

Reproducing: Learning Points and Routes to Recommend Trajectories

Experiment Design for Data Science 188.992

Dachs Fabian

e01627961@student.tuwien.ac.at

Mathis Tobias Gallus

e1621473@student.tuwien.ac.at

Pletikosic Vice

e51831449@student.tuwien.ac.at

ABSTRACT

In order to evaluate the experiment design of the paper “Learning Points and Routes to Recommend Trajectories” by Dawei Chen, Cheng Soon Ong and Lexing Xie, this paper goes into what steps need to be taken to achieve to reproduce the mentioned paper. This paper then further evaluates the conclusions made by the authors, on the reproduced results as well as for the conclusions made on the original results.

KEYWORDS

Reproducibility; trajectory recommendation; learning to rank; planning

ACM Reference Format:

Dachs Fabian, Mathis Tobias Gallus, and Pletikosic Vice. 2022. Reproducing: Learning Points and Routes to Recommend Trajectories: Experiment Design for Data Science 188.992. In *Proceedings of the Lecture 188.992 Experiment Design for Data Science (VU 2,0) 2021W (188.992 EDDS 21/22)*. Technische Universität Wien, Vienna, Austria, 7 pages. <https://doi.org/10.5281/zenodo.5919118>

1 INTRODUCTION

The title of the reproduced paper is “Learning Points and Routes to Recommend Trajectories” by Dawei Chen, Cheng Soon Ong and Lexing Xie [2]. The paper presents and compares multiple methods for predicting trajectories for tourists to explore an unknown city over different points-of-interest (POI). The aim of this paper is to combine both the prediction problem (POI ranking) as well as the planning problem (route creation).

The authors argue that millions of photos are posted every month on the internet, which can be used to extract a dataset to learn trajectories i.e., an order to visit POIs of a city so that

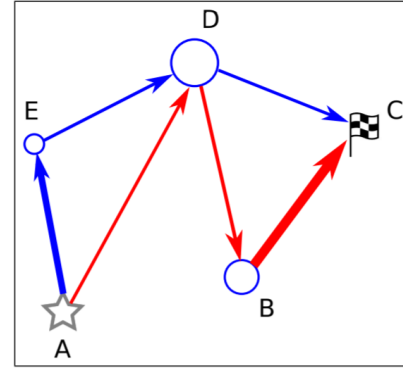


Figure 1: Trajectory recommendation [2]

the users are satisfied [2]. To properly evaluate the trajectories the paper proposes a performance metric. By using a F_1 -Score which not only measures the model suggests visiting a certain POI, but rather tests if a certain combination of POIs is in the predicted trajectory [2]. As a baseline the authors use a RANDOM prediction model, which chooses POIs uniformly at random, as well as POIPOPULARITY, which predicts trajectories solely on the popularity of the POI. From earlier work the authors used the model called PERSTOUR, which uses the POI features as well as sub-tour elimination constraints. PERSTOUR-L is a variant of PERSTOUR which constrains the trajectory length instead of the time budget. The methods POIRANK uses a point ranking method, MARKOV uses route-planning methods, RANK+MARKOV utilizes both approaches. These are also further extended by sub-tour elimination constraints in the methods MARKOVPATH and RANK+MARKOVPATH [2].

2 STRATEGIES

The authors provided the code and the data sets[1] [4] they used. The code and data were publicly available.

The aim of the reproduction was to work and research as accurate as possible. The work should be thoroughly checked first and the data and results carefully thought through. The code should be run exactly in the order described by the authors in the **readme** file, followed by cross-checking the code on a different operating system.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

188.992 EDDS 21/22, January 30–05, 2022, Vienna, Austria

© 2022 Association for Computing Machinery.

<https://doi.org/10.5281/zenodo.5919118>

If the above steps produce results, several tests should be applied to verify the legitimacy of the conclusion drawn from the original work.

3 DIFFICULTIES

3.1 Unknown parameters

Neither in the work nor in the code are precise system requirements specified. No information of platform and versions are stated.

3.1.1 Additional libraries. The **readme** file mentions loading an extra library which is linked. However, under this link there are two different libraries which are not specified in the **readme**. For this project the library `liblinear-ranksvm-2.11` was chosen.

3.1.2 Modifications. In order to successfully reproduce the provided code, multiple changes were made to successfully run the code and get results.

- (1) `np.x` to `x`: change variables
- (2) `python` instead of `cython` in `rank_markov`
- (3) `pb.solve` add specified path
- (4) added time limit (900s per Iteration)

On Windows, the following additional changes were made to the MacOS ones.

- (5) use `pb.solve` function without specified path

These code changes are all required due to new versions of individual libraries. A convenient way to simplify this for future projects would have been to use a list of the libraries used with their versions.

4 KEY FINDINGS

4.1 Comparison of reproduced results

After successfully running the models cited in the paper, we decided to compare the results we obtained with the results stated in the paper. First, we compared every result statistically, then we had a look whether the model that performed best in the paper on a certain dataset is the same as for our results and finally we try to argue whether we can come to the same conclusion as stated in the paper.

4.1.1 Statistical tests. In order to do a statistical test, we decided to perform a T-Test on every result and see whether we can find significant differences. Table 1 and table 2 show the P values we could obtain from the T-Test. The original results used for the T-Tests can be found in the appendix in table 5 and table 6, the reproduced results can be found in table 7 and table 8

No result we reproduced showed a significant difference. For the methods `POIPopularity` and `MARKOV`, we received the exact same results. The largest difference was with the

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
N	634	112	442	47	335
RANDOM	0.2883	0.3507	0.7663	0.5858	0.3030
POIPopularity	1.0000	1.0000	1.0000	1.0000	1.0000
PoiRANK	1.0000	0.9309	0.8330	1.0000	0.9392
MARKOV	1.0000	1.0000	1.0000	1.0000	1.0000
MARKOVPath	1.0000	0.8944	0.9205	1.0000	1.0000
RANK+MARKOV	0.9185	0.6639	0.7186	0.8828	0.9441
RANK+MARKOVPath	1.0000	0.7187	0.6096	0.8828	0.9392

Table 1: P-values of the T-Tests comparing the F_1 -Scores from the paper with the reproduced values

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
N	634	112	442	47	335
RANDOM	0.5442	0.2826	0.8360	0.6931	0.2136
POIPopularity	1.0000	1.0000	1.0000	1.0000	1.0000
PoiRANK	1.0000	0.9237	0.8240	1.0000	0.9651
MARKOV	1.0000	1.0000	1.0000	1.0000	1.0000
MARKOVPath	0.9396	0.8792	0.8145	1.0000	0.9552
RANK+MARKOV	0.9461	0.7303	0.6325	0.9203	0.9659
RANK+MARKOVPath	1.0000	0.7654	0.6644	0.8807	0.9653

Table 2: P-values of the T-Tests comparing the pairs- F_1 -Scores from the paper with the reproduced values

random method, but the difference still is not significant.

4.1.2 Difference in best model. Next, we compared what model performed best for each of the datasets. For most datasets we received the same method as in the paper as the best performing overall. Except for one instance, in the F_1 -Score for Melbourne dataset the method `PoiRANK` achieved better results as method `RANK+MARKOVPath`, in the paper it was the other way around. Although, here is the margin very small and the best performing models achieve very close results.

4.2 Experimental Design

4.2.1 Generation and usage of data. The evaluation of the methods stated in the paper is performed on datasets of five different cities. The authors reused four of these cities (Edinburgh, Glasgow, Osaka and Toronto) from earlier works and built the dataset for one city (Melbourne) themselves. To generate these datasets the authors used photos posted on Flickr and extracted location information from the photos metadata to recreate the order in which the user explored the POIs. This method clearly yields a lot of bias in the data. First Flickr users may not be representative or very diverse in the way the approach an exploration of a city, as their focus might be on taking photos to share on the internet rather than other motives. Second the Flickr users probably have visited other places in between two POIs without posting photos, this highly obscures the trajectory and makes it hard to learn real trajectories on this data. Another important critique on the data is in terms of user privacy. The authors did not mention any consent of the users they extracted the

data from. Although the users posted these images publicly, and we cannot evaluate the legal issues, this undoubtedly raises some ethical concern.

4.2.2 Reevaluate their conclusion. As we obtained reproduced results which were not significantly different than the results in the paper, we want to have a closer look on their conclusions. For that we used the stored results which also can be downloaded from their repository at *bitbucket*.

Plots of the results. In the paper the authors provided two tables with means and standard deviations of the scores to substantiate their conclusions. However, we had a hard time figuring out which conclusions to draw from these tables. In addition to that, as we are more interested in the uncertainty about their claims, we would have preferred the information about the standard error of the mean (sem) instead of the standard deviation (std). Figure 2 and Figure 3 show, in our opinion, a more clearer picture of the results for the dataset Edinburgh and Osaka. The blue dots represent the particular mean, the error-bars (golden) represent the standard error of the mean (sem) which is calculated as follow:

$$sem = std/\sqrt{n}$$

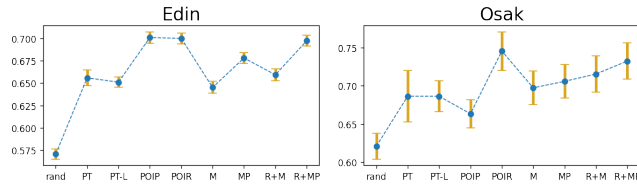


Figure 2: Mean-Values F_1 with errorbars for datasets Edinburgh and Osaka

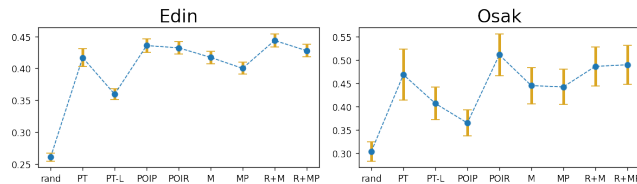


Figure 3: Mean-Values pairs- F_1 with errorbars for datasets Edinburgh and Osaka

Conclusions of plotting. We now see at first glance, that to draw substantial conclusions, the uncertainty in the Osaka dataset is too high. For the Edinburgh data set the values seems to be quite certain. However, it does not seem that there is one method which outperforms all others.

Paired t-test. We decided to test the following Hypothesis: There is no difference between the means of the best algorithm (=highest value in the table) and one of the other algorithms for each dataset (pairwise and paired t-test). The alternative Hypothesis is that the best algorithm has a greater mean-value than the other one. For this task we choose to use the paired t-test because the data and splits are the same for each algorithm. Because we do multiple test we use the Bonferroni correction in order to not inflate the possibility of making a type 1 error. Our initial alpha level is 5%, after correction (divided by 8) it is 0.625%. In the Table 3 and 4 we calculate the p-values (the best algorithm is marked with a 1 in the table). If the pvalue is less than 0.00625 we then can reject the H_0 and we can conclude that the best algorithm really performs better than the other one. This situation is marked with a "-" in the table. For all the other cases we cannot see a significant difference between the methods.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
N	634	112	442	47	335
RANDOM	-	-	-	-	-
PERS TOUR	-	1.00000	-	0.04841	-
PERS TOUR-L	-	-	-	0.02335	-
POIPOPULARITY	1.00000	-	-	-	-
POIRANK	0.41061	0.06694	0.34905	1.00000	1.00000
MARKOV	-	-	-	0.03107	-
MARKOV PATH	-	-	-	0.05814	-
RANK+MARKOV	-	0.01515	-	0.05554	-
RANK+MARKOV PATH	0.23036	0.03792	1.00000	0.22100	0.28018

Table 3: P-values from paired T-Tests. Testing highest value F-Score (=1) with all others on significant differences, p-values under 0.00625 were marked with "-".

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
N	634	112	442	47	335
RANDOM	-	-	-	-	-
PERS TOUR	0.04581	1.00000	-	0.23302	0.23755
PERS TOUR-L	-	-	-	0.02620	-
POIPOPULARITY	0.18602	-	-	-	-
POIRANK	0.05489	0.00790	0.15534	1.00000	1.00000
MARKOV	-	-	-	0.08295	-
MARKOV PATH	-	-	-	0.07155	-
RANK+MARKOV	1.00000	0.00670	1.00000	0.22660	0.28949
RANK+MARKOV PATH	-	-	0.03456	0.24745	0.35662

Table 4: P-values from paired T-Tests. Testing highest value pairs-F-Score (=1) with all others on significant differences, p-values under 0.00625 were marked with "-".

Conclusions of t-tests. From the tests we can draw two major conclusions. Nr.1: we see that the best algorithm always outperforms the RANDOM-Baseline. Nr.2: The best method depends of the specific dataset and performance metric used. However, POIRANK is in every combination either the winner or the difference from the winner is not significant.

5 CONCLUSION

All in all we can confirm and reproduce most of the results of the paper. Because of time constraints (the reproducing part was quite a challenge) we could not test all of their claims individually, but from the results of our tests so far, their claims seem very conclusive.

5.1 Reproducibility of the reproduced study

In this project we tried to improve the reproducibility for those who want to validate our results. All results can be found in the appendix.

The authors' repository was forked and expanded with the following items to ensure the reproducibility of this paper.

- added external rankSVM library
- updated code to match library versions
- requirements.txt added
- updated readme with further instructions

The code and the data used are available in a repository on Github. [3]

6 CITATIONS AND BIBLIOGRAPHIES

REFERENCES

- [1] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Data: Learning Points and Routes to Recommend Trajectories. Supplemental material, benchmark data and results retrieved from BitBucket.ORG, <https://bitbucket.org/d-chen/tour-cikm16/src/master/>.
- [2] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Learning Points and Routes to Recommend Trajectories. In *Proceedings of the 25th ACM International Conference on Information and Knowledge Management (CIKM '16)*. Indianapolis, IN, USA. <https://doi.org/10.1145/2983323.2983672>
- [3] Fabian Dachs, Tobias Gallus Mathis, and Vice Pletikoscic. 2022. edd. Data: Reproducing: Learning Points and Routes to Recommend Trajectories on Github, <https://github.com/plevice/edd.git>.
- [4] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2015. Personalized Tour Recommendation based on User Interests and Points of Interest Visit Durations.

A RESULT TABLES

A.1 Results from paper

Table 5: Performance comparison on five datasets in terms of F_1 score. The best method for each dataset (i.e., a column) is shown in bold, the second best is shown in italic.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
RANDOM	0.570 ± 0.139	0.632 ± 0.123	0.558 ± 0.149	0.621 ± 0.115	0.621 ± 0.129
PERS TOUR	0.656 ± 0.223	0.801 ± 0.213	0.483 ± 0.208	0.686 ± 0.231	0.720 ± 0.215
PERS TOUR-L	0.651 ± 0.143	0.660 ± 0.102	0.576 ± 0.141	0.686 ± 0.137	0.643 ± 0.113
POI POPULARITY	0.701 ± 0.160	0.745 ± 0.166	0.620 ± 0.136	0.663 ± 0.125	0.678 ± 0.121
POI RANK	<i>0.700 ± 0.155</i>	<i>0.768 ± 0.171</i>	<i>0.637 ± 0.142</i>	0.745 ± 0.173	0.754 ± 0.170
MARKOV	0.645 ± 0.169	0.725 ± 0.167	0.577 ± 0.168	0.697 ± 0.150	0.669 ± 0.151
MARKOV PATH	0.678 ± 0.149	0.732 ± 0.168	0.595 ± 0.148	0.706 ± 0.150	0.688 ± 0.138
RANK+MARKOV	0.659 ± 0.174	0.754 ± 0.173	0.613 ± 0.166	0.715 ± 0.164	0.723 ± 0.185
RANK+MARKOV PATH	0.697 ± 0.152	0.762 ± 0.167	0.639 ± 0.146	<i>0.732 ± 0.162</i>	<i>0.751 ± 0.170</i>

Table 6: Performance comparison on five datasets in terms of pairs- F_1 -Scores. The best method for each dataset (i.e., a column) is shown in bold, the second best is shown in italic.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
RANDOM	0.261 ± 0.155	0.320 ± 0.168	0.248 ± 0.147	0.304 ± 0.142	0.310 ± 0.167
PERS TOUR	0.417 ± 0.343	0.643 ± 0.366	0.216 ± 0.265	0.468 ± 0.376	0.504 ± 0.354
PERS TOUR-L	0.359 ± 0.207	0.352 ± 0.162	0.266 ± 0.140	0.406 ± 0.238	0.333 ± 0.163
POI POPULARITY	<i>0.436 ± 0.259</i>	0.507 ± 0.298	0.316 ± 0.178	0.365 ± 0.190	0.384 ± 0.201
POI RANK	0.432 ± 0.251	<i>0.548 ± 0.311</i>	0.339 ± 0.203	0.511 ± 0.309	0.518 ± 0.296
MARKOV	0.417 ± 0.248	0.495 ± 0.296	0.288 ± 0.195	0.445 ± 0.266	0.407 ± 0.241
MARKOV PATH	0.400 ± 0.235	0.485 ± 0.293	0.294 ± 0.187	0.442 ± 0.260	0.405 ± 0.231
RANK+MARKOV	0.444 ± 0.263	0.545 ± 0.306	0.351 ± 0.220	0.486 ± 0.288	0.512 ± 0.303
RANK+MARKOV PATH	0.428 ± 0.245	0.533 ± 0.303	<i>0.344 ± 0.206</i>	<i>0.489 ± 0.287</i>	<i>0.514 ± 0.297</i>

A.2 Reproduced results on Mac

Table 7: Performance comparison on five datasets in terms of F_1 -Scores. The best method for each dataset (i.e., a column) is shown in bold, the second best is shown in italic.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
RANDOM	0.578 ± 0.129	0.617 ± 0.117	0.555 ± 0.151	0.633 ± 0.097	0.611 ± 0.122
PERS TOUR	0.656 ± 0.223	0.801 ± 0.213	0.483 ± 0.208	0.686 ± 0.231	0.720 ± 0.215
PERS TOUR-L	0.651 ± 0.143	0.660 ± 0.102	0.576 ± 0.141	0.686 ± 0.137	0.643 ± 0.113
POI POPULARITY	0.701 ± 0.160	0.745 ± 0.166	0.620 ± 0.136	0.663 ± 0.125	0.678 ± 0.121
POI RANK	<i>0.700 ± 0.154</i>	<i>0.770 ± 0.174</i>	0.635 ± 0.140	0.745 ± 0.173	0.753 ± 0.169
MARKOV	0.645 ± 0.169	0.725 ± 0.167	0.577 ± 0.168	0.697 ± 0.150	0.669 ± 0.151
MARKOV PATH	0.678 ± 0.149	0.735 ± 0.170	0.596 ± 0.150	0.706 ± 0.150	0.688 ± 0.138
RANK+MARKOV	0.660 ± 0.174	0.744 ± 0.171	0.609 ± 0.164	0.720 ± 0.164	0.722 ± 0.184
RANK+MARKOV PATH	0.697 ± 0.153	0.754 ± 0.165	<i>0.634 ± 0.145</i>	<i>0.737 ± 0.166</i>	<i>0.750 ± 0.169</i>

Table 8: Performance comparison on five datasets in terms of pairs- F_1 -Scores. The best method for each dataset (i.e., a column) is shown in bold, the second best is shown in italic.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
RANDOM	0.266 ± 0.138	0.298 ± 0.136	0.246 ± 0.140	0.315 ± 0.127	0.295 ± 0.144
PERS TOUR	0.417 ± 0.343	0.643 ± 0.366	0.216 ± 0.265	0.468 ± 0.376	0.504 ± 0.354
PERS TOUR-L	0.359 ± 0.207	0.352 ± 0.162	0.266 ± 0.140	0.406 ± 0.238	0.333 ± 0.163
POI POPULARITY	<i>0.436 ± 0.259</i>	0.507 ± 0.298	0.316 ± 0.178	0.365 ± 0.190	0.384 ± 0.201
POI RANK	0.432 ± 0.250	<i>0.552 ± 0.313</i>	0.336 ± 0.198	0.511 ± 0.309	0.517 ± 0.295
MARKOV	0.417 ± 0.248	0.495 ± 0.296	0.288 ± 0.195	0.445 ± 0.266	0.407 ± 0.241
MARKOV PATH	0.399 ± 0.235	0.491 ± 0.297	0.297 ± 0.193	0.442 ± 0.260	0.406 ± 0.230
RANK+MARKOV	0.445 ± 0.264	0.531 ± 0.301	0.344 ± 0.215	0.492 ± 0.292	0.511 ± 0.302
RANK+MARKOV PATH	0.428 ± 0.248	0.521 ± 0.298	<i>0.338 ± 0.205</i>	<i>0.498 ± 0.293</i>	<i>0.513 ± 0.297</i>

A.3 Reproduced results on Windows

Note: As the calculations on the Melbourne dataset did take too much time, the calculations for this dataset were skipped for windows.

Table 9: Performance comparison on five datasets in terms of F_1 -Scores. The best method for each dataset (i.e., a column) is shown in bold, the second best is shown in italic.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
RANDOM	0.582 ± 0.134	0.617 ± 0.117	-	0.633 ± 0.097	0.622 ± 0.121
PERS TOUR	0.656 ± 0.223	0.801 ± 0.213	-	0.686 ± 0.231	0.720 ± 0.215
PERS TOUR-L	0.651 ± 0.143	0.660 ± 0.102	-	0.686 ± 0.137	0.643 ± 0.113
POI POPULARITY	0.701 ± 0.160	0.745 ± 0.166	-	0.663 ± 0.125	0.678 ± 0.121
POI RANK	<i>0.699 ± 0.153</i>	<i>0.770 ± 0.174</i>	-	0.745 ± 0.173	0.753 ± 0.169
MARKOV	0.645 ± 0.169	0.725 ± 0.167	-	0.697 ± 0.150	0.669 ± 0.151
MARKOV PATH	0.677 ± 0.149	0.735 ± 0.170	-	0.706 ± 0.150	0.688 ± 0.138
RANK+MARKOV	0.661 ± 0.174	0.744 ± 0.171	-	0.720 ± 0.164	0.722 ± 0.184
RANK+MARKOV PATH	0.698 ± 0.153	0.754 ± 0.165	-	<i>0.737 ± 0.166</i>	<i>0.750 ± 0.169</i>

Table 10: Performance comparison on five datasets in terms of pairs- F_1 -Scores. The best method for each dataset (i.e., a column) is shown in bold, the second best is shown in italic.

	Edinburgh	Glasgow	Melbourne	Osaka	Toronto
RANDOM	0.271 ± 0.153	0.298 ± 0.136	-	0.315 ± 0.127	0.308 ± 0.155
PERS TOUR	0.417 ± 0.343	0.643 ± 0.366	-	0.468 ± 0.376	0.504 ± 0.354
PERS TOUR-L	0.359 ± 0.207	0.352 ± 0.162	-	0.406 ± 0.238	0.333 ± 0.163
POI POPULARITY	<i>0.436 ± 0.259</i>	0.507 ± 0.298	-	0.365 ± 0.190	0.384 ± 0.201
POI RANK	0.431 ± 0.249	<i>0.552 ± 0.313</i>	-	0.511 ± 0.309	0.517 ± 0.295
MARKOV	0.417 ± 0.248	0.495 ± 0.296	-	0.445 ± 0.266	0.407 ± 0.241
MARKOV PATH	0.399 ± 0.234	0.491 ± 0.297	-	0.442 ± 0.260	0.405 ± 0.231
RANK+MARKOV	0.447 ± 0.264	0.531 ± 0.301	-	0.492 ± 0.292	0.511 ± 0.302
RANK+MARKOV PATH	0.429 ± 0.248	0.521 ± 0.298	-	<i>0.498 ± 0.293</i>	<i>0.514 ± 0.297</i>