Context Trees: Augmenting Geospatial Trajectories with Context

Alasdair Thomason, Nathan Griffiths, Victor Sanchez

Department of Computer Science,
University of Warwick, UK

June 2016

Abstract

Exposing latent knowledge in geospatial trajectories has the potential to provide a better understanding of the movements of individuals and groups. Motivated by such a desire, this work presents the *context tree*, a new hierarchical data structure that summarises the context behind user actions in a single model. We propose a method for context tree construction that augments geospatial trajectories with land usage data to identify such contexts. Through evaluation of the construction method and analysis of the properties of generated context trees, we demonstrate the foundation for understanding and modelling behaviour afforded. Summarising user contexts into a single data structure gives easy access to information that would otherwise remain latent, providing the basis for better understanding and predicting the actions and behaviours of individuals and groups. Finally, we also present a method for pruning context trees, for use in applications where it is desirable to reduce the size of the tree while retaining useful information.

1 Introduction

Exposing the latent knowledge present in geospatial trajectories has become an increasingly important research topic in recent years, due in part to the pervasiveness of location-aware hardware and the resulting availability of trajectory data. Motivated by a desire to understand the movement patterns of users, this paper presents a new data structure, the *context tree*, that summarises the context behind user actions in a single hierarchical model. Additionally, the paper proposes a method for generating context trees from geospatial trajectories and land usage information, and provides concrete implementations for each stage of the method, namely augmentation, filtering, and clustering. A context tree itself is formed of clusters at multiple scales that describe the contexts in which the user was immersed, affording easy access to information that would have previously remained hidden, forming the basis for understanding and predicting the actions and behaviours of individuals and groups.

Existing work in understanding people through the context of activities has considered various attributes as defining context, including an individual's location, the current time and weather, and other individuals who are nearby [Dey and Abowd, 1999; Schilit et al., 1994], typically using data collected from smartphones [Bao et al., 2011; Cao et al., 2010; Huai et al., 2014]. While existing approaches provide a basis for context-aware applications, they are limited by the data that can be collected directly from the user. Augmenting geospatial trajectories with land usage information enables the identification of contexts that consider the type and properties of the location of an activity.

In this paper we present the following contributions: (i) the *context tree* data structure that hierarchically represents user contexts at multiple scales, (ii) a method for constructing context trees from geospatial trajectories and land usage information, (iii) a set of concrete techniques to achieve each stage in the construction method, namely augmentation, filtering and clustering, (iv) evaluation of context trees constructed from real-world data, and an analysis of the properties that make them amenable for

Authors' contact details: {Alasdair.Thomason, Nathan.Griffiths, V.F.Sanchez-Sliva}@warwick.ac.uk

use in understanding individuals, and (v) a method of pruning context trees, to reduce their size while retaining useful information.

The remainder of this paper is structured as follows. Section 2 discusses relevant related work in location extraction and activity and context identification. In Section 3 we propose the context tree, a new data structure, and present an overview of the method employed for constructing context trees. Concrete implementations of the stages of this method are given in Sections 4 and 5. We present an evaluation of context trees in Section 6, and discuss pruning the generated trees in Section 7. Finally, we conclude the paper with a discussion of future work and applications in Section 8.

2 Related Work

Geospatial trajectories, usually collected from GPS logging devices, have been used as a basis for knowledge acquisition in many areas, including for location extraction [Andrienko et al., 2011; Ashbrook and Starner, 2002, 2003; Bamis and Savvides, 2011; Montoliu and Gatica-Perez, 2010; Thomason et al., 2015a, 2016]. Periods of low mobility are extracted from the trajectories and clustered using techniques such as DBSCAN [Ester et al., 1996] and k-means [MacQueen, 1967], identifying areas in which the time was spent. These techniques identify areas of arbitrary shape, but are incapable of identifying places where non-stationary activities took place. Augmenting identified areas with additional information, Yan et al. [2013] propose a technique for the derivation and modelling of *semantic trajectories*. However, the additional data sources are not leveraged for identifying locations, only for providing labelling after locations have been identified.

Once identified, significant locations have formed the basis for many applications, including location prediction using Markov models [Ashbrook and Starner, 2002, 2003], neural networks [Thomason et al., 2015c], periodicity-based approaches [Wang and Prabhala, 2012], and blockmodels [Fukano et al., 2013]. Using multilayer perceptrons for location prediction, Thomason et al. [2015b] evaluate extracted locations and predictions to perform automatic parameter selection for location extraction and prediction. Research has also considered predicting when a user will next visit a specific location using Bayesian inference [Gao et al., 2012], how long a user will stay at a given location [Liu et al., 2013], as well as developing techniques to apply labels in a semi-supervised manner to extracted locations to provide additional meaning [Krumm and Rouhana, 2013]. Prediction has also occurred without the need for location extraction, in the form of destination prediction, achieved by identifying similar historical trajectories to a current one through clustering approaches [Chen et al., 2010; Monreale et al., 2009; Nakahara and Murakami, 2012], Bayesian inference [Krumm and Horvitz, 2006] and hidden Markov models [Alvarez-Garcia et al., 2010]. Similarly, predicting journey duration has been explored using neural networks [Chen et al., 2009], along with predicting when two people will next meet [Yu et al., 2015], and providing recommendations to users new to a city based on the locations visited by others [Bao et al., 2015; Zheng and Xie, 2010].

While trajectories have also been used to identify non-stationary activities, in the form of transport mode identification through change-point detection and classification-based approaches [Liao et al., 2007; Patterson et al., 2003; Zheng et al., 2008a,b], many related techniques operate on different sources of data. Activity detection has been achieved from video data by Kim et al. [2010], who use Markov models to identify the activities being performed. Unfortunately, ensuring the constant availability of video data on an individual is infeasible. Research has therefore considered identifying the activity being performed from low-level sensor data (e.g. accelerometers and heart-rate) generated by devices carried by individuals, using classifiers and related techniques to label periods of data from a set of possible activities [Choudhury et al., 2008; Lee and Mase, 2002; Lester et al., 2005; Morris and Trivedi, 2011; Pirttikangas et al., 2006; Ravi et al., 2005].

Context, situation, and intention awareness have also been considered, where a context aims to identify times when a user was performing the same task, without necessarily knowing what the task is. Literature in this domain has explored using entropy-based clustering to identify contexts [Bao et al., 2011], and sequence-based approaches that consider the transitions between contexts [Lemlouma and Layaida, 2004]. Utilising contexts, research has also focused on developing architectures and applications that adapt devices based on the current context [Anagnostopoulos et al., 2006; Lemlouma and Layaida, 2004]. Situation and intention awareness is more focused on developing tools and techniques to aid a person in conducting a particular task to achieve some goal [Howard and Cambria, 2013; Vinciarelli et al., 2015], with specific examples in defence [Howard, 2002] and aviation [Endsley, 1995, 2000]. As with location extraction, however, existing techniques focus only on collected data, and do not attempt to augment this with other data available after collection. Such augmentation could offer greater insight into the entities a person was interacting with, enabling a better understanding of the actions they were performing.

Focusing on only trajectories, literature has also considered the identification of repeating patterns, both from geospatial trajectories [Cao et al., 2005, 2007; Eagle and Pentland, 2009; Giannotti et al., 2007; Gudmundsson et al., 2004], and general object movement trajectories [Li et al., 2010; Yang et al., 2003], where repeating patterns are expected to consist of activities that the user repeatedly conducts. Such patterns have also been considered as routines, where the aim is to extract features of a given day for classification (e.g. "left work at 5PM") [Farrahi and Gatica-Perez, 2008, 2010]. Patterns, and extracted location transitions, have formed the basis of user similarity identification [Xiao et al., 2012], and travel companion identification [Tang et al., 2012]. Once expected patterns for a given user or group have been extracted, anomalous actions become possible to identify. Anomaly detection has been performed on geospatial trajectories, where isolation-based outlier detection has identified anomalous subtrajectories from vehicle tracking data [Chen et al., 2011; Zhang et al., 2011]. Similarly, statistical approaches have been shown to be useful in identifying trajectories that differ from an expected pattern [Laxhammar and Falkman, 2011, 2014; Rosen and Medvedev, 2012].

Raw geospatial trajectories have been used as the basis for many different tasks and applications. While assuming the availability of additional data at time of collection is often infeasible, augmenting trajectories after collection is possible and can enrich the knowledge afforded. Applications that consider such augmented trajectories include using map data to fill in missing periods of a trajectory [Zheng et al., 2012], and using map searches augmented with trajectories from the same user to enhance destination prediction [Wu et al., 2015]. While existing work by Yan et al. [2013] has considered the augmentation of trajectories to understand the semantics behind trajectory segments, they do not attempt to utilise the semantics to influence the partitioning of trajectories or identify contexts. Understanding the semantics behind trajectories from the beginning has the potential to better understand what a person was doing and their interactions, and thus provide a foundation for identifying similar contexts.

2.1 Geospatial Datasets

Although it is increasingly becoming easier to collect geospatial data due to the proliferation of location-aware devices such as smartphones, the availability of public geospatial datasets still presents a challenge for researchers. Privacy concerns are the main obstacle to making such data publicly available, and consequently there are only a limited number of public datasets, each having certain drawbacks. To overcome these issues, many researchers have collected data themselves for use in their work [Ashbrook and Starner, 2003; Siła-Nowicka et al., 2015; Thomason et al., 2015a]. However, using such private datasets decreases the reproducibility of work and thus the use of public data is preferred. Such publicly available datasets include MIT's Reality Mining dataset [Eagle and Sandy Pentland, 2005], which uses cell towers to estimate the locations of devices belonging to 100 students. More recently, GPS-enabled devices have been used for data collection to produce Microsoft's GeoLife Trajectories [Zheng et al., 2008a, 2009, 2010], Nokia's Mobile Data Challenge (MDC) dataset [Kiukkonen et al., 2010; Laurila et al., 2012], and the Yonsei dataset [Chon et al., 2011]. While all of these datasets are available for research purposes, they have their own caveats. The Yonsei dataset contains only 2 months worth of data, while the GeoLife and MDC datasets contain vast amounts of data collected over several years from hundreds of users. The GeoLife dataset is focused on when participants were moving, and therefore does not provide continuous data. The MDC dataset, on the other hand, aims to provide continuous data collected from smartphones, although for privacy reasons the areas surrounding the known residences of participants have their accuracies significantly reduced in the data. Despite this, and due to its continuous collection methodology, the MDC dataset is still one of the most accurate and representative datasets available, containing real-world data collected from the smartphones carried by nearly 200 users over a span of 2 years. The MDC dataset also includes accuracy values, indicating how much confidence can be placed in the recorded coordinates, information which is not provided with GeoLife.

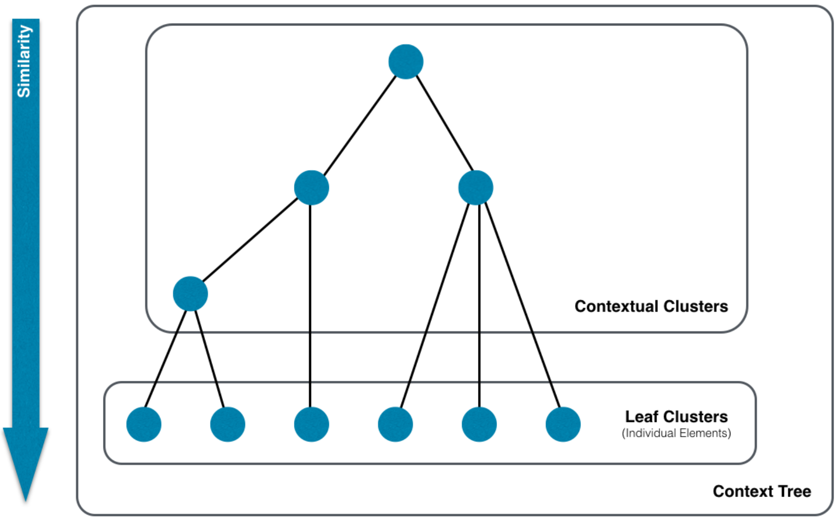


Figure 1: An abstract representation of a context tree, in which the similarity of nodes increases with depth.

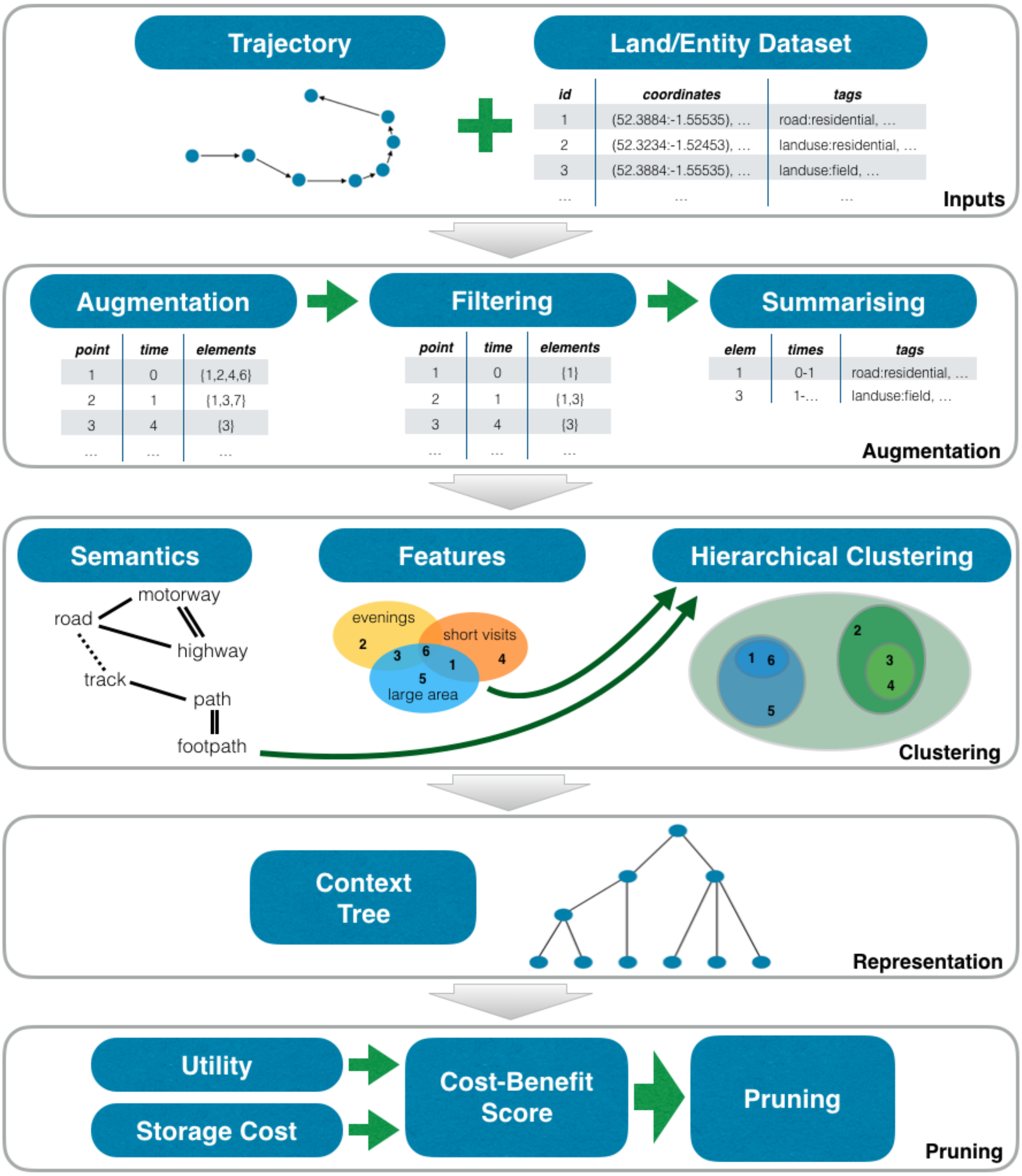
2.2 Investigative Techniques

For many techniques relating to extracting knowledge from data, collecting a concrete ground truth is infeasible. Significant location extraction, for example, can extract locations at various scales and so no single ground truth can exist. Existing literature addresses this by exploring the properties of the outputs from such techniques and comparing these properties to expected results. For instance, Guidotti et al. [2015] create synthetic trajectories with known properties and devise metrics to compare extracted locations with desirable properties. Thomason et al. [2015a] compare properties of the identified locations against acceptable ranges of values, determined from knowledge of the input data, as well as demonstrating the applicability of the technique through examples. For travel-mode classification, Siła-Nowicka et al. [2015] compare against a small set of manually labelled subtrajectories. In this paper, to cope with the limited availability of user-provided ground truth, we present an evaluation that both compares the output of the proposed approach to a limited ground truth and characterises the outputs of the algorithm through a set of metrics. While a ground truth may not exist in all domains, an understanding of the performance and applicability of the proposed approach can be achieved through characterising the outputs and manually generating or labelling subsets of the data to create a partial ground truth.

3 Proposed Structure: The Context Tree

This paper proposes and evaluates the *context tree* hierarchical data structure, that summarises the contexts that a user has been immersed within at multiple scales. Each leaf node of the tree represents a real-world feature or element that the user has likely interacted with, be it a specific building, area, or individual feature (e.g. a bench in a park). These individual elements are joined together through *context nodes* that represent a context at a specific scale, where time spent within a context means that the user likely had similar aims or goals, and are identified by exploring time the user spends interacting with elements with similar properties, or elements that are interacted with in a similar manner. As it summarises time in this way, the context tree can become the basis for understanding people from augmented geospatial data. The context tree structure is depicted in Figure 1.

Generating a context tree requires both a geospatial trajectory and a dataset of land usage features, along with a multi-stage process for augmentation, filtering and clustering this data into a useful structure. The remainder of this paper presents the proposed method for generating context trees, provides an evaluation of context trees, and presents a method of pruning context trees to reduce their size while maintaining information. The method for augmenting geospatial trajectories with land usage information, to summarise contexts into a context tree, consists of the following five stages, as depicted in Figure 2.



id	coordinates	tags
1	(52.3884:-1.55535), …	road:residential, …
2	(52.3234:-1.52453), …	landuse:residential, …
3	(52.3884:-1.55535), …	landuse:field, …
…	…	…

point	time	elements
1	0	{1,2,4,6}
2	1	{1,3,7}
3	4	{3}
…	…	…

point	time	elements
1	0	{1}
2	1	{1,3}
3	4	{3}
…	…	…

elem	times	tags
1	0-1	road:residential, …
3	1-…	landuse:field, …

Figure 2: Overview of the context tree generation framework. A trajectory is augmented with land usage data, and this augmented data is then hierarchically clustered into a context tree. Subsequently the context tree can optionally be pruned.

1. **Inputs**
   The raw geospatial trajectory and land usage data enters the system.
2. **Augmentation**
   Land usage elements likely to have been interacted with are identified by extracting all potential elements and filtering them to remove noise.
3. **Clustering**
   Filtered land usage elements, and their interactions, become the basis for contextual clustering. Clustering is achieved with a hierarchical agglomerative algorithm.
4. **Representation**
   Once clustered, the elements form a context tree data structure that can be used as the basis for further understanding the behaviours of individuals and groups.
5. **Pruning**
   Some applications may be limited by the amount of data they can store, or processing they can perform, and so it may be necessary to prune a context tree to reduce its size while maintaining as much useful information as possible. Pruning is achieved through analysing the nodes of a context tree with respect to a defined set of metrics.

In the following sections we describe the stages of augmentation (Section 4), clustering (Section 5), representation (Section 5), and pruning (Section 7) in more detail.

4 Trajectory Augmentation

In order to better understand users through their past actions, and assuming only geospatial data is available at the point of data collection, this section describes the process of trajectory augmentation that combines raw trajectories with land usage data. A trajectory is a temporally ordered sequence of data points that locate an individual or entity:

$$T = (p_1, p_2, p_3, ..., p_n)$$

where $p_i = \{t_i, l_i, a_i\}$ is an individual trajectory *point*, consisting of time (t_i), location (l_i, e.g. a $< lat, lng >$ pair) and accuracy (a_i, typically measured in metres).

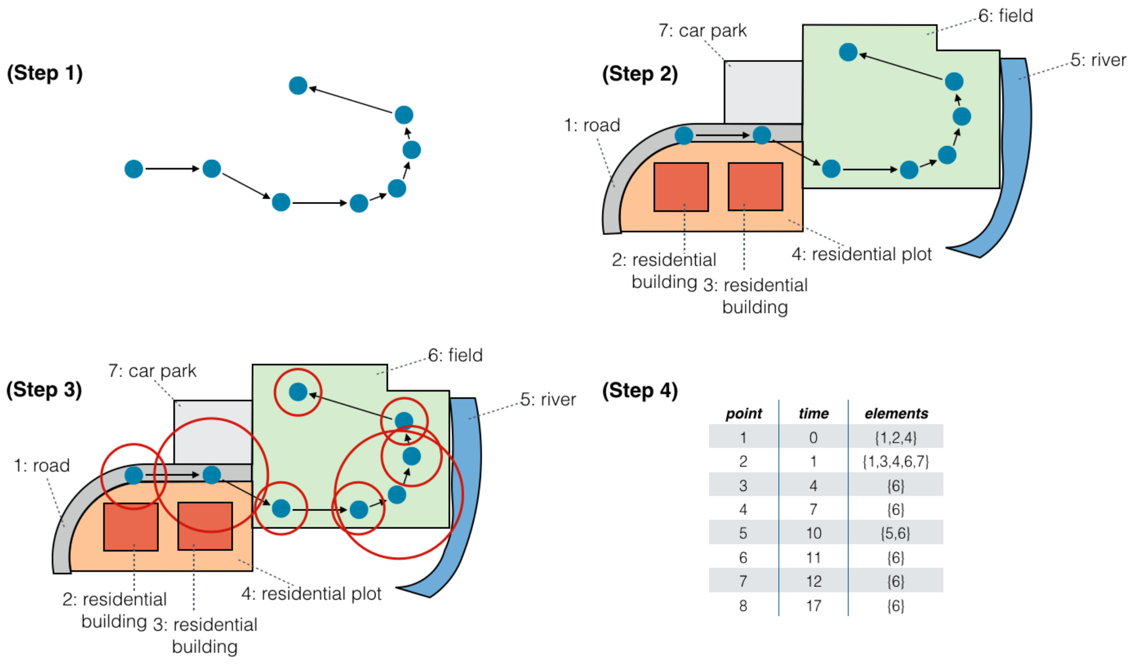
In addition to such trajectories, land usage data can also be used for identifying locations and entities that are meaningful to the user. Land usage data is assumed to be sets of *entities* with associated information. An entity, in this case, directly maps to a single real-world object, feature, or area, such as an individual postbox, field, or building. It can also refer to a collection of such entities that form a larger designation, such as a university campus or residential housing area. Each of these elements is expected to be associated with a set of geographical coordinate pairs that represent its shape and location, in addition to a set of tags in the form of 'key:value' pairs that describe properties of the element, including its type and usage (e.g. a house may be tagged as 'building:residential').

4.1 Element Extraction

The process for extracting relevant land usage elements is illustrated in Figure 3. A raw geospatial trajectory (Step 1) is overlaid on a land usage dataset (Step 2), at which point the accuracy recorded by the location measuring device (e.g. GPS, measured in metres) is used (Step 3), such that all elements that are partially or wholly within the radius are stored alongside the original trajectory point (Step 4). This procedure is completed automatically by iterating through each trajectory point and querying the land usage dataset for any element that intersects or covers any part of the accuracy radius.

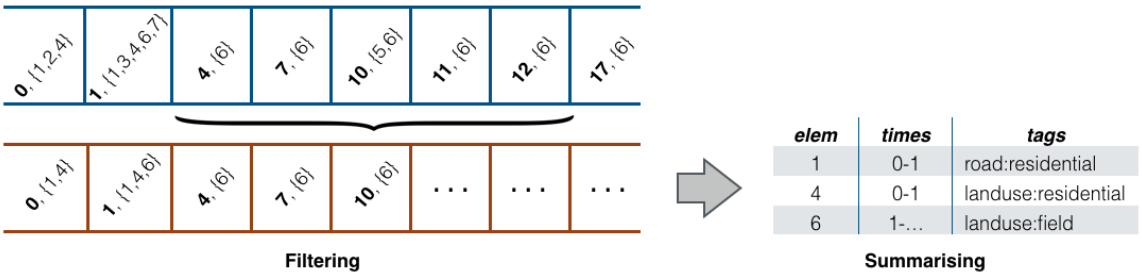
4.2 Filtering

As the element extraction process (Section 4.1) augments trajectories with all land usage elements that fall within the *accuracy radius* of a trajectory point, it is prone to including a significant number of elements with which the user was not interacting. To cope with these noise elements a filtering



point	time	elements
1	0	{1,2,4}
2	1	{1,3,4,6,7}
3	4	{6}
4	7	{6}
5	10	{5,6}
6	11	{6}
7	12	{6}
8	17	{6}

Figure 3: Trajectory augmentation procedure.



0, {1,2,4}	**1**, {1,3,4,6,7}	**4**, {6}	**7**, {6}	**10**, {5,6}	**11**, {6}	**12**, {6}	**17**, {6}
0, {1,4}	**1**, {1,4,6}	**4**, {6}	**7**, {6}	**10**, {6}	. . .	. . .	. . .

elem	*times*	*tags*
1	0-1	road:residential
4	0-1	landuse:residential
6	1-...	landuse:field

Figure 4: Example of filtering augmented trajectories to remove noise, and subsequent summarising through clustering of contiguous time periods.

procedure can be used. Our proposed filter is a generalised version of a weighted average filter, a technique typically used to smooth noisy signals, modified to operate over sets of land usage elements and depicted in Figure 4. The filter maintains a buffer of elements and selects from this buffer based on an assigned weight in a three-step process:

1. A buffer of points, and associated land usage data, is selected.
2. The land usage elements in the buffer are weighted and scored.
3. Elements are selected, based on their score, for inclusion in the output.

4.2.1 Buffer Selection

Due to the nature of geospatial data collection systems, a continuous and evenly timesliced trajectory cannot be assumed, and so selecting a buffer based on a fixed number of points would be inappropriate. Instead, we use a fixed temporal width for the buffer and consider all points that fall within this period. A buffer therefore consists of a *point under consideration*, and the points falling within δ seconds immediately before or after this point. The pseudocode for maintaining such a buffer is presented in Algorithm 1.

ALGORITHM 1 Buffer Management

```
points ← ($p_1, p_2, ...$) // input set
$\delta$ ← 300 // input parameter specifying buffer width
buffer ← [ points.shift ]
output ← [ ]
index ← null

// Build the initial buffer
while points.length > 0 do
    // If index has not been set, then we are in the first half
    if index == null && TimeBetween(buffer[0], points[0]) > $\delta$ then
        // If the next point is greater than $\delta$ seconds from the first, then the first half is full
        index ← buffer.length − 1
    // If index has been set, then we are in the second half
    else if index ! = null && TimeBetween(buffer[index], points[0]) > $\delta$ then
        break // Exit the loop as adding the next point would exceed $\delta$
    else
        buffer.append(points.shift)
    end if
end while

// Process the current buffer, increment index and maintain the new buffer
while points.length > 0 do
    output.append(Filter(buffer, index)) // Perform the actual filtering
    index ← index + 1

    // If the point for consideration is not in the buffer, then add it now
    if index == buffer.length then
        buffer.append(points.shift)
    end if

    // Remove any point from the first part that is not within $\delta$ seconds of buffer[index]
    while TimeBetween(buffer[0], buffer[index]) > $\delta$ do
        buffer.shift
        index ← index - 1
    end while

    // Add points until doing so would exceed $\delta$ seconds from buffer[index]
    while points.length > 0 && TimeBetween(buffer[index], points[0]) <= $\delta$ do
        buffer.append(points.shift)
    end while
end while

return output
```

4.2.2 Scoring

Scores are then applied to each land usage element in the buffer, weighted by the number of points the element is associated with, the accuracy of these points and the temporal distance from the point under consideration. Since we are dealing with sets, rather than the filter simply averaging values over the buffer, the process is modified by assigning weighted scores to each set element and then selecting elements according to a threshold. Combining these factors into a score, we have:

$$Score(e) = \sum_{p \in P_e} \left(\frac{1}{a_p} \times \left(1 - \frac{dist(p, p_c)}{\delta} \right) \right) \times |P_e| \tag{1}$$

where P_e is the set of all points that are associated with element e, a_p is the accuracy value of point p (in metres, such that low values indicate that there is likely to be less noise), p_c is the point under consideration, δ is the width of the buffer (i.e. the maximum number of seconds from p_c to consider) and

$dist(p_1, p_2)$ is the number of seconds between points p_1 and p_2 (temporal distance). Equation 1 gives a higher score to elements associated with a large number of high accuracy points (where high accuracy is recorded as a small value). Scores are then normalised relative to the maximum:

$$NormalisedScore(e) = \frac{Score(e)}{argmax_{Score}(Score(e) : \forall e \in \mathit{buffer})} \tag{2}$$

4.2.3 Selection

With each element in the buffer assigned a score, selection can occur either by using a fixed threshold to discard low-scoring elements, or by keeping all elements but limiting their effect through soft-thresholding. Soft-thresholding is a technique commonly applied in signal processing, where a kernel is applied to the calculated scores, forcing higher scores closer to 1 and lower scores closer to 0. While soft-thresholding removes the need to apply a fixed threshold, for this work we are only concerned with whether or not an element is included in the output set and thus we employ a threshold, t, where any element with a $NormalisedScore$ of greater than t becomes part of the output set, and the remaining elements are discarded.

4.3 Data Summarisation

Once filtered, augmented trajectories contain a record of where an individual was at a given time, along with the real-world features they were likely interacting with. These interactions are summarised into continuous spans of time by considering each land usage element encountered. If the same land usage entity is associated with two consecutive points, it can be assumed that it is also associated with the period of time between these points, if such a period of time is sufficiently small. An example summary is shown in Figure 4 (right).

If the time between consecutive points is large, it cannot be known whether the user ceased interacting with an element and resumed again before data collection next occurred, and so a limit on the time between consecutive points is specified as t_{max}. If the time between two consecutive points associated with the same element is greater than t_{max}, then the interactions are split, a technique also used in location extraction applications [Montoliu and Gatica-Perez, 2010; Thomason et al., 2016]. This results in a summary list of land usage elements along with a set of times, during which the individual can be assumed to have been interacting with the element in question.

5 Contextual Clustering

The identification of similar *contexts* is performed through clustering that considers both the properties of the elements and the properties of user interactions to determine similarity. Rather than aiming to identify a single level of clusters, which would limit the utility and applicability of the clusters to a single scale, the goal here is to build a hierarchical model, constructed by progressively merging land usage elements that represent similar contexts in a context tree, a depiction of which is shown earlier in Figure 1.

5.1 Building Clusters

Initially, each land usage element is distinct and is treated as a singleton cluster (i.e. a cluster with exactly one element). At each round of clustering, several of these clusters are merged to represent a context and a new higher level in the hierarchy, with pointers between the levels considered as *parent* and *child* relationships. That is, if two clusters at one level become merged into another cluster at the next level, the original clusters are considered as *children* of the new cluster. This section describes how clusters are merged with respect to their properties.

As discussed in Section 4, land usage elements are assumed to have a set of *tags* in the form of 'key:value' pairs that describe properties of the real-world entity to which the element relates, in addition to *geographical coordinate sets* that describe the geographical properties of the real-world entity. Once augmented and summarised, these elements are also associated with a set of *times.* When clusters are merged to create a context tree, the following procedures are used:

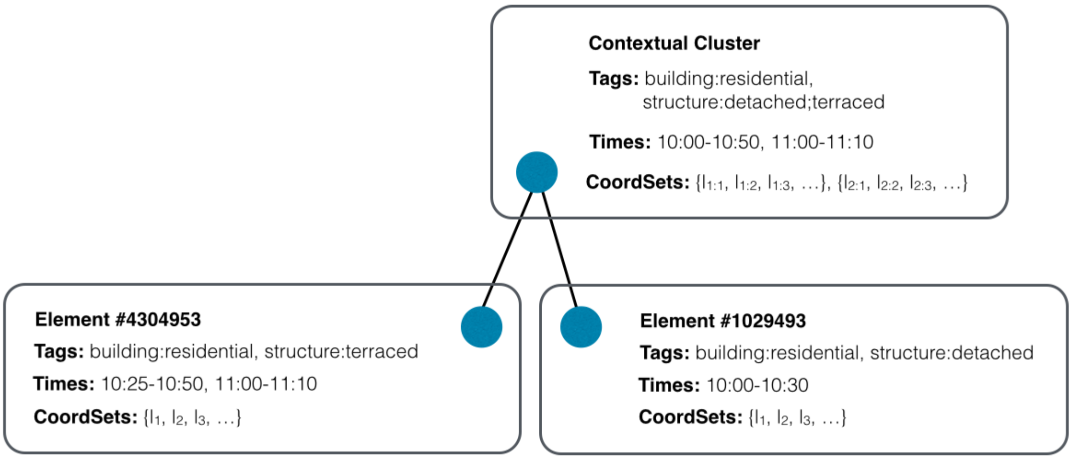

Contextual Cluster
Tags: building:residential, structure:detached;terraced
Times: 10:00-10:50, 11:00-11:10
CoordSets: {l1:1, l1:2, l1:3, …}, {l2:1, l2:2, l2:3, …}
Element #4304953
Tags: building:residential, structure:terraced
Times: 10:25-10:50, 11:00-11:10
CoordSets: {l1, l2, l3, …}
Element #1029493
Tags: building:residential, structure:detached
Times: 10:00-10:30
CoordSets: {l1, l2, l3, …}

Figure 5: Cluster merging example.

Times

The times for the merged cluster are taken to be the union of the sets of times from all child clusters, where overlapping time ranges are themselves combined into one. For example, if one cluster had the set of times {10:00-10:05, 11:00-12:00} and another had {10:04-10:20, 11:10-11:15, 12:05-12:09}, then the merged times would be {10:00-10:20, 11:00-12:00, 12:05-12:09}.

Tags

Similarly, each element has associated tags. The tags of the merged cluster are defined as the union of tags from the child clusters, where if two tags share a key but not a value, both values are stored.

Geographical Coordinate Sets

Each element contains a set of coordinates that define the geographical shape of the entity to which they relate. Merging such elements should keep each of these sets discrete, unless they intersect, in which case the coordinates belonging to both shapes are combined and replaced with their convex hull.

The merging of *Times* assumes a periodicity of 24 hours, which while reasonable for many people (i.e. those who follow a daily routine), it may not be appropriate for everyone. As such, automatic time series learning could be utilised to better learn meaningful movement patterns of the individual. While exploring such techniques is beyond the scope of this paper, there are many existing approaches that may be effective for the task [Ahmad et al., 2004]. An example merging of two elements according to these rules is shown in Figure 5, where it is assumed that there is no geographical overlap between the two elements (i.e. the coordinate sets cannot be merged).

5.2 Contextual Distance Metrics

Clustering elements together requires a distance metric to measure element similarity. While identifying contexts from certain types of data is a task considered before, and discussed in Section 2, no metrics currently exist that have been tailored to the identification of contexts from augmented geospatial trajectories. This section presents metrics that encapsulate the goals behind context extraction for this specific problem, with an emphasis on properties of the interactions and properties of the real-world features being interacted with. Having defined how elements are merged into clusters and, consequently, how two clusters are merged (Section 5.1), we can now consider the similarity between two clusters.

5.2.1 Semantic Similarity

Clusters have tags that describe properties of the real-world entities contained in the cluster, forming an ideal basis for understanding what the user might have been doing. Under the assumption that clusters with similar tags are likely to have properties in common, we use the semantic similarity between cluster tags as the basis for a distance metric. For this, we adopt the similarity measure proposed by Wu and Palmer [1994], and extended by Resnik [1999] for calculating distance between word taxonomies through WordNet [Miller, 1995]. The calculated scores are between 0 and 1 (inclusive), where a score of 1 means that the words are interchangeable. The semantic similarity between two sets of tags, t_1 and t_2, is therefore calculated as:

$$TagSim(t_1, t_2) = \frac{\sum_{t \in t_1} argmax_{Sim}(Sim(t, t_{21}), Sim(t, t_{22}), ..., Sim(t, t_{2i}))}{|t_1|} \tag{3}$$

As tag similarity is not commutative, cluster similarity is calculated as:

$$SemanticSimilarity(c_1, c_2) = \\ argmax_{TagSim}(TagSim(c_1.tags, c_2.tags), TagSim(c_2.tags, c_1.tags)) \tag{4}$$

5.2.2 Feature Similarity

The context of an activity or period of time is dependent not only on the location in which time is spent, but on additional factors. With this in mind, we propose a second similarity measure, *FeatureSimilarity*, that compares the interaction features of two clusters, specifically:

- Average interaction duration
- Most common time of day interaction begins
- Count of the number of times the element is interacted with
- Total area covered by elements (in m^2)

The value from each feature is then discretised by placing values within bins (e.g. time of day could be recorded in 4 hour increments), and converted into a single string that describes the feature and value (e.g. 'timeofday_12' would indicate that the most common time of day that interaction begins is between 12PM–4PM). This procedure generates a set of features, f_1 and f_2, for clusters c_1 and c_2, from which a similarity score is defined using the Jaccard index [Rajaraman and Ullman, 2011]:

$$FeatureSimilarity(f_1, f_2) = \frac{|f_1 \cap f_2|}{|f_1 \cup f_2|} \tag{5}$$

5.2.3 Geographical Distance

For some applications it is possible that the similarity between clusters depends upon their geographical proximity, where two clusters that are close together may have common purposes. If this property is known to be true in the data, or given the goal of clustering, then the proximity of clusters can be considered as the minimum geographical distance between elements of a cluster, calculated using the Haversine formula [Robusto, 1957]:

$$GeographicalDistance(c_1, c_2) = argmin_{distance}(distance(x_1 \in c_1, y_1 \in c_2), \ldots) \tag{6}$$

5.2.4 Hybrid Contextual Distance

Using one of the previously discussed metrics in isolation would not accurately capture the context of the individual, as context depends on more than just any one factor. Instead, we combine the *SemanticSimilarity* and *FeatureSimilarity* scores into *Hybrid Contextual Distance (HCD)*, a measure of the contextual similarity between two clusters:

$$HCD(c_1, c_2) = \\ 1 - (\lambda \times SemanticSimilarity(c_1, c_2) + (1 - \lambda) \times FeatureSimilarity(c_1, c_2)) \tag{7}$$

where λ is a user-specified weighting parameter that allows emphasis to be placed either on the semantic or feature similarity between clusters. We choose to ignore the geographical proximity of elements

ALGORITHM 2 Agglomerative Hierarchical Clustering Algorithm

```
$clusters \leftarrow elements$ // The input set of $elements$, each treated as its own cluster
while $clusters$.length $> 1$ do

    // Create an $n \times n$ matrix of distances between clusters
    $distanceMatrix \leftarrow [\, [d_{11}, ...], [d_{21}, ...], ...]$

    // Find all pairs of clusters with the smallest distance between them
    // If multiple pairs overlap (i.e. share a cluster), then group them together
    $closestGroups \leftarrow$ ClosestGroups($distanceMatrix$)

    // Merge each extracted group into a single cluster
    for $group \in closestGroups$ do
        newCluster $\leftarrow$ Merge($group$)

        // Set the old clusters as children of the new and remove the old clusters from $clusters$
        for $cluster \in group$ do
            $newCluster$.children.append($cluster$)
            $clusters$.delete($cluster$)
        end for

        // Add the merged cluster to $clusters$
        $clusters$.append($newCluster$)
    end for

end while

// By this point, $clusters$ contains a single root cluster for the hierarchy
return $clusters$.first
```

because contexts should be separate from their geographical location (e.g. visiting two cafes in different cities is likely to be indicative of the same context). If, however, additional domain knowledge is available that ties geographical locations together with additional meaning (e.g. it is known that all buildings in a given area perform a similar function), then geographical distance could be added to the HCD metric. HCD can be used as a basis for clustering elements, and thus determining which elements have similar contexts, aiding in our understanding of the individual to which the data belongs.

5.3 Hierarchical Clustering

With a distance metric in place, clustering can be performed using standard techniques. While traditional clustering is limited in that it extracts clusters at a single scale, which may not be appropriate for a given task, hierarchical clustering identifies clusters at multiple scales. We use a greedy hierarchical agglomerative clustering algorithm, presented in Algorithm 2, that extracts clusters of increasing similarity up to a single root node, creating a *context tree*. While the hierarchical agglomerative clustering algorithm is fairly standard in itself, its application to the generation of context trees is novel. The algorithm deviates slightly from existing hierarchical clustering approaches in that it is capable of extracting multiple clusters together in a single step if they have the same distance.

6 Evaluation and Results

In this work it is not practical to obtain a concrete ground truth to act as a point of comparison for evaluation because the *correctness* of an extracted set of clusters depends on the task for which the clusters will be used. In light of this, we opt to evaluate the proposed techniques with an approach similar to those followed in existing literature where a single ground truth does not exist, as discussed in Section 2.2. This is achieved by exploring the properties of the generated context trees and comparing them against expected results while providing small, representative, examples that demonstrate the

```
(Step 1)
    latlng: 52.3834499, -1.56026223
    timestamp: 2013-11-08 14:09:51.000000000 Z
    accuracy: 65.0

(Step 2)
    latlng: 52.3834499, -1.56026223
    timestamp: 2013-11-08 14:09:51.000000000 Z
    accuracy: 65.0
    data: [n_312873295, n_552101208, n_695942926, n_1014585845, n_1014585853,
      w_92341980, w_92342116, w_145179860, w_145179863, w_145179883,
      w_273005393, w_303748830, w_329376738, w_329376739, r_2437023, ...]

(Step 3)
    latlng: 52.3834499, -1.56026223
    timestamp: 2013-11-08 14:09:51.000000000 Z
    accuracy: 65.0
    data: [w_145179860, r_2437023]

(Step 4)
    w_145179860:
      tags:
        building: university
        building_levels: 3
      members: [n_1586185863, n_1586185883, n_727382425, n_1586185856, ...]
      times:
        - begin: 2013-11-08 13:13:05.000000000
          end: 2013-11-08 17:16:47.000000000
      latlngs:
        - 52.3837765, -1.5601465
        - 52.3838285, -1.5600527
        - ...
    r_2437023:
      tags: ...
```

Figure 6: Examples of the data at each stage of the augmentation and filtering processes.

utility afforded by these procedures.

This section evaluates the proposed context tree data structure, along with the generation method proposed in Sections 4 and 5. Although there are many use-cases for context trees, including as a basis for anomaly detection, location prediction and city planning, we focus on understanding the high-level behaviours of an individual throughout a 24 hour period as a representative example.

Figure 6 shows sample data at each stage of the augmentation and filtering process. Raw trajectory data, in the form of an ordered array of points (Step 1) enters the system. Each point has timestamp, longitude, latitude and accuracy values. Step 2 augments the trajectory with identifiers for all land usage elements that the user could have been interacting with at that time (as described in Section 4). This is achieved by extracting all land usage elements within the radius of the accuracy of the point and storing the identifier of each element. Step 3 shows the augmented trajectory once filtered (as described in Section 4.2), which reduces the number of elements associated with each point, with the goal of limiting them to the elements likely being interacted with. Finally, summarisation occurs, clustering together contiguous time periods that belong to the same element (as described in Section 4.3), shown in Step 4.

Once a summarised dataset has been created, a context tree can be generated using the metrics and algorithm presented in Section 5. Generating a context tree from 24 hours of data produces a fairly large tree, an extract of which is shown in Figure 7. A more in-depth analysis of the clustering procedure is presented in Section 6.5.

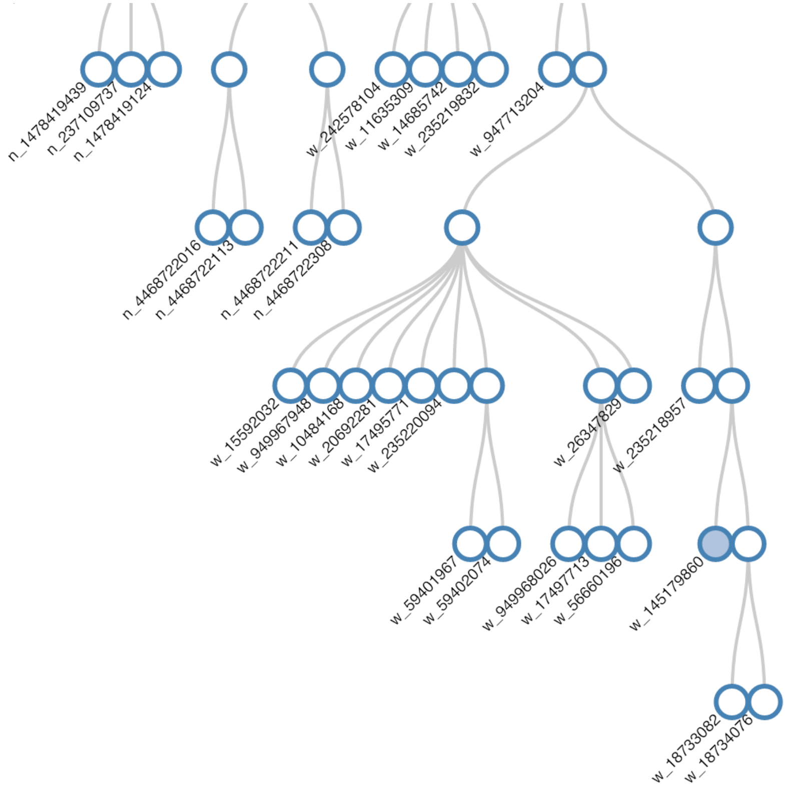


Figure 7: An extract of a tree generated using real data. The element used in the previous data examples is located in the bottom right of the image, clustered with other university buildings.

6.1 Data

Evaluating this work requires both geospatial trajectories and land usage information. The trajectories used are taken from the Nokia Mobile Data Challenge (MDC) Dataset [Kiukkonen et al., 2010; Laurila et al., 2012], as discussed in Section 2.1. From this dataset, we select the real-world data from 40 users with the largest number of trajectory points for this evaluation. While this dataset contains a vast amount of information, we only consider the timestamp, latitude, longitude and accuracy of each data point (consistent with the discussion in Section 4). In addition to this, and for comparative purposes, we also select 5 users from the GeoLife dataset [Zheng et al., 2008a, 2009, 2010] for evaluation. While the MDC dataset aims to provide continuous coordinates for the users, the GeoLife data instead only captures periods of times when the users were moving. It has the additional drawback of not including accuracy values, which are required for this work. As the data was collected using GPS-enabled devices, we opt to assume a constant accuracy of 10m for each coordinate, in line with the expected performance of GPS [Cao et al., 2009]. The trends presented in this section are consistent across both the MDC and GeoLife data, and so most of the GeoLife results are omitted for brevity, however an example can be found in Section 6.5.

A major drawback of using research datasets is that licences often prevent the publication of details that can be used to identify people or specific locations visited. Additionally, it is not possible to contact the users about whom data was collected to perform a user study. To get around these issues, we also collect a small dataset of our own. Aiming to match the methodology of the MDC data, trajectories were collected from the smartphones of 3 members of the Department of Computer Science, University of Warwick for a period of 3 days. These trajectories are used for illustration, instead of the MDC data, in Section 6.4 where the presented results contain the names of specific locations visited, and communication with the users was required.

Land usage information comes from OpenStreetMap (OSM)[1], a community-maintained map that

[1] https://openstreetmap.org/

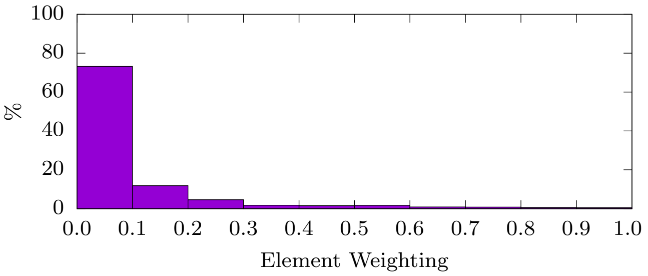
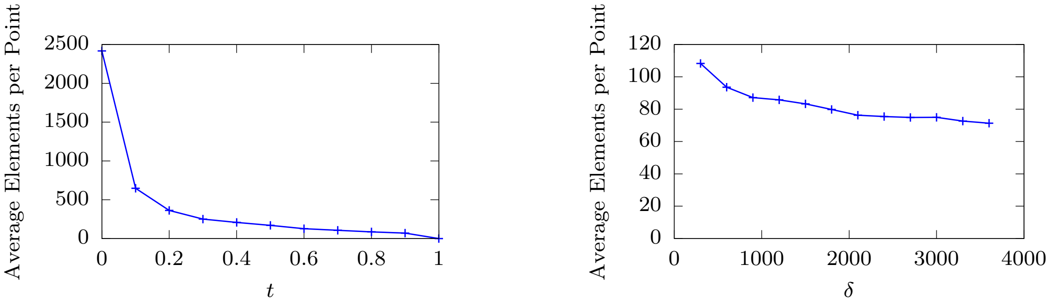


Figure 8: Distribution of element weights before filtering for an example user ($\delta = 1200$).



(a) t ($\delta = 1200$)

(b) δ ($t = 0.8$)

Figure 9: Effect of parameters on average number of elements per point post-filtering for an example user.

contains information pertaining to real-world entities, including their geographical coordinates and a set of tags that describe the entity. These entities include features such as individual items (e.g. a payphone or postbox), through to buildings and general land-usage designations (e.g. 'farmland'). The data is extremely detailed and accurate, spanning the entire world in a consistent manner, and thus forms an ideal basis for this work. The required elements are extracted from OSM through the Overpass API[2].

6.2 Filtering

The first stage in context tree generation is augmenting and filtering land usage elements. This section evaluates and characterises the performance of the filter on real-world data, by first exploring how element weights are distributed and then showing how this impacts the land usage elements that are filtered.

Filtering takes two parameters: δ and t. The parameter δ specifies the width of the buffer, in seconds, and t specifies a threshold where elements with a calculated weight of greater than t form the output set. Holding $\delta = 1200$, Figure 8 shows the distribution of weights for all elements in the filtering process (i.e. the values of *NormalisedScore* from Equation 2) for all 11,575 trajectory points belonging to a sample MDC user. The effects of t (with $\delta = 1200$) and δ (with $t = 0.8$) on the average number of elements per point post-filtering can be seen in Figures 9a and 9b, respectively. These results are consistent with expectations, as increasing t sets a higher threshold for elements to be included in the output set, and thus results in fewer elements. Increasing δ results in a greater time span considered by the filtering process, and so more elements are considered as transient, and are thus removed. Each of these figures is generated from 7 months of data from a single sample user, and while the exact numbers vary when using data from different users, the trends remain consistent across users from both datasets.

The accuracy of the trajectory points determines the radius of land usage data to consider. The effect of accuracy on the number of extracted elements, both pre- and post-filtering, is shown in Figure 10 for each of the 40 MDC users. The figure demonstrates that a larger accuracy typically results in a larger number of elements per point, and that filtering reduces this number.

[2] `https://wiki.openstreetmap.org/wiki/Overpass_API` — By default, the Overpass API is only capable of extracting elements that a coordinate pair is contained within if the element has been assigned a name. The API has therefore been modified to consider all enclosing elements in these cases.

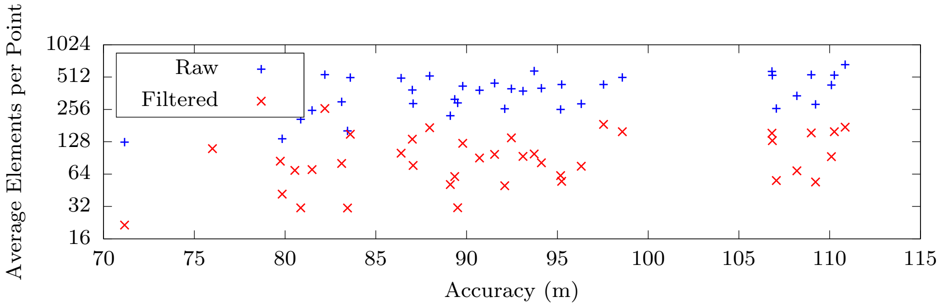


Figure 10: Effect of accuracy on number of elements, pre- and post-filtering, for different users' data.

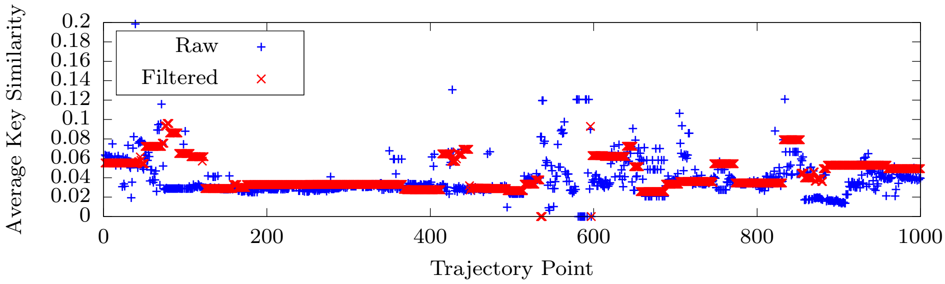


Figure 11: Effect of filtering on tag key similarity, both pre- and post-filtering.

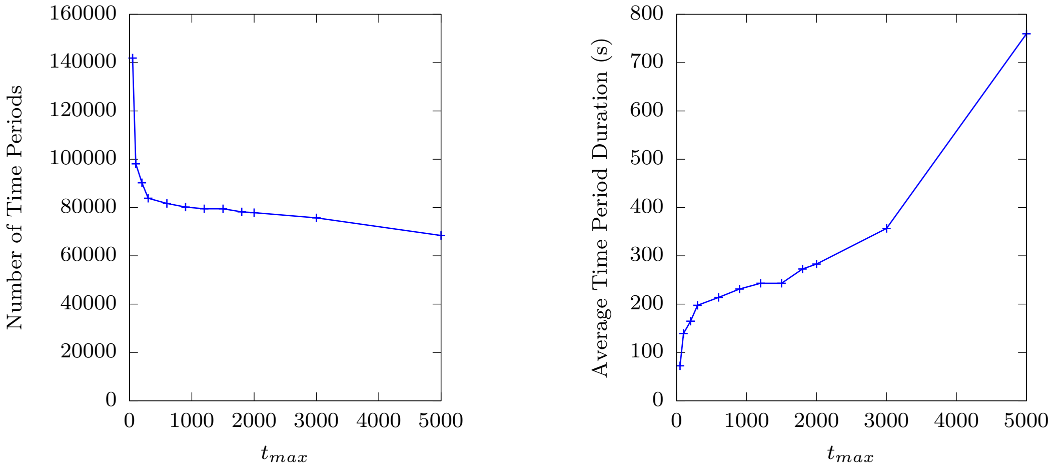
6.2.1 Filtering Characterisation

To better understand the filtering process, we explore properties of the filtered data, specifically focusing on how the elements and their semantics change. The aim of filtering is to remove noise and focus the data on elements that the user was likely interacting with at a given time. It is reasonable therefore to assume that the elements post-filtering should have more similarity than those before, with less variation caused by the inclusion of *random* elements. To explore this hypothesis, Figure 11 shows the average tag key similarity (i.e. only the key part of the 'key:value' pair that makes up an element's tags, which corresponds to broad type, e.g. 'building') both pre- and post-filtering for a given user over 1000 points of their data. This demonstrates that in the majority of cases, tag key similarity is increased, and variance significantly reduced, after filtering has occurred, indicating that the elements present post-filtering are more similar and that unrelated noise elements have been correctly removed. The semantic similarity of these tags is calculated using the method proposed in Section 5.2.1.

6.3 Summarising Data

Once the data has been filtered, it is summarised into continuous periods of time. Only one parameter, t_{max}, is required, specifying the maximum amount of time (in seconds) between consecutive points for them to be considered contiguous. Using the parameters $\delta = 1200$ and $t = 0.8$, Figure 12a shows how t_{max} affects the number of such periods extracted, and Figure 12b shows how t_{max} affects the average length of such periods.

Figures 12a and 12b show that increasing t_{max} causes fewer, but longer, time periods to be extracted. In the remainder of this paper, we use $t_{max} = 1200$ (i.e. 20 minutes) as it provides enough time for a user to have ceased interacting with an element and to have later recommenced interaction, without causing too many interactions to be needlessly split.



(a) Number of time periods

(b) Average time period duration

Figure 12: Effect of t_{max} on the summarising procedure.

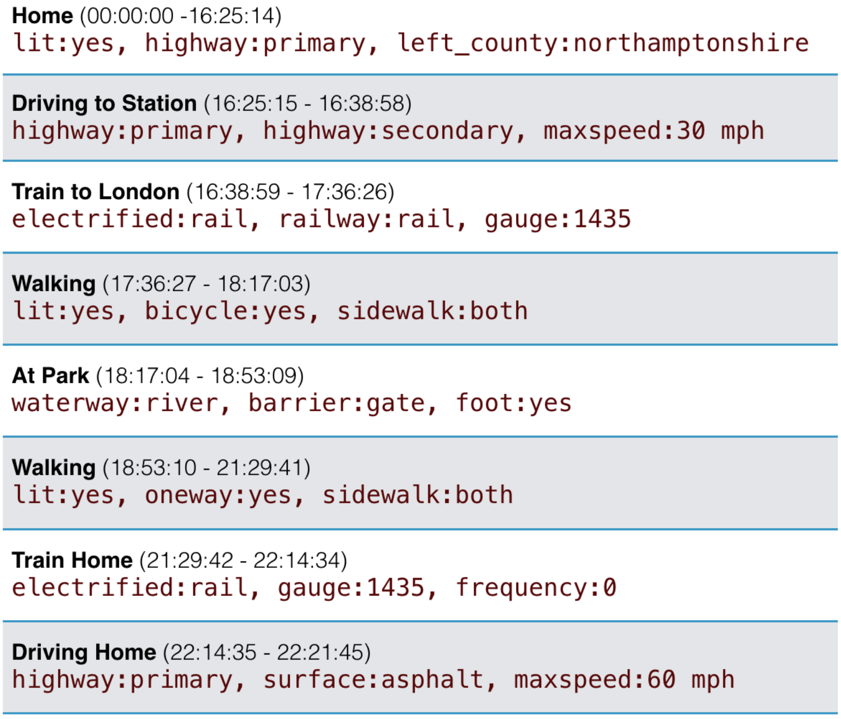
6.4 User-informed Evaluation

While there is no ground truth available for this type of problem, we can evaluate the procedure by considering desirable properties of the output for a specific application and manually compare the expected and actual results for small subsets of data. A major problem here is that when conducting such evaluations over publicly available research datasets, such as the MDC or GeoLife datasets used in this paper, or alternatives such as Yonsei, there is no mechanism for contacting users to have them verify assumptions. To overcome this problem, we opt to use data collected ourselves for this part of the evaluation, as it affords us the ability to discuss with users exactly what activities they were conducting on a given day. Details of the data collected can be found in Section 6.1.

This section presents analyses on small amounts of manually labelled real-world data with the goal of using the constructed context trees to provide meaning to high-level behaviours, with the overall aim of identifying such behaviours from the tree. The data analysed spans 24 hours from the three users of the Warwick dataset, where annotations were added manually as accurately as possible, and in consultation with the users. The augmentation and filtering procedures were run over this data and, for each labelled time period, the 3 most common element tags were identified. This is shown in Figures 13-15. The aim here is not to label the time periods with the exact activity being performed, but rather to demonstrate that a meaningful relationship exists between the tags extracted and the true activity.

In Figure 13, general labels are applied to the activities being performed, and a meaningful correlation between the tags extracted by the procedure and these labels is evident. Specific examples include the action of driving being labelled with the 'highway' key, and taking the train with 'railway'. Although the tags are not always perfect, they are indicative. For instance, when the individual was at home no residential building was identified, but an indication of the type of location was given by the tags 'lit:yes' and 'highway'. In the region where this data was collected, roads with street lighting typically signify residential areas. A similar relationship is shown in Figures 14 and 15, with labels applied hierarchically and at lower granularities. While not every item is labelled exactly, we believe this is a result of the data collection method. We used a data collection rate of one point per minute, meaning that several labelled activities consist of only 1 or 2 trajectory points, leaving little information for the procedure to utilise. Similarly, the land usage dataset contains a vast amount of information, but can be limited in parts. An example of this is that the pub which was visited at 17:25 (Figure 14) is inside a larger building. The procedure is only capable of identifying that the building was occupied by the user, but there is no information pertaining to which element inside the building was being interacted with, and so the only available information is 'building:yes'.

To quantitatively explore how well the procedure worked over these examples, each tag extracted is scored based on the relevancy to the label using three classifications: *high*, *medium*, *low/none*. These scores are manually assigned and shown in Table 1, where a *high* tag indicates that the label is very



Home (00:00:00 -16:25:14)
`lit:yes, highway:primary, left_county:northamptonshire`

Driving to Station (16:25:15 - 16:38:58)
`highway:primary, highway:secondary, maxspeed:30 mph`

Train to London (16:38:59 - 17:36:26)
`electrified:rail, railway:rail, gauge:1435`

Walking (17:36:27 - 18:17:03)
`lit:yes, bicycle:yes, sidewalk:both`

At Park (18:17:04 - 18:53:09)
`waterway:river, barrier:gate, foot:yes`

Walking (18:53:10 - 21:29:41)
`lit:yes, oneway:yes, sidewalk:both`

Train Home (21:29:42 - 22:14:34)
`electrified:rail, gauge:1435, frequency:0`

Driving Home (22:14:35 - 22:21:45)
`highway:primary, surface:asphalt, maxspeed:60 mph`

Figure 13: Manually labelled data (in bold) compared against extracted element labels.

well correlated to the activity (e.g. 'building:residential' to the activity 'Home'), *medium* indicates that there is some link (e.g. 'surface:asphalt' to 'Driving on a main road'), and *low/none* being given to tags with little or no relationship to the activity (e.g. 'highway:bus_stop' to 'Attending lecture'). Figure 16a shows the proportion of tags assigned to each of these weightings, demonstrating that the procedure identified tags with *high* or *medium* relevancy 69.7% of the time. We also consider the highest-ranked tag assigned to each labelled time period and the proportion of time periods represented by each tag score is shown in Figure 16b. From these results, it is clear that while in the three examples, only 32.8% of tags were awarded a *high* relevancy score, 60.0% of labels have at least one tag with such a score, and 88.9% contain at least one tag with a score of *high* or *medium*. This indicates that while not all of the 3 tags per label were useful, in nearly all cases, at least one of them was.

While this evaluation is limited in that it only considers 3 days worth of data from 3 different users, it provides an indication that the techniques discussed previously are extracting useful and correct elements. This is demonstrated by showing that there is a strong relationship between the tags identified by the system and the labels manually assigned to data as a partial ground truth. A complete ground truth is not possible in this domain, since the desirable properties of context trees will vary significantly based on their intended use, however we believe that this exploration goes some way to demonstrating the accuracy of the technique.

Table 1: Summary of tags and frequency count for each type of interactions scored based on the relevancy of each tag (High, Medium and Low/None).

Label	**Tag**	**S**	**#**	**Tag**	**S**	**#**
Home	landuse:residential	H	2	barrier:kissing_gate	L	1
	highway:residential	H	2	oneway:no	L	1
	building:residential	H	1	maxspeed:30	L	1
	building:garage	M	1	highway:primary	L	1
	lit:yes	M	1	left_county:nor...	L	1
Walking (res.)	landuse:residential	H	1			
Walking (shops)	amenity:parking	M	1			
Walking (road)	sidewalk:both	H	2	highway:bus_stop	M	1
	highway:secondary	H	1	bicycle:yes	M	1
	oneway:yes	M	2	ref:lmngtns	L	1
	lit:yes	M	2	public_transport:pay...	L	1
	boundary:public...	L	1			
Walking (park)	leisure:park	H	1	waterway:river	M	1
	foot:yes	H	1	barrier:gate	M	1
	barrier:kissing_gate	M	1			
Driving (res.)	landuse:residential	H	2			
Driving (road)	highway:tertiary	H	6	maxspeed:60	M	3
	highway:primary	H	2	maxspeed:30	M	2
	highway:secondary	H	1	maxspeed:20	M	2
	oneway:yes	M	4	amenity:university	L	2
	highway:bus_stop	M	3	type:multipolygon	L	1
	surface:asphalt	M	3			
Parking (uni)	amenity:university	M	1	type:multipolygon	L	1
Work (office)	building:university	H	2	highway:footway	L	1
	building_levels:4	M	2	highway:service	L	1
Walking (uni)	amenity:university	H	4	landuse:grass	M	1
	highway:crossing	H	2	type:multipolygon	L	4
Eating (rest.)	level:0	M	1	area:yes	L	1
	level:1	M	1	lit:yes	L	1
	building:yes	M	1	surface:asphalt	L	1
Eating (pub)	building:yes	M	1	area:yes	L	1
	level:0	M	1			
Work (library)	amenity:library	H	2	type:multipolygon	L	2
	amenity:university	M	2			
Work (lecture)	surface:asphalt	L	2	highway:bus_stop	L	1
	type:multipolygon	L	1	oneway:yes	L	1
	lit:yes	L	1			
Visiting friend	amenity:university	M	1	type:multipolygon	L	1
	building:yes	M	1			
Petrol station	operator:tesco	H	1	amenity:fuel	H	1
	opening_hours:24/7	H	1			
Union (uni)	amenity:university	M	1	type:multipolygon	L	1
Bar	building:yes	M	2	oneway:yes	L	2
	surface:asphalt	L	2			
Train	electrified:rail	H	2	railway:rail	H	1
	gauge:1435	H	2	frequency:0	L	1

Home (00:00:00 - 08:58:29) `landuse:residential, building:residential, building:garage`		
Walking Dog (08:58:30 - 09:15:18) `highway:bus_stop, oneway:yes, highway:secondary`	**Walking along residential roads** (08:58:30 - 09:04:47) `landuse:residential`	
	Walking along main road (09:04:58 - 09:07:57) `highway:bus_stop, oneway:yes, highway:secondary`	
	Walking through shopping area (09:07:58 - 09:10:04) `amenity:parking`	
	Walking along footpath, through park (09:10:05 - 09:13:12) `barrier:kissing_gate, leisure:park`	
	Walking along residential roads (09:13:13 - 09:15:18) `landuse:residential, barrier:kissing_gate`	
Home (09:15:19 - 09:35:15) `landuse:residential, highway:residential, barrier:kissing_gate`		
Travel to Work (09:35:16 - 09:47:54) `maxspeed:20 mph, highway:tertiary, oneway:yes`	**Drive to work** (09:35:16 - 09:43:40) `maxspeed:20 mph, highway:tertiary, oneway:yes`	**Driving along residential road** (09:35:16 - 09:37:21) `landuse:residential`
		Driving along main road (09:37:22 - 09:43:40) `maxspeed:20 mph, highway:tertiary, oneway:yes`
	Parking in car park (09:43:41 - 09:46:50) `type:multipolygon, amenity:university`	
	Walk to building (09:46:51 - 09:47:54)	
Work (09:47:55 - 17:15:59) `highway:footway, building:yes, highway:service`	**In office** (09:47:55 - 12:09:55) `highway:footway, building_levels:4, building:university`	
	Lunch (12:09:56 - 13:13:04) `level:0, level:1, area:yes`	**Walking across campus** (12:09:56 - 12:18:20) `type:multipolygon, amenity:university`
		Eating at restaurant (12:18:21 - 13:04:36) `level:0,level:1,area:yes`
		Walking across campus (13:04:37 - 13:13:04) `type:multipolygon, amenity:university, highway:crossing`
	In office (13:13:05 - 17:15:59) `highway:footway, building_levels:4, highway:service`	
Evening (17:16:00 - 23:59:59) `building:yes, level:0, area:yes`	**Walking across campus** (17:16:00 - 17:25:28) `type:multipolygon, amenity:university, highway:crossing`	
	Eating at pub (17:25:29 - 23:36:48) `building:yes, level:0, area:yes`	
	Walking across campus (23:36:49 - 23:49:25) `type:multipolygon, amenity:university, landuse:grass`	
	Driving home (23:49:26 - 23:59:59) `highway:bus_stop, oneway:yes, highway:tertiary`	**Driving along main road** (23:49:26 - 23:54:39) `highway:bus_stop, oneway:yes, highway:tertiary`
		Dropping off passenger in residential area (23:54:40 - 23:57:48) `landuse:residential`
		Driving along main road (23:57:49 - 23:59:59) `highway:bus_stop, oneway:yes, maxspeed:20 mph`

Figure 14: Manually labelled data (in bold) compared against extracted element labels.

Home (00:00:00 - 07:13:32) `oneway:no, maxspeed:30 mph, highway:residential`	
Driving to University (07:13:33 - 07:44:27) `highway:tertiary, surface:asphalt, maxspeed:60 mph`	

On Campus
(07:44:28 - 15:02:18)
`type:multipolygon,`
`amenity:university,`
`building:yes`
	In library (07:44:28 - 08:05:05)
	`type:multipolygon, amenity:university, amenity:library`

	Lecture (08:05:06 - 09:01:15) `highway:bus_stop, surface:asphalt, lit:yes`
	In Library (09:01:16 - 13:02:58) `type:multipolygon, amenity:university, amenity:library`
	Lunch at Friend's accommodation (13:02:59 - 15:02:18) `type:multipolygon, amenity:university, building:yes`

Trip to Local Store
(15:02:19 - 15:22:25)
`amenity:university,`
`operator:tesco,`
`amenity:fuel`
	Drive to shop (15:02:19 - 15:10:02)
	`amenity:university, maxspeed:30 mph, highway:tertiary`

	At petrol station (15:10:03 - 15:17:59) `operator:tesco, opening_hours:24/7, amenity:fuel`
	Drive to campus (15:18:00 - 15:22:25) `type:multipolygon, amenity:university`

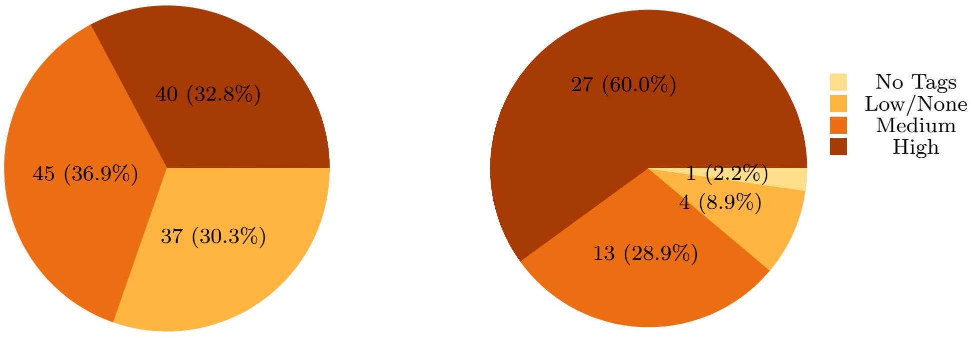
On Campus
(15:22:26 - 18:06:58)
`highway:bus_stop,`
`oneway:yes,`
`surface:asphalt`
	In union building (15:22:26 - 15:48:59)
	`type:multipolygon, amenity:university`

	Lecture (15:49:00 - 18:06:58) `highway:bus_stop, oneway:yes, surface:asphalt`
Driving Home (18:06:59 - 18:38:06) `highway:tertiary, surface:asphalt, maxspeed:60 mph`	

Evening
(18:38:07 - 23:20:00)
`surface:asphalt,`
`lit:yes,`
`oneway:no`
	Walking to town (18:38:07 - 19:01:37)
	`ref:lmngtns, public_transport:pay_scale_area, boundary:public_transport`

	Eating at bar/restaurant (19:01:38 - 20:47:46) `surface:asphalt, lit:yes, building:yes`
	At bar (20:47:47 - 21:39:22) `surface:asphalt, oneway:no, building:yes`
	At bar (21:39:23 - 23:20:00) `surface:asphalt, oneway:no, building:yes`

Figure 15: Manually labelled data (in bold) compared against extracted element labels.



(a) Overall tag relevancy

(b) Best tag relevancy per time period

Figure 16: Proportion of relevant tags.

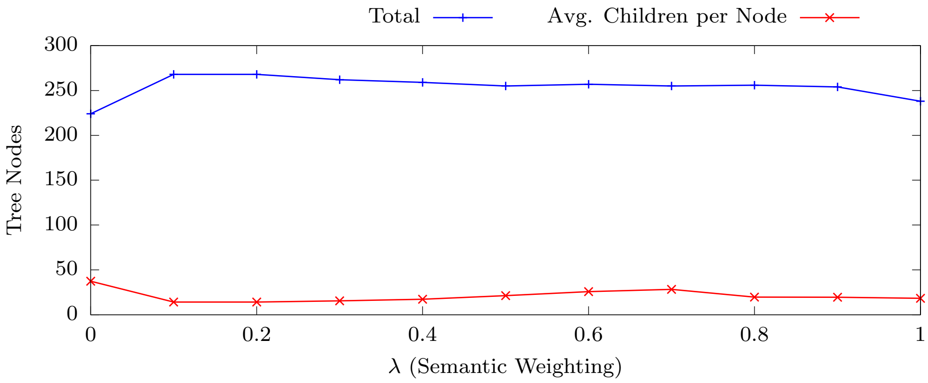


Figure 17: Relationship between λ and number of tree nodes.

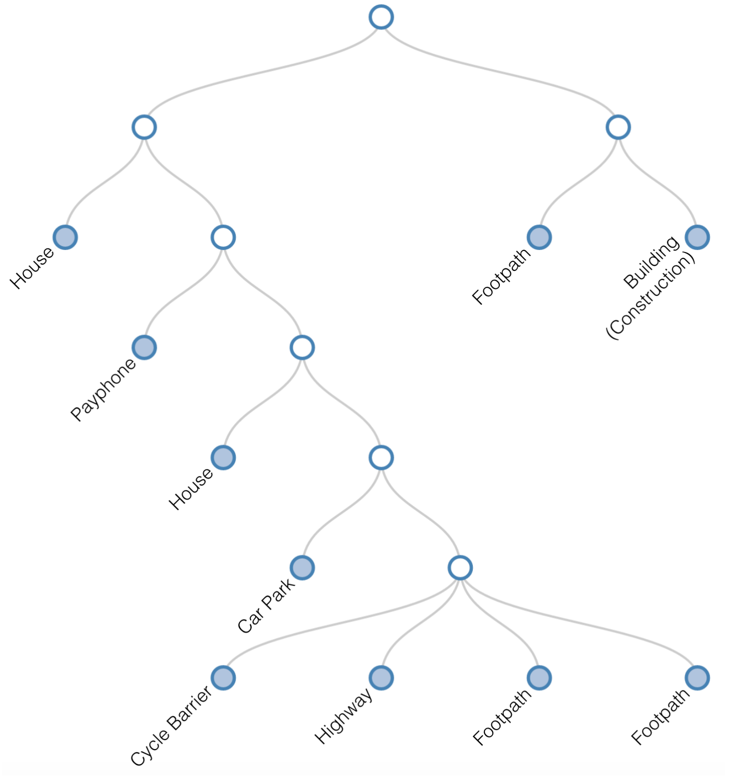


Figure 18: Example context tree: geographic clustering.

6.5 Context Trees

When constructing context trees from summarised data (Section 5), the only required parameter is λ, which specifies the weighting to be given to *semantic similarity* as part of the *Hybrid Contextual Distance* distance metric (Equation 7). A weighting of 1 will construct a tree based only on the semantic similarity between node tags, and a weighting of 0 will construct a tree based only on the similarity of features, with any value in between using a combination of the two. The relationship between λ and the number of nodes in a context tree is shown in Figure 17 (generated using 24 hours of a single users' data, filtered with parameters $\delta = 1200$, $t = 0.8$, and $t_{max} = 1200$). While the number of nodes does not vary drastically with λ, the meaning behind the clusters does.

Since our work on understanding context from trajectories augmented with land usage information is novel, there are no existing baseline methods or ground truth datasets to compare against. Instead, we take the closest method to a baseline that exists and compare the results against this. Figures 18 and 19 show the results of clustering context trees using naïve distance metrics that consider only geographic

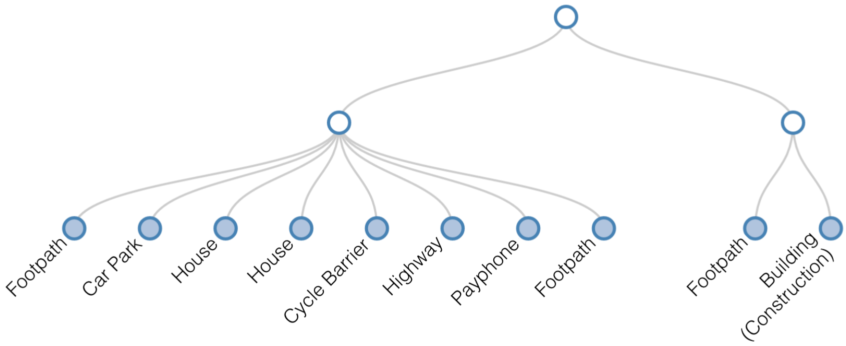


Figure 19: Example context tree: temporal clustering.

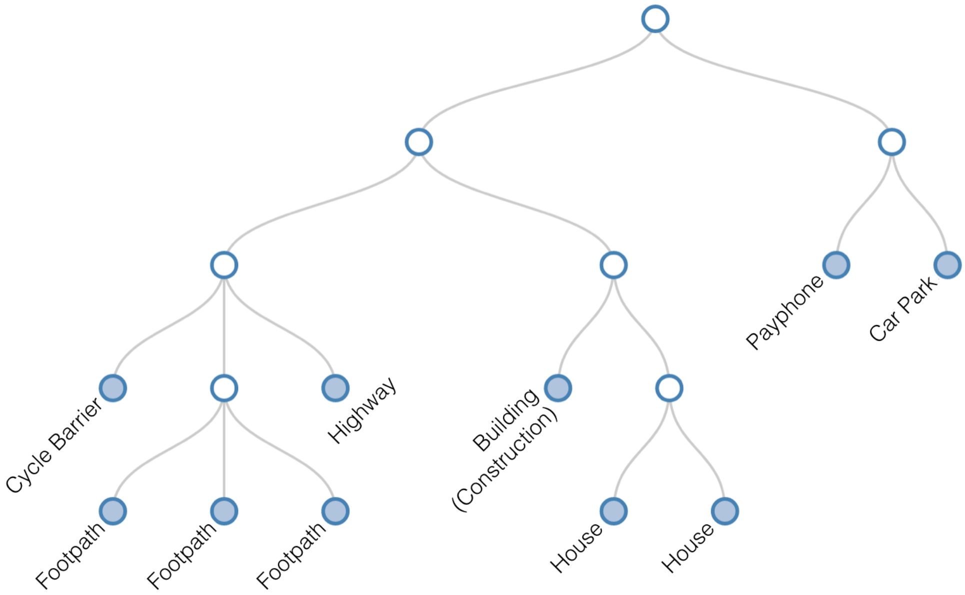


Figure 20: Example context tree: semantic clustering ($\lambda = 1$).

distance between elements (Figure 18) and temporal distance between interactions (Figure 19). While these figures only show one small example, the results are representative of using such metrics in that the elements clustered together have no clear contextual relationship. This is in contrast to the context trees generated from the same data using the Hybrid Contextual Distance metric, along with different values of λ, as shown in Figures 20–22.

In all of these examples, the element identifier has been manually replaced with a descriptive keyword to represent the element. Semantic clustering (Figure 20) creates distinctive groups for buildings, footpaths and public amenities, as the elements in these groups are similar, while feature-based clustering (Figure 21) creates groups that are less easily identifiable and relate to properties of the elements (e.g. the footpaths are not grouped because they were not encountered in the same journey, but rather were used at different times of the day). Finally, hybrid clustering (Figure 22) shows properties of both semantic and feature-based clustering where both the description of the element and properties of the interaction with the element are considered to create clusters. Selecting an appropriate value of λ is application-specific.

These context trees provide only small examples of the differences between trees generated with naïve distance metrics (Figures 18 and 19) and those generated with the HCD metric (Figures 20–22). In order to quantify such differences, and given knowledge of the data and how it was collected, we opt to make several assumptions of expected properties of generated context trees and explore the extent to which these expectations are violated with each distance metric. While this is of course a subjective evaluation, and the utility will vary based on the specific application the context tree is put to, it goes some way to

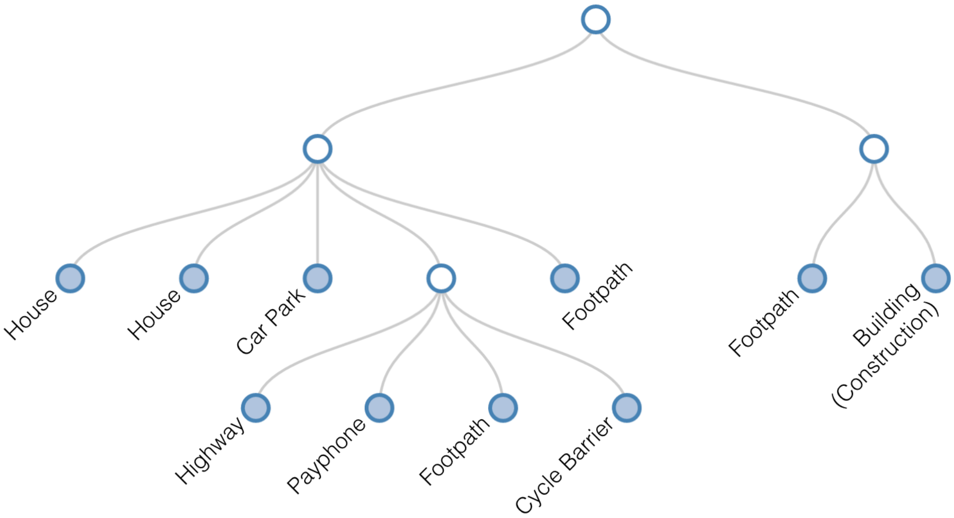


Figure 21: Example context tree: feature-based clustering ($\lambda = 0$).

providing an indicator of the utility of this approach in lieu of a ground truth. The assumptions made are:

1. Buildings should be grouped together unless they have very different uses (e.g. residential buildings should not be in the same group as office buildings).
2. Roads should be grouped together, with elements relating to roads grouped at a higher level (e.g. junctions).
3. Public amenities should be grouped together unless the interactions have very different properties.

These assumptions focus on the semantics of elements, but the features also need to be considered when exploring possible reasons for clusters being split. For instance, if a person visited many houses as part of their job, it would be reasonable to assume that these houses should be semantically close to the residence of the individual in the context tree, but not at exactly the same level. The usefulness of such assumptions will depend on the application, but it is possible to see that when aiming to characterise how a person has spent their time, it is beneficial to identify the times spent at residential buildings separately to those spent at work. On the small example context trees shown in this section, geographic and temporal clustering (Figures 18 and 19) violate all 3 assumptions. Semantic clustering (Figure 20) best adheres to these assumptions, with the houses grouped at the same level and the building under construction close by in the next level up. Similarly, the footpaths are together with the cycle barrier, a related element, and highway one level up. Feature-based clustering (Figure 21) has fewer valid assumptions than semantic clustering, as it only considers the interactions with the elements and not the elements themselves. Although the houses are together in a single cluster, they are also joined with the car park and footpath. Finally, hybrid clustering (Figure 22) is very similar to semantic clustering with the exception that the highway is no longer situated close to the footpaths, but is further up the context tree by itself. This still leaves 2 of the assumptions strictly adhered to, with 1 very close. A change that can be explained by the consideration of interaction features, where the highway has a different profile of interaction than the footpath and cycle barrier elements. Again, these are small examples, however the trends present have been observed to be consistent across larger context trees.

With a better understanding of filtering, summarising and clustering, we turn our attention to exploring how data influences the properties of the generated context tree. Focusing on 21 days of data from a single user, Figure 23 shows repetition in data by using the first day as a set of *training* data and calculating the coverage (i.e. the proportion of *test* data present in the *training* data) for each following day, shown by the blue *Fixed* line. Additionally, the red *Retrained* line shows the coverage when using all previous days (i.e. 0 to $n-1$, where n is the current day) as the *training* set. The total number of nodes, number of leaf nodes, and total count of time periods for a context tree generated using the same data

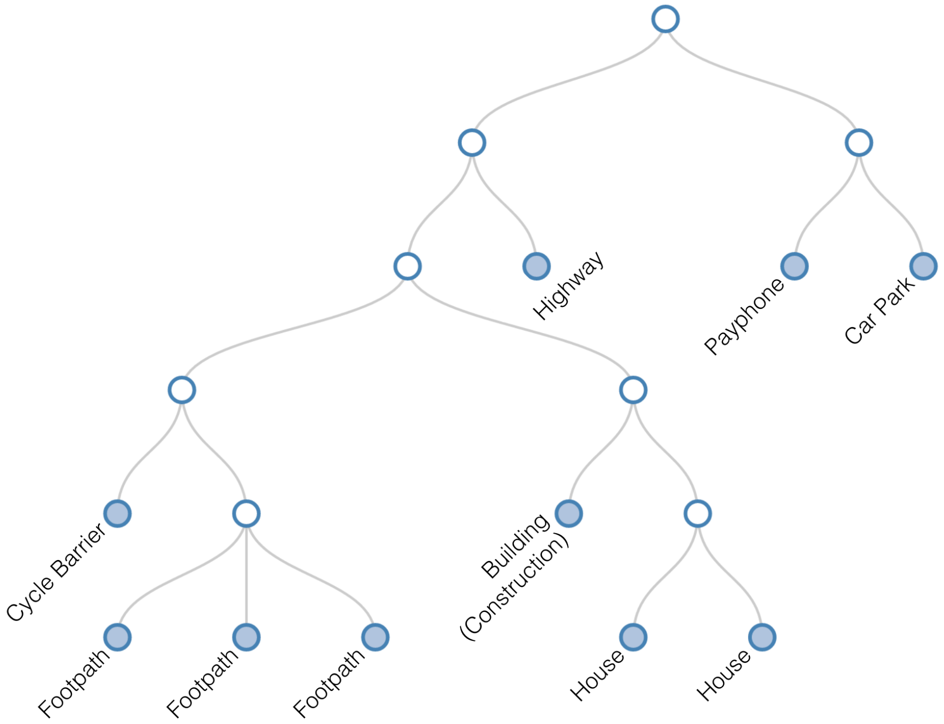


Figure 22: Example context tree: hybrid clustering ($\lambda = 0.6$).

(where day n shows a summary for a tree built using all data from days 0 to n) are shown in Figure 24a. Please note that no data was recorded during day 5 for this sample user in the MDC dataset.

Figure 23 begins with a low coverage for both *Fixed* and *Retrained* lines, indicating that few elements encountered in day 1 were present in the training set (day 0). However, while the *Fixed* score remains low for days 2–4, the *Retrained* score approaches 100%. In this instance, this is indicative of the user visiting elements that they did not encounter in the initial training day (day 0), but that they did encounter during subsequent days, as the *Retrained* line includes all previous days as training data. The figure shows similar results for the remaining test days, where during day 9 the user visited only locations visited during day 0 and during days 9, 11 and 16–20 the user encountered no new elements as the score for *Retrained* is at 100%. Figure 24a shows how these properties relate to the size of context trees generated. The number of *leaf nodes* is the number of unique elements and the number of *time periods* is a count of the total number of (non-unique) elements encountered. That is, if the user encountered the same element 3 times, or 3 different elements, both would count as 3 time periods. At day 1 the number of time periods is roughly the same as the number of leaf nodes, indicating that all elements were encountered approximately once. As time goes by, more elements are encountered, but a large number of existing elements are revisited, demonstrated by the disproportionate rise in the number of time periods. This indicates that over a short period, where the user likely remained within a single region, the size of the tree does not increase significantly as additional data is added. However, considering trees over larger time periods will not have the same property as the user will likely visit new regions with entirely new leaf nodes. Figure 24b shows a similar graph as Figure 24a, however it was generated using data from a user of the GeoLife dataset instead of the MDC dataset. As is evidenced by the figures, the procedure extracts similar trends in users from each dataset.

This section has characterised the outputs and properties of the context tree generation procedure presented in Section 3. While the concept of a ground truth for this work is not applicable, and existing approaches for comparison are lacking, through the provision of multiple small examples and a discussion of general trends we have demonstrated the applicability of the approach presented in this paper to the task of identifying similar contexts and storing such information in a hierarchical data structure.

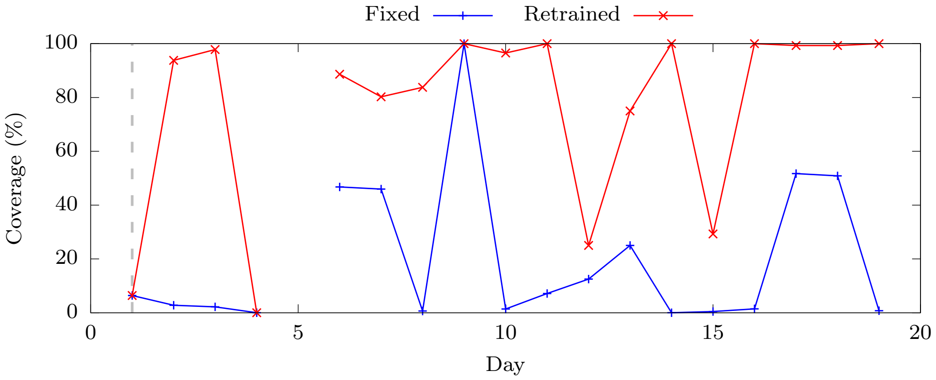
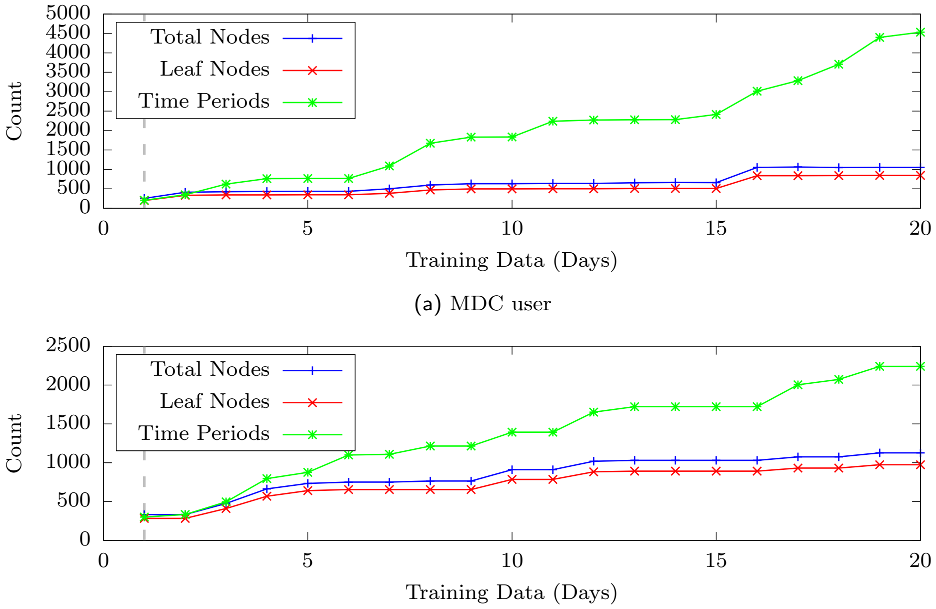


Figure 23: Land usage coverage with an initial training period of 24 hours (indicated by the dotted line).



(a) MDC user

(b) GeoLife user

Figure 24: Training data against number of tree nodes.

7 Context Tree Pruning

Storing context trees in their entirety maintains the maximum amount of information, however there are applications where reducing the size of a tree may be desirable. Memory-constrained devices, for example, may be better able to make use of a reduced size context tree as this would require lower storage requirements, and also enable quicker search due to the reduced number of nodes. Furthermore, reducing the size of context trees may have application-specific benefits, such as preventing overfitting when learning prediction models. In both of these cases, it is desirable to *prune* the tree to reduce the amount of data stored while maintaining as much information as possible. This section presents a method for such pruning, that although requires additional processing to select nodes eligible to be removed, results in smaller context trees that require less memory to store and fewer operations to search. A representation of a pruned context tree can be seen in Figure 25.

7.1 Pruning Criteria

Pruning is performed depth-first, evaluating each cluster to determine whether the additional overhead of storing the node is outweighed by the utility it affords. Each cluster is considered using the *null*

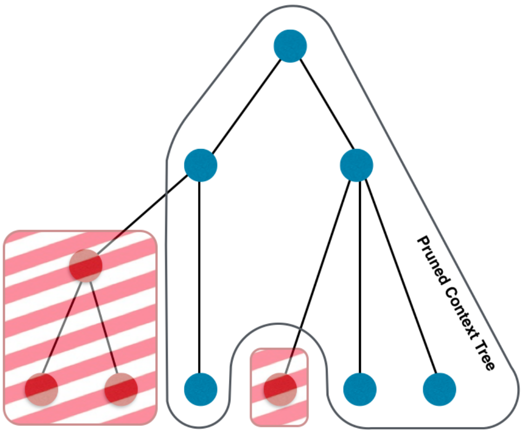


Figure 25: An example of a pruned context tree (with removed nodes crossed through).

hypothesis, and the hypothesis rejected when the utility of storing the cluster is above a threshold. Any cluster for which we are unable to reject the hypothesis is pruned, and its parent is marked as eligible for pruning. As metrics do not already exist for this task, we adapt existing metrics used in related domains for the purpose of context tree pruning.

7.1.1 Storage Cost

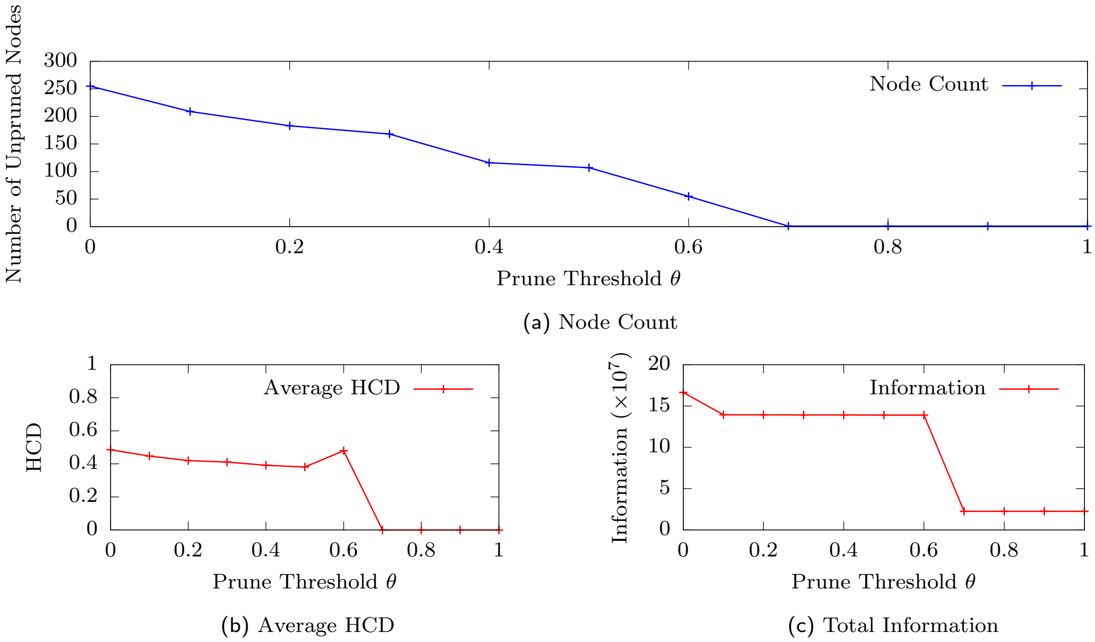
Clusters are scored according to two metrics: their storage cost and their utility. To determine the cost of storing a cluster, it is important to understand how clusters are built up in a context tree (described in Section 5.1). When merging two clusters together to form a parent cluster, the aspects that belong to each cluster are considered in turn; specifically the *tags*, *times* and *coordinate sets*. Sets of tags are combined from the child clusters by taking their union, while times and coordinate sets are merged in such a way that overlapping components are combined into single elements, and thus through the combining of child clusters into a parent cluster, information has been removed. The cost of storing an additional node is therefore the cost of storing the individual components (e.g. time range) that are present in a child, but not present in the same form in its parent. Assuming uniform cost for each component:

$$Cost(C|P) = \\ \xi + |C_{times} \setminus P_{times}| + |C_{coordsets} \setminus P_{coordsets}| + |\cup_{s \in C_{coordsets}} s \setminus \cup_{s \in P_{coordsets}} s| \quad (8)$$

Where $\xi > 0$ is a small, manually selected, penalty that represents the overhead of storing each cluster, C_{times} is the set of time ranges that are associated with cluster C and $C_{coordsets}$ is the set of coordinate sets associated with cluster C. Remembering that the coordinate sets belonging to a cluster themselves contain sets of points (i.e. $C_{coordsets} = \{\{p_{1:1}, p_{1:2}, p_{1:3}, ...\}, \{p_{2:1}, p_{2:2}, p_{2:3}, ...\}, ...\}$), $\cup_{s \in C_{coordsets}} s$ is taken to be the set of all points associated with any coordinate set that belongs to cluster C. Having ξ as non-zero represents that there is always a (small) cost associated with each cluster. Equation 8 will need tuning based on the specific application to better represent the true cost of storing a node, but it provides a basic foundation.

7.1.2 Cluster Utility

Determining the utility of a cluster is difficult and is dependent on the specific use of the context tree. For this reason, any application of the approach will need to consider the goal of pruning and use this to inform the measurement of the utility afforded by a specific cluster. We adopt a general approach that can be tailored to specific needs by providing a measure of the information lost if the parent of a cluster were used to represent the child, similar in idea to the *Kullback-Leibler divergence* used to measure the difference between probability distributions. As parents contain a superset of the children, we consider the utility of a child cluster (C) given its parent (P) to be the proportion of information present in the parent that is not covered by the child, where the measure of information must consider the attributes



(a) Node Count

(b) Average HCD

(c) Total Information

Figure 26: Effect of θ on number of nodes in a sample context tree ($\lambda = 0.5, \xi = 1$).

(i.e. tags, times, and coordinate sets) present in each cluster:

$$Information(C) = \sum_{t \in C_{times}} duration(t) + \sum_{s \in C_{coordsets}} area(s) + |C_{tags}| \tag{9}$$

Providing even weighting to the different elements for the measure of utility:

$$Utility(C|P) = 1 - \left(\frac{1}{3} \frac{\sum_{t \in C_{times}} duration(t)}{\sum_{t \in P_{times}} duration(t)} + \frac{1}{3} \frac{\sum_{s \in C_{coordsets}} area(s)}{\sum_{s \in P_{coordsets}} area(s)} + \frac{1}{3} \frac{|C_{tags}|}{|P_{tags}|} \right) \tag{10}$$

Specifically, this metric considers the proportion of time, area and tags covered by the child with respect to the parent, and holds true to the aims of such a metric to produce a score of 0 if the parent and child contain identical information and a score approaching 1 if the child only represents a fraction of the parent.

7.1.3 Cost-Benefit Score

The *cost-benefit score* of a cluster is taken to be the utility of the cluster divided by the storage cost:

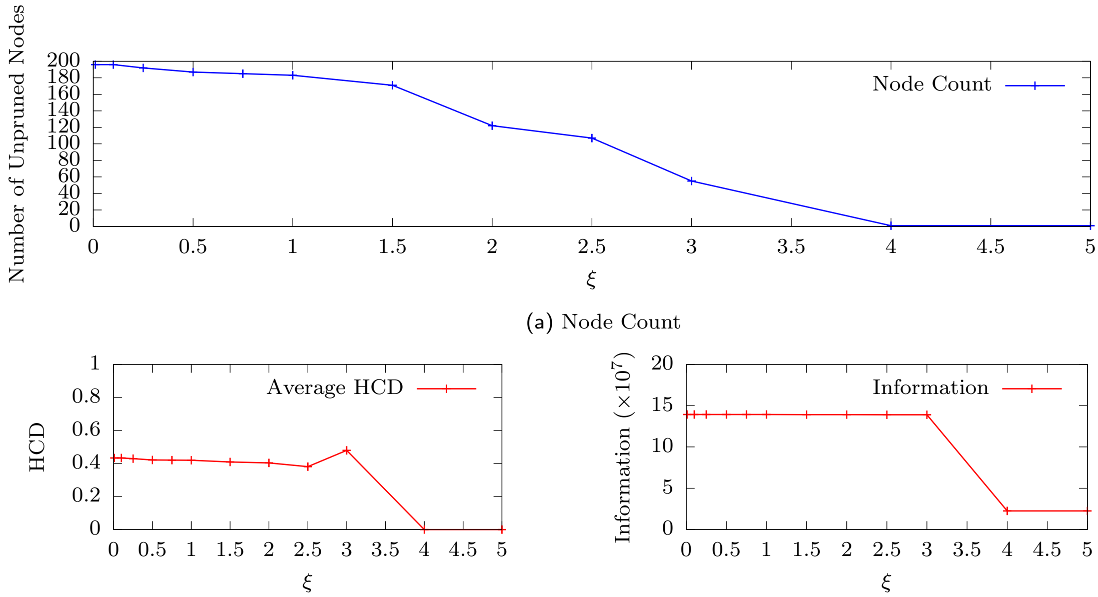
$$CostBenefitScore(C|P) = \frac{Utility(C|P)}{Cost(C|P)} \tag{11}$$

While utility is normalised between 0 and 1 as it represents the proportion of the parent that is not covered by the child, cost only has a minimum bound of ξ (Section 7.1.1), where $\xi > 0$. Depending upon the application, it may be desirable to also normalise cost relative to the current context tree. Using this metric on nodes depth-first, pruning should occur for any cluster C with parent P and $CostBenefitScore(C|P) < \theta$, where θ is the *pruning threshold* and C has no unpruned children.

7.2 Pruning Evaluation

Pruning requires a pre-built context tree and two parameters, namely θ and ξ, where θ provides a threshold for pruning, and ξ is a penalty associated with every node when calculating its *storage cost.*

Figure 26 shows the effect of varying θ when pruning a context tree generated from the same data and parameters as those used in Figure 17, with $\lambda = 0.5$ and $\xi = 1$. From this figure it is possible to see that the number of nodes in a context tree can be drastically reduced while maintaining the majority of the information. Selecting $\theta = 0.6$, the resultant pruned context tree contains approximately 20% of



(a) Node Count

(b) Average HCD

(c) Total Information

Figure 27: Effect of ξ on number of nodes in a sample context tree ($\lambda = 0.5, theta = 0.2$).

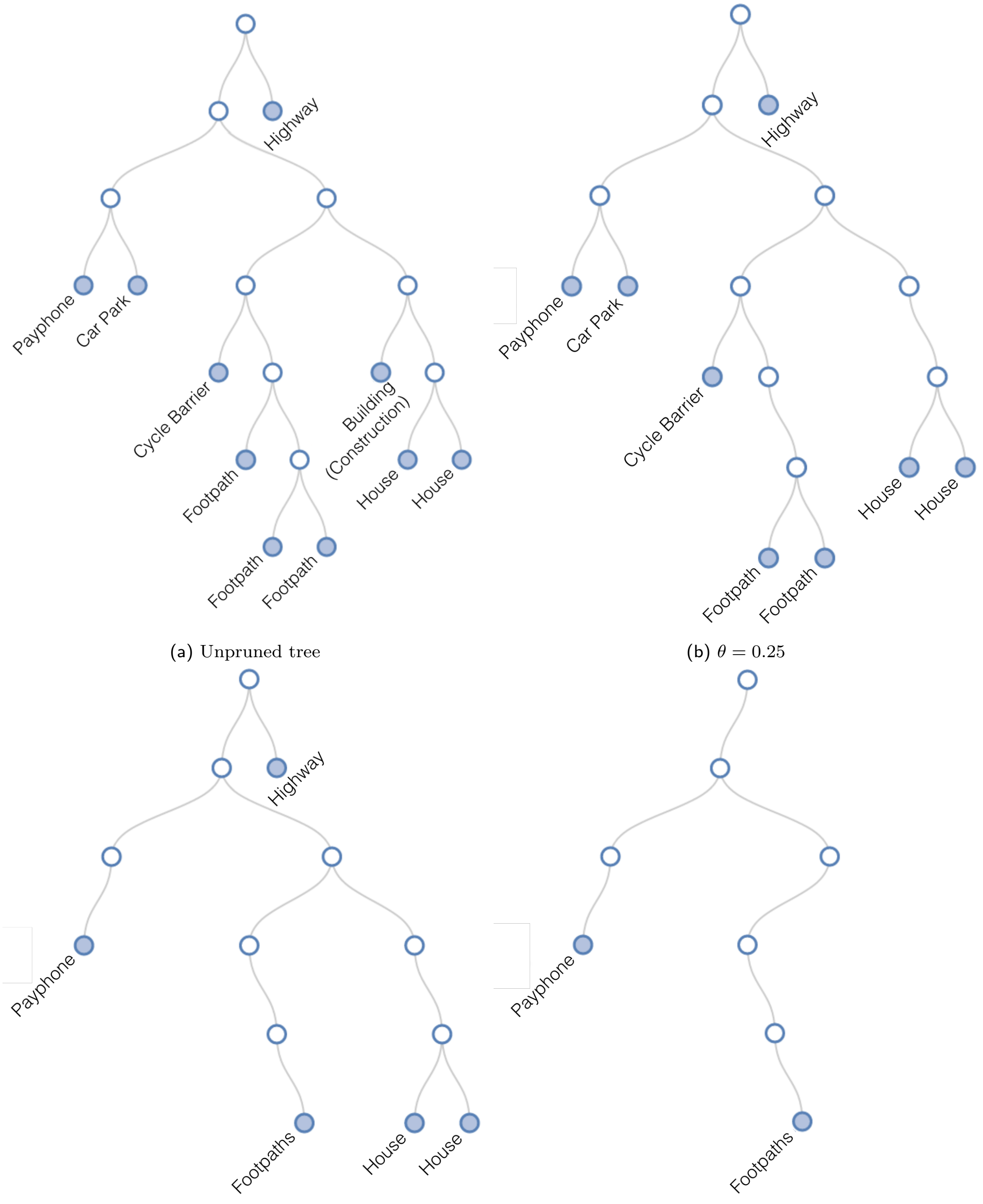
the nodes present in the unpruned tree, but maintains almost 70% of the useful information. While the process to prune the context tree adds in additional complexity, the resultant tree is considerably more compact and thus applications that require storing or searching the tree will have significantly lower overhead.

Using the same data again, but this time holding $\theta = 0.2$, Figure 27 shows the effect of changing ξ on the number of unpruned nodes, average HCD and information. Increasing either θ or ξ reduces the number of nodes left after pruning (Figures 26a and 27a), as increasing θ specifies a higher threshold required to maintain a node, and increasing ξ assigns a higher cost to each node, making it less likely to exceed the threshold. The results also demonstrate that as more nodes are pruned from the context tree, the average distance of the remaining nodes becomes smaller (i.e. they become more similar, Figures 26b and 27b). Finally, Figures 26c and 27c demonstrate that although pruning does reduce the total information in the tree, it does so gradually until the number of unpruned nodes approaches 0, under the definition of information presented in Equation 9. This helps to demonstrate the effectiveness of pruning as the number of nodes in the tree can be drastically reduced, but the amount of information remains high.

Figure 28 shows how pruning affects trees generated from real-world data (using the same data and clustering as in Figure 22). With the lowest value of θ ($\theta = 0.25$ shown in Figure 28b), only two leaf nodes have been pruned: one of the footpaths and one of the buildings. Increasing θ ($\theta = 0.35$ shown in Figure 28c) causes more leaf nodes to be pruned, and a further increase ($\theta = 0.45$ shown in Figure 28d) has the effect of pruning entire sub-trees, resulting in a much smaller and more compact tree. Although containing less information, such pruned trees provide benefits in resource-constrained applications where storing and processing an entire tree may be infeasible.

8 Conclusion

This work has presented the *context tree* hierarchical data structure that summarises user contexts at multiple scales. In addition to this, we proposed a method for constructing context trees from geospatial trajectories and land usage datasets. The context tree is a novel data structure that provides rapid access to summary information about a user's interactions with their environment, and thus provides a foundation for further analysis, understanding and modelling of the behaviours of individuals and groups. Furthermore, this work has presented an analysis of both context trees and the associated generation procedure, using real-world data and a partial ground truth, alongside a proposed method of pruning context trees to reduce their size, thus requiring less processing and memory for further applications.



(a) Unpruned tree

(b) $\theta = 0.25$

(c) $\theta = 0.35$

(d) $\theta = 0.45$

Figure 28: Context tree pruning for different values of θ, with $\xi = 1.5$.

The data employed for evaluation came from the publicly available MDC [Kiukkonen et al., 2010; Laurila et al., 2012] and GeoLife [Zheng et al., 2008a, 2009, 2010] datasets, consisting of GPS trajectories from real individuals, in addition to data collected ourselves.

Constructing context trees begins with processing of land usage data in a manner that considers both the extraction of relevant land usage information and filtering to remove noise, in addition to providing a novel technique for clustering related land usage elements to expose contexts by considering both properties of the real-world entities that the user interacted with, and properties of the interaction itself (e.g. the time and duration). These processes are combined with an agglomerative hierarchical clustering technique to generate the context tree.

By summarising contexts into a single data structure, it becomes easier to detect changes in routine through anomaly identification, identify similarities and differences between users to spot those with commonalities such as similar jobs or habits, and predict users' future actions. These areas are proving to be increasingly important to the provision of tailored and useful services both on individual and societal scales. Future work will expand existing techniques applied to locations and contexts by increasing their applicability to context trees. For example, expanding location prediction to operate over contexts such as those identified through contextual clustering would provide the ability to predict not only where a user is likely to be going, but also properties of the interaction, such as when and for how long. Furthermore, predictions need not relate to specific locations or entities, but rather to contexts and thus it would become possible to predict that a user will go to, for example, a building with certain properties without the need to identify exactly which building will be the target.

Acknowledgements

Portions of the research in this paper used the MDC Database made available by Idiap Research Institute, Switzerland and owned by Nokia [Kiukkonen et al., 2010; Laurila et al., 2012].

References

Saif Ahmad, Tugba Taskaya-Temizel, and Khurshid Ahmad. 2004. Summarizing Time Series: Learning Patterns in 'Volatile' Series. In *Proceedings of the 5th International Conference on Intelligent Data Engineering and Automated Learning*, pages 523–532, Exeter. Springer. doi: 10.1007/978-3-540-28651-6_77.

Juan Antonio Alvarez-Garcia, Juan Antonio Ortega, Luis Gonzalez-Abril, and Francisco Velasco. 2010. Trip destination prediction based on past GPS log using a Hidden Markov Model. *Expert Systems With Applications*, 37(12):8166–8171. doi: 10.1016/j.eswa.2010.05.070.

Christos Anagnostopoulos, Athanasios Tsounis, and Stathes Hadjiefthymiades. 2006. Context Awareness in Mobile Computing Environments. *Wireless Personal Communications*, 42(3):445–464. doi: 10.1007/s11277-006-9187-6.

Gennady Andrienko, Natalia Andrienko, Christophe Hurter, Salvatore Rinzivillo, and Stefan Wrobel. 2011. From Movement Tracks Through Events to Places: Extracting and Characterizing Significant Places from Mobility Data. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pages 161–170, Providence. ISBN 978-1-4673-0015-5. doi: 10.1109/VAST.2011.6102454.

Daniel Ashbrook and Thad Starner. 2002. Learning Significant Locations and Predicting User Movement with GPS. In *Proceedings of the 6th International Symposium on Wearable Computers*, pages 101–108, Seattle. ISBN 0-7695-1816-8. doi: 10.1109/ISWC.2002.1167224.

Daniel Ashbrook and Thad Starner. 2003. Using GPS to Learn Significant Locations and Predict Movement Across Multiple Users. *Personal and Ubiquitous Computing*, 7(5):275–286. doi: 10.1007/s00779-003-0240-0.

Athanasios Bamis and Andreas Savvides. 2011. Exploiting Human State Information to Improve GPS Sampling. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 32–37, Seattle. doi: 10.1109/PERCOMW.2011.5766898.

Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. 2015. Recommendations in Location-based Social Networks: A Survey. *GeoInformatica*, 19(3):525–565. ISSN 1384-6175. doi: 10.1007/s10707-014-0220-8.

Tengfei Bao, Huanhuan Cao, Enhong Chen, Jilei Tian, and Hui Xiong. 2011. An Unsupervised Approach to Modelling Personalized Contexts of Mobile Users. *Knowledge and Information Systems*, 31(2):345–370. doi: 10.1007/s10115-011-0417-1.

Huanhuan Cao, Tengfei Bao, Qiang Yang, Enhong Chen, and Jilei Tian. 2010. An Effective Approach for Mining Mobile User Habits. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, pages 1677–1680, Toronto. doi: 10.1145/1871437.1871702.

Huiping Cao, Nikos Mamoulis, and David Cheung. 2005. Mining Frequent Spatio-temporal Sequential Patterns. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 82–89, New Orleans. doi: 10.1109/ICDM.2005.95.

Huiping Cao, Nikos Mamoulis, and David W. Cheung. 2007. Discovery of Periodic Patterns in Spatiotemporal Sequences. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):453–467. ISSN 1041-4347. doi: 10.1109/TKDE.2007.1002.

Qing Cao, Bouchra Bouqata, Patricia D Mackenzie, Daniel Messier, and Josheph J Salvo. 2009. A Grid-based Clustering Method for Mining Frequent Trips from Large-scale, Event-based Telematics Datasets. In *Proceedings of the 2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 2996–3001, San Antonio. doi: 10.1109/ICSMC.2009.5345924.

Chao Chen, Daqing Zhang, Pablo Samuel Castro, Nan Li, Lin Sun, and Shijian Li. 2011. Real-time Detection of Anomalous Taxi Trajectories from GPS Traces. In *Proceedings of the 8th International ICST Conference on Mobile and Ubiquitous Systems*, pages 63–74, Copenhagen. doi: 10.1007/978-3-642-30973-1_6.

Ling Chen, Mingqi Lv, and Gencai Chen. 2010. A System for Destination and Future Route Prediction Based on Trajectory Mining. *Pervasive and Mobile Computing*, 6(6):657–676. doi: 10.1016/j.pmcj.2010.08.004.

Peng Chen, Zhao Lu, and Junzhong Gu. 2009. Vehicle Travel Time Prediction Algorithm Based on Historical Data and Shared Location. In *Proceedings of the 5th International Joint Conference on INC, IMS and IDC*, pages 1632–1637, Seoul. doi: 10.1109/NCM.2009.138.

Yohan Chon, Elmurod Talipov, Hyojeong Shin, and Hojung Cha. 2011. Mobility Prediction-based Smartphone Energy Optimization for Everyday Location Monitoring. In *Proceedings of the 17th International Conference on World Wide Web*, pages 82–85, Seattle. doi: 10.1145/2070942.2070952.

Tanzeem Choudhury, Sunny Consolvo, Beverly Harrison, Jeffrey Hightower, Louis LeGrand, Ali Rahimi, Adam Rea, Gaetano Borriello, Bruce Hemingway, Predrag Klasnja, Karl Koscher, James A Landay, Jonathan Lester, Danny Wyatt, and Dirk Haehnel. 2008. The Mobile Sensing Platform: An Embedded Activity Recognition System. *Pervasive Computing*, 7(2):32–41. doi: 10.1109/MPRV.2008.39.

Anind Dey and Gregory Abowd. 1999. Towards a Better Understanding of Context and Context-Awareness. In *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pages 304–307, Karlsruhe. doi: 10.1007/3-540-48157-5_29.

Nathan Eagle and Alex Sandy Pentland. 2009. Eigenbehaviors: Identifying Structure in Routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066. doi: 10.1007/s00265-009-0739-0.

Nathan Eagle and Alex Sandy Pentland. 2005. Reality mining: sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268. doi: 10.1007/s00779-005-0046-3.

Mica R Endsley. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37:32–64.

Mica R Endsley. 2000. Theoretical Underpinnings of Situation Awareness: A Critical Review. *Situation Awareness Analysis and Measurement*, pages 3–28.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the 16th International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland.

Katayoun Farrahi and Daniel Gatica-Perez. 2008. Daily Routine Classification from Mobile Phone Data. In *Proceedings of the 5th International Workshop on Machine Learning for Multimodal Interaction*, pages 173–184, Utrecht. ISBN 978-3-540-85852-2. doi: 10.1007/978-3-540-85853-9_16.

Katayoun Farrahi and Daniel Gatica-Perez. 2010. Probabilistic Mining of Socio-Geographic Routines from Mobile Phone Data. *IEEE Journal of Selected Topics in Signal Processing*, 4(4):746–755. doi: 10.1109/JSTSP.2010.2049513.

Jun Fukano, Tomohiro Mashita, Takahiro Hara, and Kiyoshi Kiyokawa. 2013. A Next Location Prediction Method for Smartphones Using Blockmodels. In *Proceedings of the IEEE Conference on Virtual Reality*, pages 1–4, Orlando. doi: 10.1109/VR.2013.6549434.

Huiji Gao, Jiliang Tang, and Huan Liu. 2012. Mobile Location Prediction in Spatio-Temporal Context. In *Proceedings of the Nokia Mobile Data Challenge (MDC) Workshop in Conjunction with Pervasive*, Newcastle.

Fosca Giannotti, Mirco Nanni, Fabio Pinelli, and Dino Pedreschi. 2007. Trajectory pattern mining. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 330–339, San Jose. ISBN 978-1-59593-609-7. doi: 10.1145/1281192.1281230.

Joachim Gudmundsson, Marc van Kreveld, and Bettina Speckmann. 2004. Efficient Detection of Motion Patterns in Spatio-temporal Data Sets. In *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*, pages 250–257, Washington DC. doi: 10.1145/1032222.1032259.

Riccardo Guidotti, Roberto Trasarti, and Mirco Nanni. 2015. TOSCA: TwO-Steps Clustering Algorithm for Personal Locations Detection. In *Proceedings of the 23rd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Bellevue.

Newton Howard. *Theory of Intention Awareness in Tactical Military Intelligence: Reducing Uncertainty by Understanding the Cognitive Architecture of Intentions.* AuthorHouse, Bloomington, 2002.

Newton Howard and Erik Cambria. 2013. Intention awareness: improving upon situation awareness in human-centric environments. *Human-centric Computing and Information Sciences*, 3(1):17. doi: 10.1186/2192-1962-3-9.

Baoxing Huai, Enhong Chen, Hengshu Zhu, Hui Xiong, Tengfei Bao, Qi Liu, and Jilei Tian. 2014. Toward Personalized Context Recognition for Mobile Users: A Semisupervised Bayesian HMM Approach. *ACM Transactions on Knowledge Discovery from Data*, 9(2):10:1–10:29. doi: 10.1145/2629504.

Eunju Kim, Sumi Helal, and Diane Cook. 2010. Human Activity Recognition and Pattern Discovery. *Pervasive Computing*, 9(1):48–53. doi: 10.1109/MPRV.2010.7.

Niko Kiukkonen, Jan Blom, Olivier Dousse, Daniel Gatica-Perez, and Juha Laurila. 2010. Towards Rich Mobile Phone Datasets: Lausanne Data Collection Campaign. In *Proceedings of the First Workshop on Modeling and Retrieval of Context*, Berlin.

John Krumm and Eric Horvitz. 2006. Predestination: Inferring Destinations from Partial Trajectories. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, pages 243–260, Irvine. doi: 10.1007/11853565_15.

John Krumm and Dany Rouhana. 2013. Placer: Semantic Place Labels from Diary Data. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 163–172, Zurich. doi: 10.1145/2493432.2493504.

Juha K Laurila, Daniel Gatica-Perez, Imad Aad, Jan Blom, Olivier Bornet, Trinh Minh Tri Do, Olivier Dousse, Julien Eberle, and Markus Miettinen. 2012. The Mobile Data Challenge: Big Data for Mobile Computing Research. In *Proceedings of the Nokia Mobile Data Challenge (MDC) Workshop in Conjunction with Pervasive*, Newcastle.

Rikard Laxhammar and Goran Falkman. 2011. Sequential Conformal Anomaly Detection in Trajectories Based on Hausdorff Distance. In *Proceedings of the 14th International Conference on Information Fusion*, pages 1–8, Chicago.

Rikard Laxhammar and Goran Falkman. 2014. Online Learning and Sequential Anomaly Detection in Trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1158–1173. doi: 10.1109/TPAMI.2013.172.

Seon-Woo Lee and Kenji Mase. 2002. Activity and Location Recognition Using Wearable Sensors. *Pervasive Computing*, 1(3):24–32. doi: 10.1109/MPRV.2002.1037719.

Tayeb Lemlouma and Nabil Layaida. 2004. Context-aware Adaptation for Mobile Devices. In *Proceedings of the IEEE International Conference on Mobile Data Management*, pages 106–111, Berkeley. doi: 10.1109/MDM.2004.1263048.

Jonathan Lester, Tanzeem Choudhury, Nicky Kern, Gaetano Borriello, and Blake Hannaford. 2005. A Hybrid Discriminative/Generative Approach for Modeling Human Activities. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 766–772, Edinburgh.

Zhenhui Li, Bolin Ding, Jiawei Han, Roland Kays, and Peter Nye. 2010. Mining Periodic Behaviors for Moving Objects. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1099–1108, Washington DC. doi: 10.1145/1835804.1835942.

Lin Liao, Donald J Patterson, Dieter Fox, and Henry Kautz. 2007. Learning and Inferring Transportation Routines. *Artificial Intelligence*, 171(5-6):311–331. doi: 10.1016/j.artint.2007.01.006.

Siyuan Liu, Huanhuan Cao, L Li, and MengChu Zhou. 2013. Predicting Stay Time of Mobile Users with Contextual Information. *IEEE Transactions on Automation Science and Engineering*, 10:1026–1036. doi: 10.1109/TASE.2013.2259480.

James MacQueen. 1967. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the 5th Berkeley Symposium on Math, Statistics, and Probability*, pages 281–297, Berkeley.

George Miller. 1995. WordNet: a Lexical Database for English. *Communications of the ACM*, 38(11): 39–41. doi: 10.1145/219717.219748.

Anna Monreale, Fabio Pinelli, Roberto Trasarti, and Fosca Giannotti. 2009. Wherenext: A location predictor on trajectory pattern mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 637–646, Paris. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557091.

Raul Montoliu and Daniel Gatica-Perez. 2010. Discovering Human Places of Interest from Multimodal Mobile Phone Data. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Multimedia*, pages 12:1–12:10, Limassol. ISBN 978-1-4503-0424-5. doi: 10.1145/1899475.1899487.

Brendan Tran Morris and Mohan Manubhai Trivedi. 2011. Trajectory Learning for Activity Understanding: Unsupervised, Multilevel, and Long-Term Adaptive Approach. *IEEE Transactions on Pattern Analysis and Machine Learning*, 33(11):2287–2301. doi: 10.1109/TPAMI.2011.64.

Fumitaka Nakahara and Takahiro Murakami. 2012. A Destination Prediction Method Based on Behavioral Pattern Analysis of Nonperiodic Position Logs. In *Proceedings of The 6th International Conference on Mobile Computing and Ubiquitous Networking*, pages 32–39, Okinawa.

Donald J Patterson, Lin Liao, Dieter Fox, and Henry Kautz. 2003. Inferring High-Level Behavior from Low-Level Sensors. In *Proceedings of the 5th International Conference on Ubiqutous Computing*, pages 73–89, Seattle. doi: 10.1007/978-3-540-39653-6_6.

Susanna Pirttikangas, Kaori Fujinami, and Tatsuo Nakajima. 2006. Feature Selection and Activity Recognition from Wearable Sensors. In *Proceedings of the 3rd International Symposium on Ubiqutous Computing Systems*, pages 516–527, Seoul. doi: 10.1007/11890348_39.

Anand Rajaraman and David Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2011. ISBN 1107015359.

Nishkam Ravi, Nikhil Dandekar, Preetham Mysore, and Michael L Littman. 2005. Activity Recognition from Accelerometer Data. In *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence*, pages 1541–1546, Pittsburgh.

Philip Resnik. 1999. Semantic Similarity in a Taxonomy: An Information-based Measure and its Application to Problems of Ambiguity in Natural Language. *Journal of Artificial Intelligence Research*, 11: 95–130. doi: 10.1613/jair.514.

C Carl Robusto. 1957. The Cosine-Haversine Formula. *The American Mathematical Monthly*, 64(1): 38–40.

Olov Rosen and Alexander Medvedev. 2012. An On-line Algorithm for Anomaly Detection in Trajectory Data. In *Proceedings of the American Control Conference*, pages 1117–1122, Montreal. ISBN 978-1-4577-1095-7. doi: 10.1109/ACC.2012.6315346.

Bill Schilit, Norman Adams, and Roy Want. 1994. Context-Aware Computing Applications. In *Proceedings of the 1st Workshop on Mobile Computing Systems and Applications*, pages 85–90, Santa Cruz. doi: 10.1109/WMCSA.1994.16.

Katarzyna Siła-Nowicka, Jan Vandrol, Taylor Oshan, Jed A Long, Urška Demšar, and A Stewart Fotheringham. 2015. Analysis of Human Mobility Patterns from GPS Trajectories and Contextual Information. *International Journal of Geographical Information Science*, pages 1–26. doi: 10.1080/13658816.2015.1100731.

Lu-An Tang, Yu Zheng, Jing Yuan, Jiawei Han, Alice Leung, Chih-Chieh Hung, and Wen-Chih Peng. 2012. On Discovery of Traveling Companions from Streaming Trajectories. In *Proceedings of the 28th IEEE International Conference on Data Engineering*, pages 186–197, Washington DC. doi: 10.1109/ICDE.2012.33.

Alasdair Thomason, Nathan Griffiths, and Matthew Leeke. 2015a. Extracting Meaningful User Locations from Temporally Annotated Geospatial Data. In *Internet of Things: IoT Infrastructures*, volume 151 of *LNICST*, pages 84–90. Springer. doi: 10.1007/978-3-319-19743-2_13.

Alasdair Thomason, Nathan Griffiths, and Victor Sanchez. 2015b. Parameter Optimisation for Location Extraction and Prediction Applications. In *Proceedings of the 2015 IEEE International Conference on Pervasive Intelligence and Computing*, pages 2173–2180, Liverpool. doi: 10.1109/CIT/IUCC/DASC/PICOM.2015.322.

Alasdair Thomason, Matthew Leeke, and Nathan Griffiths. 2015c. Understanding the Impact of Data Sparsity and Duration for Location Prediction Applications. In *Internet of Things: IoT Infrastructures*, volume 151 of *LNICST*, pages 192–197. Springer. doi: 10.1007/978-3-319-19743-2_29.

Alasdair Thomason, Nathan Griffiths, and Victor Sanchez. 2016. Identifying Locations from Geospatial Trajectories. *Journal of Computer and System Sciences*, 82:566–581. doi: 10.1016/j.jcss.2015.10.005.

Alessandro Vinciarelli, Anna Esposito, Elisabeth André, Francesca Bonin, Mohamed Chetouani, Jeffrey F Cohn, Marco Cristani, Ferdinand Fuhrmann, Elmer Gilmartin, Zakia Hammal, Dirk Heylen, Rene Kaiser, Maria Koutsombogera, Alexandros Potamianos, Steve Renals, Giuseppe Riccardi, and Albert Ali Salah. 2015. Open Challenges in Modelling, Analysis and Synthesis of Human Behaviour in Human–Human and Human–Machine Interactions. *Cognitive Computation*, 7(4):397–413. doi: 10.1007/s12559-015-9326-z.

Jingjing Wang and Bhaskar Prabhala. 2012. Periodicity Based Next Place Prediction. In *Proceedings of the Nokia Mobile Data Challenge (MDC) Workshop in Conjunction with Pervasive*, Newcastle.

Zhengwei Wu, Haishan Wu, and Tong Zhang. 2015. Predict User In-world Activity via Integration of Map Query and Mobility Trace. In *Proceedings of the 4th International Workshop on Urban Computing*.

Zhibiao Wu and Martha Palmer. 1994. Verb Semantics and Lexical Selection. In *Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138. doi: 10.3115/981732.981751.

Xiangye Xiao, Yu Zheng, Qiong Luo, and Xing Xie. 2012. Inferring Social Ties Between Users with Human Location History. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):3–19. doi: 10.1007/s12652-012-0117-z.

Zhixian Yan, Dipanjan Chakraborty, Christine Parent, Stefano Spaccapietra, and Karl Aberer. 2013. Semantic trajectories: Mobility data computation and annotation. *ACM Transactions on Intelligent Systems and Technology*, 4(3). doi: 10.1145/2483669.2483682.

Jiong Yang, Wang Wang, and Philip S. Yu. 2003. Mining Asynchronous Periodic Patterns in Time Series Data. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):613–628. doi: 10.1109/TKDE.2003.1198394.

Zhiwen Yu, Hui Wang, Bin Guo, Tao Gu, and Tao Mei. 2015. Supporting Serendipitous Social Interaction Using Human Mobility Prediction. *IEEE Transactions on Human-Machine Systems*, 45:811–818. doi: 10.1109/THMS.2015.2451515.

Daqing Zhang, Nan Li, Zhi-Hua Zhou, Chao Chen, Lin Sun, and Shijian Li. 2011. iBAT: Detecting Anomalous Taxi Trajectories from GPS Traces. In *Proceedings of the 13th International Conference on Ubiquitous Computing*, pages 99–108, Beijing. ISBN 978-3-642-30972-4. doi: 10.1007/978-3-642-30973-1.

Kai Zheng, Yu Zheng, Xing Xie, and Xiaofang Zhou. 2012. Reducing Uncertainty of Low-Sampling-Rate Trajectories. In *IEEE 28th International Conference on Data Engineering*, pages 1144 – 1155, Washington DC. doi: 10.1109/ICDE.2012.42.

Yu Zheng and Xing Xie. 2010. Learning Travel Recommendations From User-generated GPS Traces. *ACM Transactions on Intelligent Systems and Technology*, 2(2):2:1–2:29. doi: 10.1145/1889681.1889683.

Yu Zheng, Quannan Li, Yukun Chen, Xing Xie, and Wei-Ying Ma. 2008a. Understanding Mobility Based on GPS Data. In *Proceedings of the 10th International Conference on Ubiquitous Computing*, pages 312–321, Seoul. ISBN 978-1-60558-136-1. doi: 10.1145/1409635.1409677.

Yu Zheng, Like Liu, Longhao Wang, and Xing Xie. 2008b. Learning Transportation Mode from Raw GPS Data for Geographic Applications on the Web. In *Proceedings of the 17th International Conference on World Wide Web*, pages 247–256, Beijing. ISBN 9781605580852. doi: 10.1145/1367497.1367532.

Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma. 2009. Mining Interesting Locations and Travel Sequences from GPS Trajectories. In *Proceedings of the 18th International Conference on World Wide Web*, pages 791–800, Madrid. doi: 10.1145/1526709.1526816.

Yu Zheng, Xing Xie, and Wei-Ying Ma. 2010. GeoLife: A Collaborative Social Networking Service Among User, Location and Trajectory. *IEEE Database Engineering Bulletin*, 33(2):32–39.