mean AUC = 72% (left bottom); $\boldsymbol{\lambda = 0.05,\ \varepsilon = 0.3,\ \rho = 0.1,\ \alpha = 0.01}$; mean AUC = 67% (right bottom)

AUC (% of devices)	AUC <= 0.60 (43%)		0.60 < AUC <= 0.75 (11%)		AUC > 0.75 (46%)		Total	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Accuracy	0.43	0.33	0.57	0.26	0.95	0.12	0.68	0.35
Number of clusters	5.17	2.16	5.86	1.88	2.64	1.43	2.94	2.2
Number of days	138.63	121.61	158.81	137.09	70.58	61.21	99.63	125.21
Number of data points	7666.1	9031.48	13928.33	10929.61	5002.6	6628.94	5667.56	7542.08

Table 3: Prediction of usual movements statistics.

Change of movement patterns: Comparison of clusters over time indicate that movement patterns are stable for majority of devices. Hotelling's_T-Squared test was used to compare data in clusters over time. The obtained p-values were high (>0.1) which indicates that clusters representing movement patterns are consistent over time (Figure 9). However, as indicated earlier and depicted in Figure 8, a small portion of devices record data with radically different pattern of movement over time. Further work and additional data are needed to evaluate reasons for that irregularities.

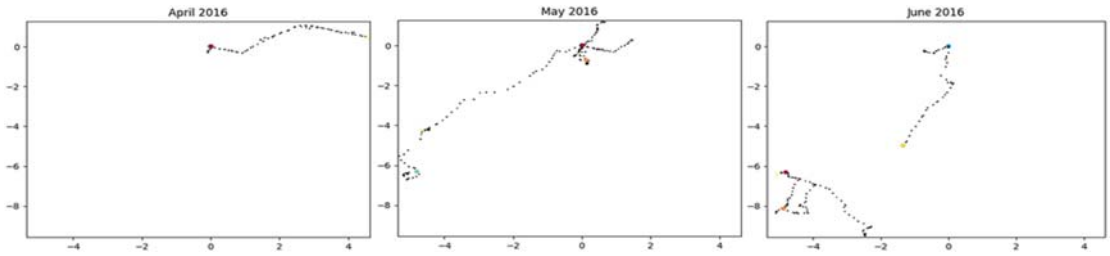


Figure 8: Monthly clusters of a device with 3 months of data.

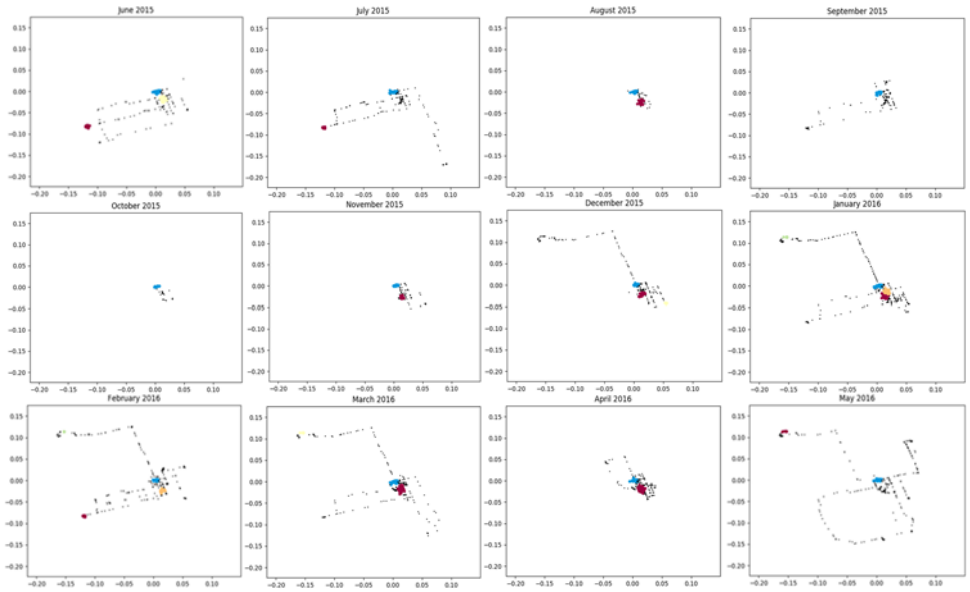


Figure 9: Clusters over time for three devices with the highest # of data points and longest recorded usage.

Detection of Unusual Movements: Finally, the data allows for detection of patterns in unusual activities, including wandering patterns. Note that in this project we were unable to link data to any AD information, thus could not distinguish data corresponding to wandering episodes from those that are anomalies simply not following patterns (i.e., individuals with AD accompanying family members). Therefore, we focus on purely data-driven approach to detecting and predicting unusual locations. To do so, the data D^{w}_{Norm} are filtered to remove all points belonging to normal locations $C_1..C_K$ as well as points in between (i.e., driving or walking between $C_1..C_K$). The resulting dataset D_{Anom} represents anomalies in the data, some of which may be indicative of wandering events.

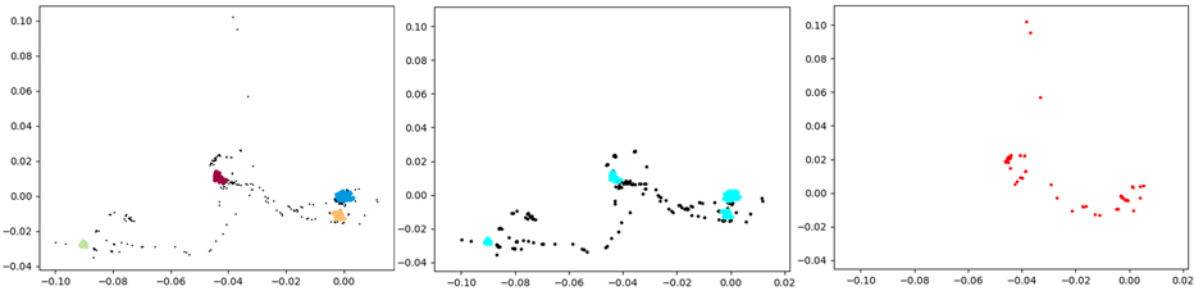


Figure 10: All data (left); normal locations and points in between (middle); filtered data (right).

AUC (% of devices)	AUC <= 0.60 (56%)		0.60 < AUC <= 0.75 (25%)		AUC > 0.75 (19%)		Total	
	Mean	SD	Mean	SD	Mean	SD	Mean	SD
Accuracy	0.87	0.16	0.87	0.15	0.89	0.08	0.90	0.14
Number of days	73.98	72.02	145.39	150.32	104.96	107.53	99.63	125.21
Number of data points	4234.47	5776.22	10317.94	9030.3	7080.26	10205.6	5667.56	7542.08
Number of unusual points	50.97	70.75	136.98	163.32	98.36	171.52	81.49	127.58

Table 4: Prediction of unusual movements statistics.

Supervised learning methods, can be then applied to identify if there are patterns in frequency of wandering and general area in which individuals with AD go. Specifically, we applied classification and as expected for most of the devices (56%) AUC is less than 0.60 which is indicative of unpredictable movements. Anomalies are depicted in Figure 10. Statistics of the classification are also shown in table 4.

4. Conclusion

The majority of people with AD are in danger of wandering including getting lost. Subsequently, these individuals may get hurt, cause extreme distress for family and caregivers, and require costly search efforts. The presented research aimed at finding patterns of movement that can eventually lead to prediction of wandering. The work represents first step in a long-term project. The method is based on spatiotemporal clustering used to detect normal locations where individuals typically are. Then clusters are passed to a machine learning algorithm to construct models for predicting typical locations of GPS tracker wearers. The method achieved the best AUC (and accuracy) of 76% in predicting locations just based on date and time. This high accuracy indicates highly regular lifestyle of most of individuals whose data were analyzed. This is an encouraging result, because it potentially allows for analyzing non-standard patterns of movement that may correspond to wandering and getting lost. Detection of unusual movements gave the initial AUC of 0.60 that is indicative of irregular movements outside of typical locations.

One important limitation of the presented work is that data are not limited to AD patients, and in fact there is no information about the device wearers at all. Currently the work is being extended by directed data collection from GPS trackers linked to clinical and socioeconomic information. Individuals with confirmed stage 3-6 Alzheimer's disease will be tracked for 2-3 years to collect sufficient data for prediction of wandering, and possibly linking wandering to progression of AD. On methodological side, the movement patterns are being linked to landmark data extracted from Open Street Maps. This will allow for detection of patterns related not only to coordinates and their relationships, but more importantly to what is located at a given location. The work is also extended with the possibility of predicting movement based on the last known location.

5. References

Algase DL, Moore DH, Vandeweerd C, Gavin-Dreschnack DJ. Mapping the maze of terms and definitions in dementia-related wandering. Aging & mental health. 2007 Nov 1;11(6):686-98.

Ali N, Luther SL, Volicer L et al. Risk assessment of wandering behavior in mild dementia. International journal of geriatric psychiatry. 2016 Apr 1;31(4):367-74.

Alzheimer's Association, 2017. Alzheimer's and Dementia Caregiver Center: Wandering and Getting Lost. Retrieved from http://www.alz.org/care/alzheimers-dementia-wandering.asp#who

Ankerst M, Breunig MM, Kriegel HP, Sander J. OPTICS: ordering points to identify the clustering structure. InACM Sigmod record 1999 Jun 1 (Vol. 28, No. 2, pp. 49-60). ACM.

Ashbrook D, Starner T. Using GPS to learn significant locations and predict movement across multiple users. Personal and Ubiquitous computing. 2003 Oct 1;7(5):275-86.

Delaunay, A., Guérin, J. Wandering Detection Within an Embedded System for Alzheimer Suffering Patients. AAAI Publications.

Feher M, Forstner B. Identifying and utilizing routines of human movement. InEngineering of Computer Based Systems (ECBS-EERC), 2011 2nd Eastern European Regional Conference on the 2011 Sep 5 (pp. 135-138). IEEE.

Hightower J, Consolvo S, LaMarca A, Smith I, Hughes J. Learning and recognizing the places we go. InInternational Conference on Ubiquitous Computing 2005 Sep 11 (pp. 159-176). Springer, Berlin, Heidelberg.

Kearns WD, Fozard JL, Nams VO, Craighead JD. Wireless telesurveillance system for detecting dementia. Gerontechnology. 2011:90.

Liaw A, Wiener M. Classification and regression by randomForest. R news. 2002 Dec 3;2(3):18-22.

Lin M, Hsu WJ. Mining GPS data for mobility patterns: A survey. Pervasive and Mobile Computing. 2014 Jun 1;12:1-6.

Lin Q, Zhang D, Huang X, Ni H, Zhou X. Detecting wandering behavior based on GPS traces for elders with dementia. InControl Automation Robotics & Vision (ICARCV), 2012 12th International Conference on 2012 Dec 5 (pp. 672-677). IEEE.

Martino-Saltzman D, Blasch BB, Morris RD, McNeal LW. Travel behavior of nursing home residents perceived as wanderers and nonwanderers. The Gerontologist. 1991 Oct 1;31(5):666-72.

Mayo Clinic, 2017. Alzheimer's caregiving: How to ask for help. Retrieved from https://www.mayoclinic.org/healthy-lifestyle/caregivers/in-depth/alzheimers-caregiver/art-20045847

McLachlan G, Peel D. Finite mixture models. John Wiley & Sons; 2004 Mar 22.

Rowe MA, Bennett V. A look at deaths occurring in persons with dementia lost in the community. American Journal of Alzheimer's Disease & Other Dementias®. 2003 Nov;18(6):343-8.

Sander J, Ester M, Kriegel HP, Xu X. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. Data mining and knowledge discovery. 1998 Jun 1;2(2):169-94.

Shoval N, Auslander GK, Freytag T et al. The use of advanced tracking technologies for the analysis of mobility in Alzheimer's disease and related cognitive diseases. BMC geriatrics. 2008 Dec;8(1):7.

Shoval N, Wahl HW, Auslander G et al. Use of the global positioning system to measure the out-of-home mobility of older adults with differing cognitive functioning. Ageing & Society. 2011 Jul;31(5):849-69.

Sposaro F, Danielson J, Tyson G. iWander: An Android application for dementia patients. InEngineering in Medicine and Biology Society (EMBC), 2010 annual international conference of the IEEE 2010 Aug 31 (pp. 3875-3878). IEEE

Tung JY, Rose RV, Gammada E. Measuring life space in older adults with mild-to-moderate Alzheimer's disease using mobile phone GPS. Gerontology. 2014;60(2):154-62.

Vuong NK, Chan S, Lau CT, Lau KM. Feasibility study of a real-time wandering detection algorithm for dementia patients. InProceedings of the First ACM MobiHoc Workshop on Pervasive Wireless Healthcare 2011 May 16 (p. 11). ACM.

Vuong NK, Chan S, Lau CT. Application of machine learning to classify dementia wandering patterns. Gerontechnology. 2014;13(2):294.

Wojtusiak J, Levy C, Williams A, Alemi F. Predicting Functional Decline and Recovery following Hospitalization of Residents in Veterans Affairs Nursing Homes. The Gerontologist. 2016; 56(1).

Yang YT, Kels CG. Does the Shoe Fit? Ethical, Legal, and Policy Considerations of Global Positioning System Shoes for Individuals with Alzheimer's Disease. Journal of the American Geriatrics Society. 2016 Aug;64(8):1708-1715.

Yin J, Yang Q, Pan JJ. Sensor-based abnormal human-activity detection. IEEE Transactions on Knowledge and Data Engineering. 2008 Aug;20(8):1082-90.

Zhang J, Zheng Y, Qi D. Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction. InAAAI 2017 Feb 12 (pp. 1655-1661).

Zheng VW, Zheng Y, Xie X, Yang Q. Collaborative location and activity recommendations with GPS history data. InProceedings of the 19th international conference on World wide web 2010 Apr 26 (pp. 1029-1038). ACM.

Zheng Y, Li Q, Chen Y, Xie X, Ma WY. Understanding mobility based on GPS data. InProceedings of the 10th international conference on Ubiquitous computing 2008 Sep 21 (pp. 312-321). ACM.

A publication of the *Machine Learning and Inference Laboratory*
College of Health and Human Services
George Mason University
Fairfax, VA 22030-4444 U.S.A.
http://www.mli.gmu.edu

Editor: J. Wojtusiak

The *Machine Learning and Inference (MLI) Laboratory Reports* are an official publication of the Machine Learning and Inference Laboratory, which has been published continuously since 1971 by R.S. Michalski's research group (until 1987, while the group was at the University of Illinois, they were called ISG (Intelligent Systems Group) Reports, or were part of the Department of Computer Science Reports).