Days of Our Lives: Assessing Day Similarity from Location Traces

James Biagioni

Department of Computer Science
University of Illinois at Chicago
Chicago, IL USA
jbiagi1@uic.edu

John Krumm

Microsoft Research
Microsoft Corporation
Redmond, WA USA
jckrumm@microsoft.com

Abstract. We develop and test algorithms for assessing the similarity of a person's days based on location traces recorded from GPS. An accurate similarity measure could be used to find anomalous behavior, to cluster similar days, and to predict future travel. We gathered an average of 46 days of GPS traces from 30 volunteer subjects. Each subject was shown random pairs of days and asked to assess their similarity. We tested eight different similarity algorithms in an effort to accurately reproduce our subjects' assessments, and our statistical tests found two algorithms that performed better than the rest. We also successfully applied one of our similarity algorithms to clustering days using location traces.

Keywords: location traces, similarity, anomaly detection, clustering

1 Introduction

Both consumers and corporations recognize the value of location traces for understanding daily habits and anticipating occasional needs, and the proliferation of GPS-equipped smart phones is making them ever easier to collect. These traces can help in understanding our daily activities; in particular, we can use location traces to find anomalous days and to cluster similar days, leading to a better understanding of our daily routines. Both of these tasks require a way to compare days to one another.

This paper develops and tests algorithms to measure the similarity of days represented by location traces, tested against similarity assessments from real users. With a reliable way to measure similarity we can find days that stand out from the rest as anomalies, which may indicate confusion (an important phenomenon to detect among populations of users with cognitive impairments [3]) or a change of habits. We can also make sensible clusters of days that belong together to assess variety and make predictions about how a day will evolve, providing useful basic knowledge for future adaptive systems to leverage. We believe this is the first effort aimed at measuring the similarity of days using location traces in a way that reflects human assessments.

A variety of sensors could be used to characterize a day, such as activity measured on a person's mobile phone, desktop computer, vehicle, social networking sites, biometric sensors, *etc*. Our work is aimed at location traces, usually measured with GPS. One advantage of this is that location is a constantly existent state (if not always measurable) as opposed to event-based activities, such as texting events, that only happen occasionally. Location is also dynamic for most people and easy to measure

adfa, p. 1, 2013.

outdoors with GPS. These characteristics make location a convenient variable to use for measuring the similarity between a person's days.

The GIS community has looked extensively at location trace similarity, *e.g.* [1], but these efforts are aimed primarily at machine processing. We are interested in matching human assessments of similarity, which appears more commonly in research for anomaly detection. In [2], Ma detects anomalies from GPS traces by first representing a normal trace as a sequence of rectangles on the ground. An anomaly is declared if a new trace's rectangles are sufficiently different from those of the normal trace. Here the similarity measure is explicit in that it depends on a quantity measuring the geographic difference between the normal trip and the query trip. It also ignores time. In [3], Patterson *et al.* detect anomalous behavior based on GPS tracking. They train a dynamic, probabilistic model from a person's historical GPS traces. If the uncertainty of the trained model exceeds the uncertainty of a general prior model of human motion, then the system declares an anomaly. This is an example of an implicit similarity measure. Both [2] and [3] are aimed at detecting anomalies in the lives of the cognitively impaired. The system of Giroux *et al.* [4] has the same goal, only they use sensors in a home to detect anomalies in predefined daily routines, like making coffee. An anomaly is declared if the normal sequence of events is violated or if the timing of the sequence is sufficiently different from normal. Researchers have also detected anomalies in video, such as Xian and Gong [5], whose system automatically builds models of normality from training video.

All of these techniques depend on learning a model of normal behavior from observation, which means they must be trained anew for new subjects. One of our goals is to find a single similarity measure that works well for multiple people, without requiring any training. In addition, previous techniques detect dissimilar behavior based on an algorithm or threshold designed by the researcher. Instead, another of our goals is to find a similarity measure that approximates what a human subject would say about their own data. Achieving these goals will allow us to provide future adaptive systems with a way to accurately reproduce human assessments of day similarity that works well for the general population, and requires no training time; perhaps helping to mitigate the cold-start problem in relevant applications. Toward this end, we gathered GPS data from 30 volunteer subjects and then asked them to assess the similarity of their own days. Armed with this ground truth data, we tested various similarity measures and were able to find two that reproduced the assessments from our subjects quite well. We begin by describing how we gathered the data for our experiment.

2 GPS Data and Preprocessing

In order to perform our experiments for assessing day similarity based on location traces, we gathered GPS data from the vehicles of volunteers. This section describes our data logging and preprocessing for the experiment described in Section 3.

2.1 GPS Data from Volunteers

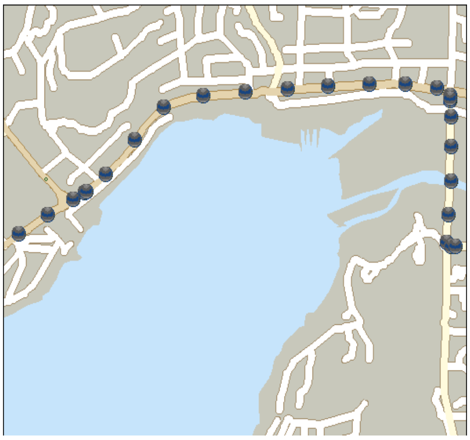


Figure 1: A short sequence of GPS points sampled at an interval of 10 seconds.

We logged GPS data from 30 volunteers (8 female). Each volunteer borrowed a RoyalTek RBT-2300 GPS logger and placed it in their main vehicle, powered by the cigarette lighter. All our subjects were employees of Microsoft Corporation in Redmond, WA, USA, and most were compensated with a US$ 30 cafeteria spending card. A few subjects agreed to participate without any compensation. Our goal was to collect at least six weeks of data from each subject. In the end we obtained GPS data for an average of 46 days from each subject, varying from 20 to 60 days, where the majority of the recorded drives consisted of simple weekday home/work commute trips and weekend drives in the local community; a dataset we believe generalizes well to the larger population of people with regular work routines. Each subject was in possession of the GPS logger for at least six weeks, but some did not drive every day. In order to reach 30 subjects, we started logging with 39 subjects, but later found that 9 did not provide suitable data due to mysterious stoppages in logging, a late refusal to log, frequent sharing of their vehicle (which violated our survey criteria), and two unexpected departures. We also ignored two subjects who had only 14 and 18 days of logging.

The loggers were set to record a time-stamped coordinate pair (latitude, longitude) every 10 seconds. Figure 1 shows a short sequence of GPS points from one of our subjects with 10-second sampling. Since we ran our loggers without their rechargeable batteries, they logged only when the vehicle's cigarette lighter was powered. For some vehicles, this happens only when the vehicle is turned on, and for others the cigarette lighter is powered continuously. In our preprocessing, detailed below, we filled in gaps corresponding to these and other limitations of the recorded GPS stream.

2.2 GPS Data Preprocessing

In order to attach some semantic information to the raw GPS data, our first preprocessing step was to automatically detect the time and location of all *stops* in the raw traces. For our purposes, a stop is defined as any location in the GPS data where we detect that the subject/vehicle remained within a 300-meter (radius) circular region for 5 minutes or more. These parameters were chosen based on a training dataset, whose subjects were not included in our final evaluation.

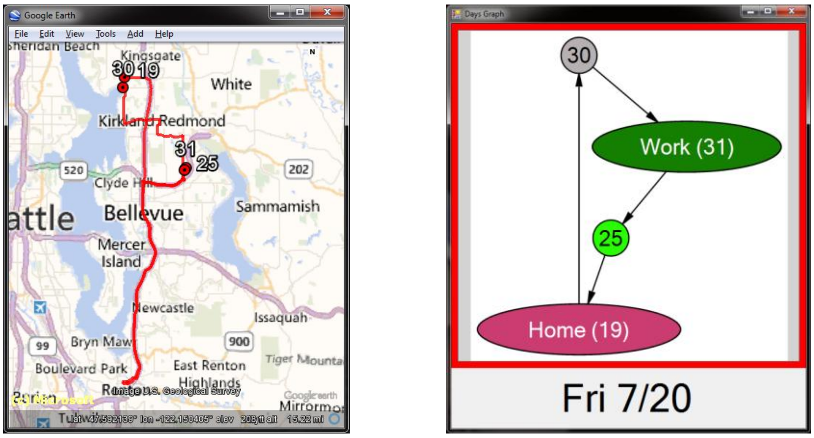
In order to produce an initial set of candidate stops, we first made a linear time-ordered pass through the GPS trace data and marked those locations that met our stop criterion, defined above. Because a stop location that was visited more than once during the course of the recorded GPS trace would have > 1 stop representation in our data, we then collapsed those redundant representations into one *final* stop. Doing so allowed us to associate a set of aggregate knowledge with the actual stop location. For example, consider the case of a subject's work location; over the course of a typical work-week their trace data will initially represent "work" with five separate stop representations (one for each day). By collapsing these five representations into one, we obtain one stop location that represents the aggregate knowledge of the original five (*i.e.*, days of the week the location was visited, times the subject arrived/departed, etc.), which is significantly more useful than five disparate time/location observations. In order to collapse the stops, we applied agglomerative clustering [6] to the candidate stops using the same 300-meter distance threshold (as above) as the criterion for merging.

Once we determined the final set of stops, we then leveraged the aggregate information contained therein to apply semantic labels to certain stops. Specifically, we used data from the American Time Use Survey (ATUS) [7] to classify the most-likely pair of stops as either *Home* or *Work* locations. Since our final stops contained knowledge of the days and times of arrival/departure, length of stay, and frequency of visits, we built and trained a classifier to perform probabilistic Home/Work labeling based entirely on these criteria. Since Home/Work stops occur very frequently in many subjects' GPS datasets, it was important to be able to distinguish them for our subjects' later assessment of their data. Specifically, having these labels helped our subjects orient themselves quickly and easily to the type of days they were observing *(e.g.*, weekday/weekend), and distinguish between regular and anomalous days more easily.

Finally, as one last preprocessing step, we created a *symbolized stop representation* of each day of data from the raw GPS traces (where a *day* is defined from 4:00am – 3:59am). Specifically, for each location in the raw GPS data, we replaced its coordinate pair with its associated *Stop ID* (a unique identifier associated with each stop), and interpolated in time for those vehicles that logged only when they were turned on. If a given coordinate pair was not associated with (*i.e.*, located at) a stop location, it was replaced with a *From Stop ID-To Stop ID* pair, denoting travel between stops. Simplifying the raw trace data into a series of symbols denoting time spent at (and traveling between) stops not only provides us with a more compact representation of the trace data, but also a more abstract representation for use with evaluation algorithms that aren't geographically-aware (see Section 4).

3 Human Assessment of Day Similarity

Our goal is to find an algorithm that can assess the similarity of days in a way that matches human assessment. Toward this end, we asked each of our subjects to make similarity assessments of their own location data. Guided by one of the authors, each of our 30 subjects was invited to run a custom program that displayed, and asked them to make similarity assessments on their own recently recorded data. The



(a) An interactive map for viewing a day's GPS data.

(b) A graph view of a day's GPS data, showing stops and the trips between them.

Figure 2: Two visualizations of a day for our subjects.

program started by displaying a calendar indicating the days for which we had GPS data available for the subject. For a selected day, the program showed that day's location traces in three different ways:

1. **Map -** An interactive map, shown in Figure 2(a), displayed the stops we found (as described in Section 2.2), each with its unique ID number. It also showed the GPS traces between the stops. This visualization emphasized the spatial layout of the day's trips and stops.
2. **Graph** – An interactive graph, as in Figure 2(b), showed the subject's stops as nodes and their trips as straight edges. Thicker edges indicated more trips between their connected stops. The *Home* and *Work* stops were labeled if we found them, otherwise stops were labeled with only their unique ID number that matched the numbers on the map. Clicking on a node or edge in the graph highlighted the corresponding stop or GPS trace on the map, making for convenient exploration. This visualization emphasized the number of stops and the transitions between them.

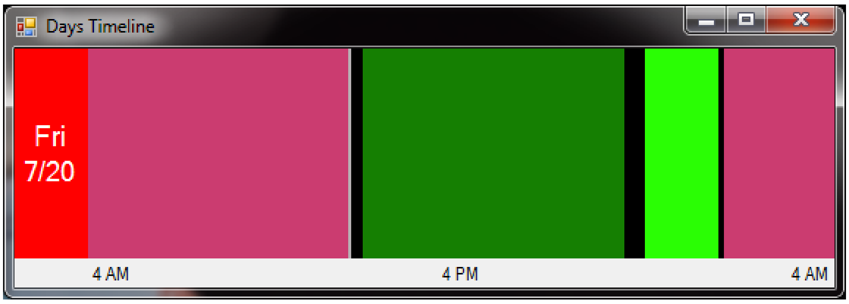


Figure 3: Timeline view of a day, showing stops as blocks of color and trips between stops as narrower bands of black.

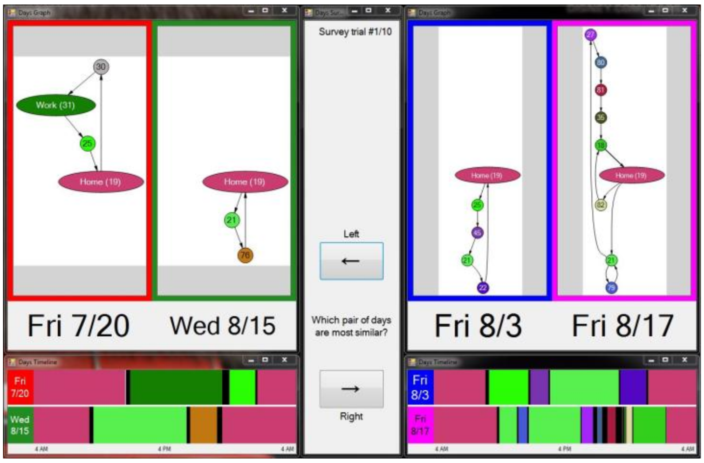


Figure 4: The main part of our user study, where we asked subjects to indicate which pair of days was most similar to each other.

3. **Timeline -** A timeline, as in Figure 3, displayed each stop in a different color block, laid out along a horizontal timeline. The time periods denoting trips between stops were colored black. This gave a temporal view of the day that was lacking in the other two visualizations.

After starting the program, we asked each of our subjects to familiarize themselves with the visualizations by picking a day and briefly describing it to us using the visualizations.

The main part of our user study came next: each subject was asked to assess the relative similarity of pairs of pairs of their days. That is, each subject was shown four randomly selected days simultaneously, using the visualizations described above, and as shown in Figure 4. We then asked the subject to indicate which of the two pairs was most similar. For instance, if the two pairs of days were A & B and C & D, we asked the subject to indicate if A & B were more similar to each other than C & D, or vice-versa. We chose this simple assessment after first piloting a different survey that asked subjects to give a numerical similarity rating to a pair of days. This proved too difficult, so we reverted to this simpler question about the relative similarity of pairs of days. The example shown in Figure 4 is a good representation of the complexity of the typical comparison problem presented to our subjects; with an average of 5 stops per day, the left-most pair of days represents a simpler case, and the right-most pair a more complex case. Each subject rated 30 pairs of pairs, which took approximately 30 minutes in total for each subject.

With these partial rankings, we next experimented with several different algorithms for assessing day similarity that we hoped would accurately reproduce the assessments of our subjects.

4 Algorithms for Assessing Day Similarity

To find an algorithm that computes a numerical similarity (or "distance" score) between pairs of days that matches the similarity rankings of our subjects, we implemented and evaluated four trajectory similarity algorithms in both *standard* and *modified* forms. The *standard* form of each algorithm is that given by its original definition, described in the following sub-sections. The *modified* form of each algorithm consisted of its original definition being adapted to use Dynamic Time Warping (DTW) [8], a technique which allows us to relax the assumption that activities between pairs of days be aligned in time. For example, consider two days A and B consisting of the same simple "Home → Work → Home" activity pattern. On Day A, the subject leaves home at 8:30am, arrives at work at 9am, departs work at 6pm, and returns home at 6:30pm. On Day B, the subject leaves home at 8am, arrives at work at 8:30am, departs work at 5:30pm, and returns home at 6pm. Since days A and B both consist of a 9-hour work-day with a 30-minute commute from/to home, subjectively speaking they are virtually identical. However, because of the 30-minute time-shift between them, they will necessarily incur a penalty from any objective similarity measure. Therefore, our motivation behind evaluating a DTW-modified version of each algorithm was to establish whether our subjects ignore these shifts in time, and if so, to more accurately capture and reproduce the subjective nature of their evaluations.

Formally speaking, in the *modified* implementation of each algorithm we measured the DTW-distance (*DTW*) by bootstrapping the corresponding *distance* function defined by each algorithm. The DTW-distance between days A and B is computed as follows, where A = $\langle a_1, a_2, \ldots, a_n \rangle$, Head(A) = a_1, Tail(A) = $\langle a_2, \ldots, a_n \rangle$, and each a_i corresponds to either a Stop ID or coordinate pair depending on the algorithm being modified (and, similarly for B):

$$DTW(A,B) = \begin{cases} 0, if\ length(A) = 0\ and\ length(B) = 0 \\ \infty, if\ length(A) = 0\ or\ length(B) = 0 \\ distance\big(Head(A), Head(B)\big) + min \begin{cases} DTW\big(A, Tail(B)\big) \\ DTW(Tail(A), B) \\ DTW\big(Tail(A), Tail(B)\big) \end{cases} \end{cases}$$

In effect, dynamic time warping warps the time axes of the two sequences so they match optimally. Below we describe the four standard trajectory similarity algorithms.

4.1 Edit Distance

Edit distance measures the number of *edit* operations needed to transform one string of symbols into another. In our case, this algorithm operates on the symbolized stop representation of our trace data (as discussed in Section 2.2), and therefore the *symbols* referred to here correspond to Stop IDs and From Stop ID-To Stop ID pairs.

Valid edit operations include: *insertion*, *deletion*, and *substitution*. In our evaluation, we used the canonical Levenshtein [9] implementation of this algorithm, where a unit cost is assigned to each of these operations. The result of this evaluation metric, in both its standard (denoted “without dynamic time warping”) and modified (denoted “with dynamic time warping”) form can be seen in Figure 5.

4.2 Distance Sensitive Edit Distance

The standard edit distance algorithm (described in Section 4.1 above) operates entirely on the symbolized stop representation of a given day, without taking into consideration the stops’ geographic locations. In order to account for the geographic location of stops we modified the standard Levenshtein algorithm [9] to use great-circle distance, measured using the Haversine formula [10], as its cost function for each of the edit operations. This means, for example, that the cost of performing the *substitution* operation for two Stops #60 and #157 is no longer 1, but rather the distance in meters between Stops #60 and #157 according to their coordinate locations. The results of this evaluation metric can be seen in Figure 5.

4.3 Stop Type Distance

The symbolized stop representation of a subject’s days requires an exact correspondence between IDs to be considered a match. Because this definition can be overly restrictive, we generalized the representation of each stop by classifying its location *type*. In order to perform this classification, we provided the coordinates of each stop to Bing Local Search, which returned a list of categorized businesses and their distances from our stop within a radius of 250 meters. Example business types include “Restaurant,” “Grocery & Food Stores” and “Banks & Credit Unions,” among many others. Using this data we then built a location-type probability distribution for each stop, based on the proportion of returned business types and weighted by their distance from the original stop location.

Replacing each Stop ID with its corresponding location-type probability distribution, we then computed the distance between days as the sum of the KL-divergence [11] scores between their probability distributions. The results of this evaluation metric can be seen in Figure 5.

4.4 Sum of Pairs Distance

This metric [12] computes the distance between days based on their raw location traces, rather than the symbolized stop representations used above. As a result, this metric does not take into account any of the related semantic information.

Sum of pairs distance measures the sum of the great-circle distance between every pair of trace locations (coordinate pairs). Since this metric requires that the traces for days A and B be of equal length, we first perform simple linear interpolation and then compute their distance. The results of this evaluation metric can be seen in Figure 5.

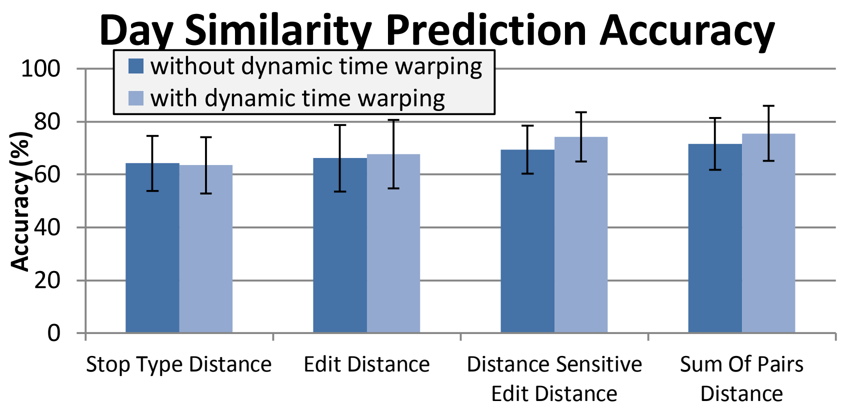


Figure 5: Accuracy results for our eight similarity algorithms. Error bars show +/- 1 standard deviation over all 30 test subjects.

5 Results

We evaluated our similarity algorithms both on the task of matching our subjects' similarity assessments and on a clustering task.

5.1 Matching Subjects' Similarity Assessments

We ran our eight similarity algorithms on the data from our 30 subjects. Recall that each subject was shown 30 sets of 4 days each. Each set of four days was split into two pairs, and the subject chose which pair was most similar. We gave these same sets of days to our similarity algorithms and recorded their assessment of which days were most similar. The accuracy results we report show the proportion of human decisions our algorithms were able to correctly reproduce.

The accuracy results are shown in Figure 5. Ignoring statistical significance, the best performing algorithm was Sum of Pairs Distance with Dynamic Time Warping (w/DTW), with a mean accuracy of 75.5% (*SD*=10.4%). This algorithm looks at the great circle distance between points in the two location traces, with local adjustments for time shifts. In second place was Distance Sensitive Edit Distance w/DTW with an overall mean accuracy of 74.2% (*SD*=9.3%). The fact that our two best-performing algorithms both base their distance metric on actual geographic distance is telling; clearly our subjects associate *day similarity* with *geographic proximity*.

Since we computed the accuracy for each subject, this provided 30 sample accuracies for each algorithm, allowing for a statistical analysis. We began with a one-way, repeated-measures ANOVA test, which resulted in $F(7,29) = 11.22, p = 5.45 \times 10^{-12}$. This is evidence that the choice of algorithm has a statistically significant effect on accuracy. We next performed one-tailed, paired-sample *t*-tests of the means between each pair of algorithms, with a Holm-Bonferroni [13] correction to account for the multiple *t*-tests. Of the 28 possible pairs of algorithms, 16 pairs had statistically significant mean accuracy differences at the $\alpha = 0.05$ level. Table 1 tallies the wins and losses of each algorithm. The algorithm with the best performance record is Distance Sensitive Edit Distance w/DTW, with five wins and no losses. The next best algorithm is Sum of Pairs Distance w/DTW, with four wins and no losses. There was no statistically significant difference in performance between these two algorithms. Of these two, Sum of Pairs Distance w/DTW is easier to implement, since it does not require the identification of stops in the location traces. While the two best-performing algorithms both used DTW, it produced a statistically significant performance improvement for only the Distance Sensitive Edit Distance algorithm, over its non-DTW counterpart.

Overall, for accuracy and ease of implementation, we are inclined to recommend Sum of Pairs Distance w/DTW as the best algorithm we tested for assessing the similarity of days.

5.2 Application to Clustering

One application of our similarity measure is clustering, where we can find groups of similar days. We tested this by having our 30 subjects assess clusterings of their own days. We clustered days with a spectral clustering algorithm (eigenvectors of random walk Laplacian, with *k*-means [14]). We computed clusters using the Edit Distance w/o DTW algorithm as our distance metric. Edit Distance w/o DTW had a mean accuracy of 66.2% (*SD*=12.5%), slightly lower than the best accuracy of 75.5% for Sum of Pairs Distance w/DTW. We used Edit Distance w/o DTW for our survey, because at the time we conducted our study we hadn't yet been able to test for the best performing algorithm.

For the clustering portion of the survey, we asked each subject to increment through the number of clusters, *k*, starting at two. For each *k*, the program displayed the clustered days in groups using the same visualizations described in Section 3. An example of a timeline showing three clusters is depicted in Figure 6, where the day-groupings are indicated by the colored labels on the left-hand side of each row.

Table 1: Number of statistically significant wins and losses for our similarity algorithms.

Algorithm	Wins	Losses
Edit Distance w/o DTW	0	3
Edit Distance w/DTW	0	1
Distance Sensitive Edit Distance w/o DTW	2	2
Distance Sensitive Edit Distance w/DTW	5	0
Stops Categories Distance w/o DTW	0	4
Stops Categories Distance w/DTW	0	4
Sum of Pairs Distance w/o DTW	3	0
Sum of Pairs Distance w/DTW	4	0

Each subject was asked to pick the best *k* and then to rate the clustering on a Likert scale by indicating their level of agreement with the statement, “My days have been accurately separated into sensible groups.” The results of this question are shown in Figure 7, where we see that 20 out of 30 subjects answered either “Agree” or “Strongly agree”, indicating that the clustering was generally successful. This, in turn, further supports the assertion that our Edit Distance w/o DTW similarity algorithm comes close to matching human similarity assessments. We would expect Sum of Pairs Distance w/DTW to work even better, since it was the most accurate algorithm based on our analysis in Section 5.1.

6 Conclusions

Based on a survey of 30 subjects, we assessed the accuracy of 8 different similarity algorithms on their location traces. We found that two algorithms, Sum of Pairs Distance w/DTW and Distance Sensitive Edit Distance w/DTW, worked best at matching human assessments of day similarity. We also showed that one of our similarity algorithms worked well for clustering days of location traces, based on an evaluation from our subjects.

In addition to clustering, these similarity algorithms can potentially be used to find anomalies and help predict behavior. None of our algorithms depend on training, so they are generic across all users, and therefore relatively easy to use.

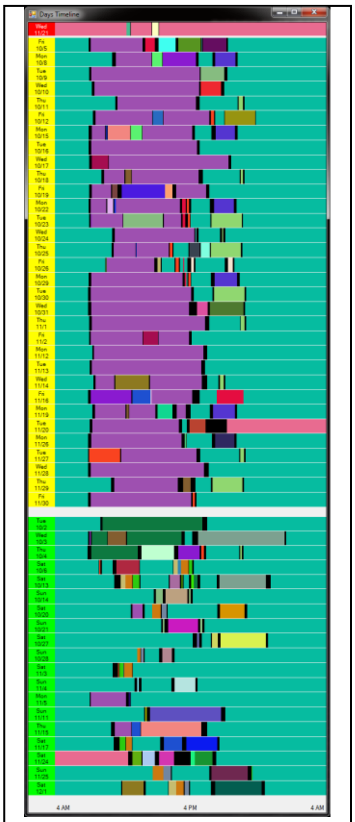


Figure 6: These are three clusters shown on a timeline. Each row is one day. The single day in the top cluster is an outlier. The main central cluster shows 32 work days, and the bottom cluster shows 19 non-work days.

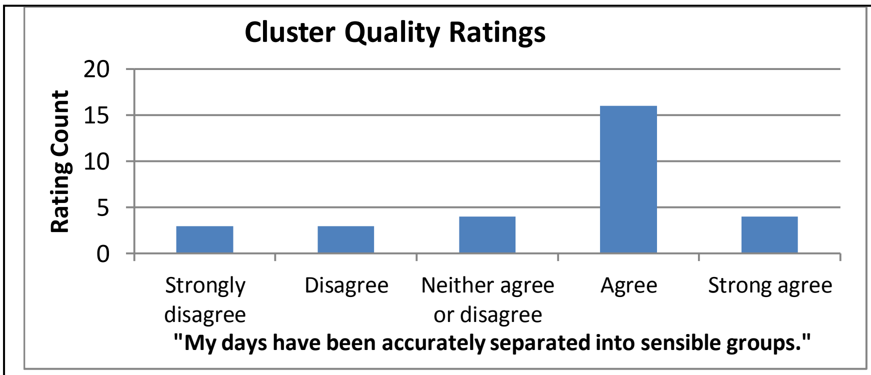


Figure 7: Most of our subjects were happy with the clustering results.

We envision future work in this area may explore other similarity algorithms as well as experiments to detect anomalies. We would expect anomaly detection to work well because of the good performance shown here by our algorithms at matching human assessments of the similarity of days.

References

1. Deng, K., et al., *Trajectory Indexing and Retrieval*, in *Computing with Spatial Trajectories*, Y. Zheng and X. Zhou, Editors. 2011, Springer: New York.
2. Ma, T.-S., *Real-Time Anomaly Detection for Traveling Individuals*, in *Eleventh International ACM SIGACCESS Conference on Computers and Accessibility(ASSETS '09)*. 2009: Pittsburgh, PA USA. p. 273-274.
3. Patterson, D.J., et al., *Opportunity Knocks: a System to Provide Cognitive Assistance with Transportation Services*, in *International Conference on Ubiquitous Computing (UbiComp 2012)*. 2004, Springer. p. 433-450.
4. Giroux, S., et al., *Pervasive Behavior Tracking for Cognitive Assistance*, in *1st International conference on PErvasive Technologies Related to Assistive Environments (PETRA '08)*. 2008, ACM.
5. Xiang, T. and S. Gong, *Video Behavior Profiling for Anomaly Detection* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2008. **30**(5): p. 893 - 908.
6. Hastie, T., R. Tibshirani, and J. Friedman, in *The Elements of Statistical Learning*. 2009, Springer: New York. p. 520-528.
7. United States Bureau of Labor Statistics. *American Time Use Survey (ATUS)*. Available from: http://www.bls.gov/tus/.
8. Yi, B.-K., H.V. Jagadish, and C. Faloutsos, *Efficient Retrieval of Similar Time Sequences Under Time Warping*, in *14th International Conference on Data Engineering*. 1998: Orlando, Florida USA. p. 201-208.
9. Levenshtein, V., *Binary Codes Capable of Correcting Deletions, Insertions and Reversals.* Soviet Physics Doklady, 1966. **10**(8): p. 707-710.
10. Sinnott, R.W., *Virtues of the Haversine.* Sky and Telescope, 1984. **68**(2): p. 159.
11. Kullback, S., *Information Theory and Statistics*. 1968, Mineola, NY USA: Dover.
12. Agrawal, R., C. Faloutsos, and A. Swami, *Efficient Similarity Search in Sequence Databases*, in *4th International Conference on Foundations of Data Organization and Algorithms (FODO '93)*. 1993: Chicago. p. 69-84.
13. Holm, S., *A Simple Sequentially Rejective Multiple Test Procedure.* Scandinavian Journal of Statistics 1979. **6**(2): p. 65–70.
14. Luxburg, U.v., *A Tutorial on Spectral Clustering.* Statistics and Computing, 2007. **17**(4): p. 395-416.