A dataset for determining user preferences of users on personal vehicles

Aleksandr Borodinov
Geoinformatics and Information Security Department,
Samara National Research University,
Samara, Russia
aaborodinov@yandex.ru

Vladislav Myasnikov
Geoinformatics and Information Security Department,
Samara National Research University,
Samara, Russia
vmyas@geosamara.ru

Alexander Yumaganov
Geoinformatics and Information Security Department,
Samara National Research University,
Samara, Russia
yumagan@gmail.com

***Abstract*—The paper considers the problem of matching GPS tracks to a road network. We presented a map-matching algorithm based on dynamic programming. We collected the tracks of movement around the city of several users on personal vehicles with various trip types to test the proposed algorithm. The data collected after matching to the road network can be used to further identify user preferences and to build a transport recommender system.**

Keywords—GPS Trajectory, map matching, road network.

I. Introduction

The area of recommendation systems has increased significantly over the past few years. Advertising on online resources offers users various products [1,2], based on the purchase history and viewing products in online stores. Streaming services select films and compose playlists of musical compositions for each individual user [3]. Research teams and large IT companies compiled data sets for each field of application to compare the various machine learning methods used in recommender systems. Transport navigation systems are one of the new areas of application of recommendation systems [4,5]. However, generally accepted machine learning methods and datasets for such systems do not yet exist. At the moment, researchers are trying to use publicly available data about user trips, such as OpenStreetMap, Strava or data on taxi driver trips [6,7]. The main disadvantage of such data is the inability to divide the available tracks by users to identify their preferences in choosing a route. Another drawback is the lack of information about the type and purpose of the trip. In the case when the trip is working or a navigation system has been used with the definition of the shortest path, the user preferences received will be unreliable.

The second section presents data on the collected tracks of user trips by personal vehicles. The second section presents data on the collected tracks of user trips by personal transport. The algorithm for linking tracks to the road network and the experimental results are described in the third section. At the end of the work, we presented a conclusion and possible directions for further work and research.

II. Data collection

We collected data in Samara in a large city with a population of about a million for 6 months from June to December 2019. Nine people of different sex, age, marital status and income, who are employees of Samara University, recorded the tracks of their trips. Users recorded work trips (a trip from home to work and from work to home) in an amount of at least 25 tracks and personal trips (all other trips) in an amount of at least 25 tracks. We define the route from the departure point to the destination point as a trip. In total, users recorded 489 tracks. 338 tracks were recorded on weekdays and 151 tracks were recorded on weekends. The generalized characteristics of the obtained data for all recorded tracks are presented in table 1.

TABLE I. GENERALIZED CHARACTERISTICS OF THE DATA

Data Characteristic	**Trip distance**	**Trip time**
Total value	4523 km	183 h 54 min 20 s
Mean	9249 m	22 min 33 s
Median value	5783 m	16 min 35 s
Maximum value	74405 m	2 ч 18 min 50 s
Minimum value	1264 m	2 min 13 s

Users recorded trips using personal smartphones with Android and iOS operating systems. Such a recording method is close to the real scenario of using navigation systems, in which the user receives a route or information about the load on the transport network through his smartphone. In this case, the navigation application on the smartphone can record data on the user's movement during interaction with the application.

All recorded tracks on google maps are shown in figures 1 and 2. The recorded tracks cover a significant part of the city's road network. Figure 2 left shows a user who, for six months, used the same route to and from work, and in Figure 2 right, the user used various routes to travel, depending on the congestion of the road network and weather conditions.

III. Algorithm for building a track using GPS points

A. Input data

Let $\{\overline{x}_i, t_i\}_{i=\overline{0,I-1}}$ - the data recorded during the trip, where $\overline{x}_i = (x_i, y_i, z_i)$ are the GPS coordinates of the trip, t_i is the recording time of the *i*-th route coordinate. We take $t_0 = 0$ for the recording start time.

We describe the road network as a directed graph $G = (V, W)$, where V - is the set of vertices of the graph, and W - is the set of edges of the graph connecting the vertices of V. Vertices have coordinates $\overline{x}_v = (x_v, y_v, z_v)$ and the traffic light presence $S(v) = \begin{cases} 0, & lack, \\ 1, & presence \end{cases}$. We describe the edge as $w_{v1,v2} = \begin{cases} \varnothing, & if\ there\ is\ no\ way\ from\ v1\ to\ v2, \\ (l^w; \upsilon^w_{\max}; h^w; X^w; c^w), & otherwise \end{cases}$, where l^w - edge length w, $\upsilon^w_{\max}$ - maximum permissible speed on w, X^w - set of points defining an edge w, road network ring code

$c^w = \begin{cases} 0, & \textit{not in the ring road}, \\ \textit{ring road code}, & \textit{otherwise} \end{cases}$. Edge type h^w can take the following values:

$$h^w = \begin{cases} 0 & - \ 1 \ \textit{lane} \\ 1 & - \ 2 \ \textit{lanes} \\ 2 & - \ 3 \ \textit{lanes} \\ 3 & - \ > 3 \ \textit{lanes without a central dividing strip}. \\ 4 & - \ > 2 \ \textit{lanes with a central dividing strip} \\ 5 & - \ > 4 \ \textit{lanes with a central dividing strip}, \\ & (\textit{Controlled} - \textit{access highway}) \end{cases}$$

The maximum permissible speed in the graph $\upsilon^0_{\max} = \max\limits_{w \in W} \upsilon^w_{\max}$ and current average speeds υ^w_{avr} for each edge.

B. *Algorithm parameters*

Let $\rho_{\min}$ - minimum matching distance (10 m); $\rho_{\max}$ - maximum matching distance (50 m); γ - time increase factor; δ - increasing the field of view step (0,2 m); K - the number of points in the group (3-5). Custom parameter α: $\alpha = \left(2\sigma^2\right)^{-1}, \quad \sigma = 20$.

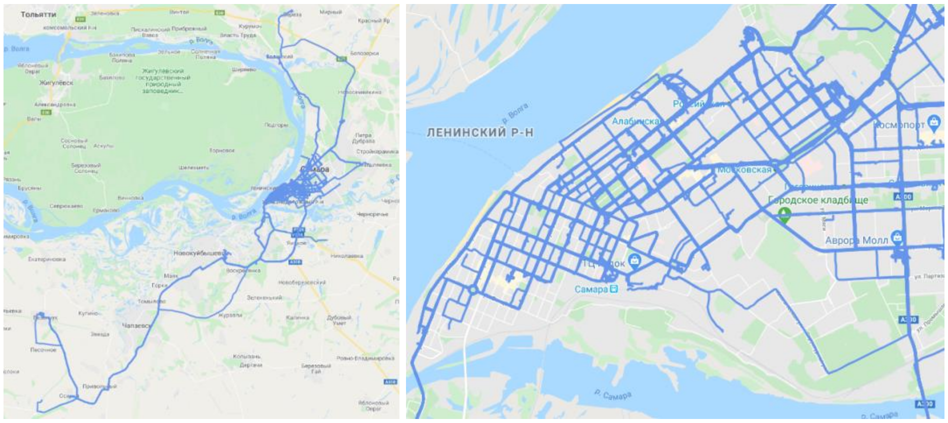

Fig. 1. Collected tracks on a large-scale map (left) and on a small-scale map (right).

C. *The result of the algorithm*

The result of the algorithm is the set $\{x_i, t_i, w_i, r_i, \upsilon_i\}_{i=\overline{0,I-1}}$, consisting of an adjusted sequence of points $\{x_i, t_i\}_{i=\overline{0,I-1}}$, for any of which an edge in the road network is indicated w_i that $x_i \in X^{w_i}$ and the outliers indicator r_i that $r_i = \begin{cases} 1, & \textit{i-th point is not an outlier}, \\ 0, & \textit{otherwise} \end{cases}$, and the estimated speed at the point is υ_i.

D. *Algorithm*

Step 1. For each point $\bar{x}_i, t_i$ we find the nearest edge w and match the point onto the edge as follows (ρ Euclidean):

$$w_i = \arg\min_{w \in W} \rho(\bar{x}_i, \bar{w}),$$

$$\bar{x}_i = \arg\min_{\underline{x} \in w_i} \rho(\bar{x}, \bar{x}_i).$$

Step 2. Consistently look at all the points by K pieces. For $K = 3: \overline{x_{i-1}}, \overline{x_i}, \overline{x_{i+1}}$. If all $\overline{x_{i\pm k}} \in w_i \ \& \ \rho(\overline{x_{i\pm k}}, \overline{x_{i\pm k}}) < \rho_{\min}$, then write the points to the result $\overline{x_{i\pm k}} := \overline{x_{i\pm k}}, \ w_{i\pm k} := w_{i\pm k}, \ r_{i\pm k} := 1$.

Then we select and consider all such sequences of points, find the minimum and maximum. In the case when the sequence is violated in time and position, we use the algorithm for linking points to a specific path described below.

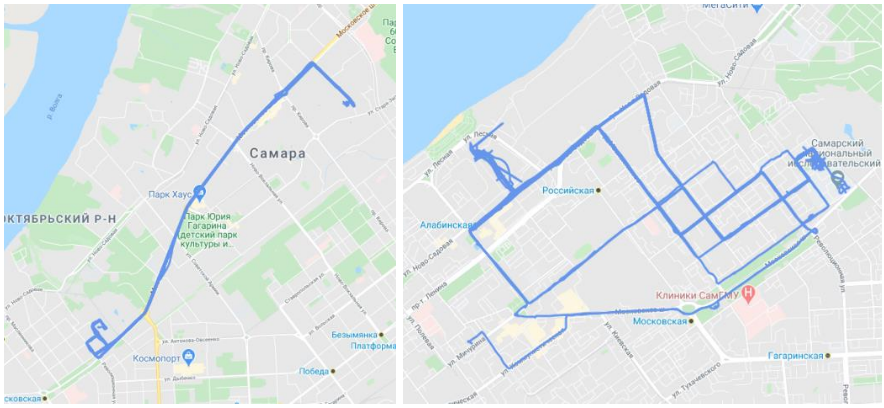

Fig. 2. An example of two user preferences in choosing a route to work.

After performing step 2, we get the matched sections of the path with gaps as shown in Figure 3. The blue color represents the attached points to the corresponding edges of the graph of the road network.

Next, we consider some arbitrary fragment from i_0 to i_1, i.e. points $\{\overline{x_{i_0}}, \overline{x_{i_0+1}}, \ldots, \overline{x_{i_1}}\}$.

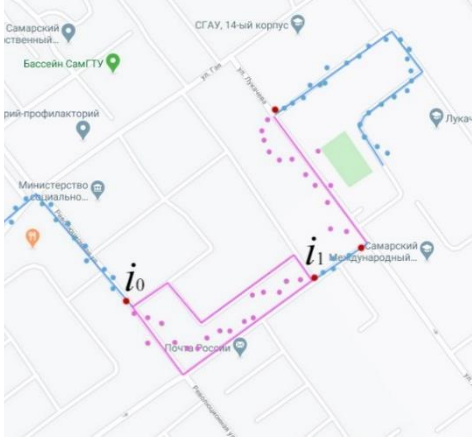

Fig. 3. The matched sections of the path with gaps.

Step 3. We determine the time interval for each point from $i_0 \to i_1$ to the extreme and determine the appearance physical possibility of this point. If $\frac{\rho(\overline{x_i}, \overline{x_{i_0}})}{t_i - t_{i_0}} > \upsilon_{\max}$ or $\frac{\rho(\overline{x_{i_1}}, \overline{x_i})}{t_{i_1} - t_i} > \upsilon_{\max}$ then the point is not taken into account and is further considered an outlier $r_i := 0$.

Step 4. We define a subgraph from point i_0 to i_1. We define the shortest path for this $i_0 \to i_1$ [8,9]. After that, we find a point $\overline{x}$ in the center of the shortest path and build a circle with a radius $R = (1+\delta)\cdot\max(\rho(\overline{x_i}, \overline{x_{i_0}}), \rho(\overline{x_{i_1}}, \overline{x_i}))$. In the subgraph we include all the vertices that fall into this circle and the corresponding edges.

Step 5. We find all the paths without loops in the resulting subgraph between i_0 and i_1. Denote this set P_{i_0,i_1}, where $\forall\ p \in P_{i_0,i_1} : p = (w_{i_0,i*}; w_{i*,\ldots}; \ldots; w_{\ldots,i_1})$.

For each path $p \in P_{i_0,i_1}$ we apply the developed algorithm for matching points to a specific path based on dynamic programming. Dynamic programming is often used to solve the map matching problem, which is confirmed by a large number of works [10-12].

E. *Algorithm for matching points to a specific path*

Further, to simplify the presentation (but without loss of generality), we consider $i_0 = 0$ and $i_1 = I - 1$. As a criterion for the quality of matching, we use the following:

$$J_p = \sum_{i=0}^{I-1} \exp(-\alpha \left\| \overline{x_i} - \overline{x_i^p} \right\|^2).$$

Suppose we have points $\{\overline{x}_i\}_{i=0}^{I-1}$ $(i_1 - i_0 + 1 = I)$ and they must be matched on the path $p = \{v_{n_0}, v_{n1}, \ldots, v_{K-1}\}$, where v_n has coordinates, v_{n_j} and $v_{n_{j+1}}$ connected by an edge. We discretize the possible positions of points $p(w_{v_{n_j} v_{n_{j+1}}}) = \{v_{n_0}, v_{n1}, \ldots, v_{K-1}\}$. For car discretization $\Delta \approx 2$ м. Let the total number of positions □ N, moreover $p(0) \sim v_{n_0}$, $p(N-1) \sim v_{n_{K-1}}$. We calculate I arrays of characteristics of the proximity of a point x_i to p how $\varphi_i(n) = \exp(-\alpha \left\| \overline{x_i} - p(n) \right\|^2)$.

The task is to find the sequence $n(i)_{\overline{i=0,I-1}} : \sum_{i=0}^{I-1} \varphi_i(n(i)) \to \max$, where $n(i) \geq n(i-1)$.

The main recurrence ratio (for the dynamic programming algorithm):

$$\max_{n(i)} \sum_{i=0}^{I-1} \varphi_i(n(i)) = \max_{\overline{n(i_l)=n(i_{l-1}),N}} \begin{bmatrix} \varphi_i(n(i_l)) + \\ \max\limits_{n(i) \leq n(i_l)} \sum\limits_{i=0}^{i_l - 1} \varphi_i(n(i)) + \\ \max\limits_{n(i) \geq n(i_l)} \sum\limits_{i=i_l-1}^{I-1} \varphi_i(n(i)) \end{bmatrix},$$

Denote $\varphi_j(n) = \max\limits_{n(i):i \leq j} \sum\limits_{i=0}^{j} \varphi_i(n(i))$, $\varphi_i(n)$ - similarity, $\varphi_i(n)$ - max integral similarity, $\pi_i(n)$ - point position list.

The result is contained in $\pi_0(0)$ and $\varphi_0(0)$.

Algorithm (start from the end):

```
for $i = I - 1, 0$
    for $n = \overline{N - 1, 0}$
        if ($i == I - 1$)
            if ($n == N - 1$)
                $\varphi_i(N-1) = \varphi_i(N-1)$
                $list = new\ List$
                $list.add(N-1)$
                $\pi_i(N-1) = list$
            else
                if $\varphi_i(n) > \varphi_i(n-1)$
                    $list = new\ List$
                    $list.add(n)$
                    $\pi_i(n) = list$
                    $\varphi_i(n) = \varphi_i(n)$
        else
            $\varphi_i(n) = \varphi_i(n-1)$
            $\pi_i(n) = \pi_i(n-1)$
    else // $(i == I-1)$
        if $(n == N-1)$
            $\varphi_i(N-1) = \varphi_i(N-1) + \varphi_{i+1}(N-1)$
            $\pi_i(N-1) = \pi_{i-1}(N-1)$
            $\pi_i(N-1).add(N-1)$
        else
            if $\varphi_i(n) + \varphi_{i+1}(n) > \varphi_i(n-1)$
                $list = new\ List$
                $\pi_i(n) = list$
                $list = copy(\pi_{i+1}(n))$
                $list.add(n)$
                $\varphi_i(n) = \varphi_i(n) + \varphi_{i+1}(n)$
            else
                $\varphi_i(n) = \varphi_i(n-1)$
                $\pi_i(n) = \pi_i(n-1)$
```

The result of the algorithm for matching the track to the road network is shown in Figure 4. The purple line shows the GPS coordinates of the track, the green line shows the matching to the road network.

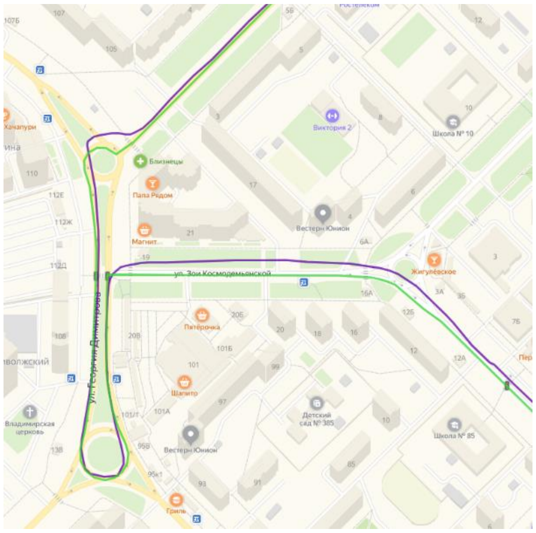

Fig. 4. Track matched to the road network (green) and track with raw GPS coordinates (purple).

IV. Conclusion

In our work, we presented a dataset containing tracks of user trips by personal vehicles. The paper also presents an algorithm for matching GPS travel tracks to a road network. The results of the algorithm are demonstrated on the city's road network. The presented data set of matched tracks to the road network can be used in the transport recommendation system development to obtain a profile of individual user preferences as a further area of research.

Acknowledgment

The authors would like to thank Geoinformatics and Information Security Department, Samara National Research University for taking part in a survey on data collection.

References

[1] T. Joachims, “Optimizing search engines using clickthrough data,” Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133-142, 2002.

[2] X. He, “Practical lessons from predicting clicks on ads at Facebook,” Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014. DOI: 10.1145/2648584.2648589.

[3] Y. Koren, R. Bell and C. Volinsky, “Matrix Factorization Techniques for Recommender Systems,” Computer, vol. 42, no. 8, pp. 30-37, 2009. DOI: 10.1109/MC.2009.263.

[4] P. Campigotto, C. Rudloff, M. Leodolter and D. Bauer, “Personalized and Situation-Aware Multimodal Route Recommendations: The FAVOUR Algorithm,” IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 1, pp. 92-102, 2017. DOI: 10.1109/TITS.2016.2565643.

[5] V. V. Myasnikov, “Reconstruction of functions and digital images using sign representations,” Computer Optics, vol. 43, no. 6, pp. 1041-1052, 2019. DOI: 10.18287/2412-6179-2019-43-6-1041-1052.

[6] X. Huang, “Grab-posisi: An extensive real-life GPS trajectory dataset in southeast Asia,” Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Prediction of Human Mobility, PredictGIS, pp. 1-10, 2019. DOI: 10.1145/3356995.3364536.

[7] J. Lian and L. Zhang, “One-month Beijing taxi GPS trajectory dataset with taxi IDS and vehicle status,” DATA - Proceedings of the 1st Workshop on Data Acquisition To Analysis, Part of SenSys, pp. 3-4, 2018. DOI: 10.1145/3277868.3277870.

[8] A.A. Agafonov, A.S. Yumaganov and V.V. Myasnikov, “Big data analysis in a geoinformatic problem of short-term traffic flow forecasting based on a K nearest neighbors method,” Computer Optics, vol. 42, no. 6, pp. 1101-1111, 2018. DOI: 10.18287/2412-6179-2018-42-6-1101-1111.

[9] A.A. Agafonov and V.V. Myasnikov, “Numerical route reservation method in the geoinformatic task of autonomous vehicle routing,” Computer Optics, vol. 42, no. 5, pp. 912-920, 2018. DOI: 10.18287/2412-6179-2018-42-5-912-920.

[10] T. Yokota, M. Okude, T. Sakamoto and R. Kitahara, “Fast and robust map-matching algorithm based on a global measure and dynamic programming for sparse probe data,” IET Intelligent Transport Systems, vol. 13, no. 11, pp. 1613-1623, 2019. DOI: 10.1049/iet-its.2019.0178.

[11] B.Y. Chen, H. Yuan, Q. Li, W.H.K. Lam, S.-L. Shaw and K. Yan, “Map-matching algorithm for large-scale low-frequency floating car data,” International Journal of Geographical Information Science, vol. 28, no. 1, pp. 22-38, 2014. DOI: 10.1080/13658816.2013.816427.

[12] Y. Li, Q. Huang, M. Kerber, L. Zhang and L. Guibas, “Large-scale joint map matching of GPS traces,” GIS: Proceedings of the ACM International Symposium on Advances in Geographic Information Systems, pp. 214-223,, 2013. DOI: 10.1145/2525314.2525333.