12th IEEE International Conference
www.aict.info/2018

AICT 2018
INTERNATIONAL CONFERENCE

APPLICATION OF INFORMATION AND COMMUNICATION TECHNOLOGIES - AICT2018

17-19 October 2018, Almaty, Kazakhstan

CONFERENCE PROCEEDINGS

17-19 October 2018, Almaty, Kazakhstan

www.aict.info/2018

12th IEEE International Conference

www.aict.info/2018

AICT 2018

INTERNATIONAL CONFERENCE

APPLICATION OF INFORMATION AND COMMUNICATION TECHNOLOGIES - AICT2018

17-19 October 2018, Almaty, Kazakhstan

CONFERENCE PROCEEDINGS

2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)

IEEE Catalog Number: CFP1856H-PRT
ISBN: 978-1-5386-6467-4

Additional copies of this publication are available from
Curran Associates, Inc.
57 Morehouse Lane
Red Hook, NY 12571 USA
+1 845 758 0400
+1 845 758 2633 (FAX)
email: curran@proceedings.com

© Designed by the AICT2018 Publication Team, 2018.

IEEE Catalog Number: CFP1856H-PRT
ISBN: 978-1-5386-6467-4

SESSION 3. HIGH PERFORMANCE COMPUTING AND MACHINE LEARNING

Common Movement Prediction using Polynomial Regression

Timur Bakibayev
Almaty Management University
School of Engineering Management
Almaty, Kazakhstan
email: timurbakibayev@gmail.com

Akbota Kulzhanova
Al-Farabi Kazakh National University
Faculty of Information Technologies
Almaty, Kazakhstan
email: neverever1594@gmail.com

Abstract. **This paper describes an algorithm for geographical coordinates processing using polynomial regression to calculate the common behavior of a movement. The primary advantage of this algorithm is having a map of trajectories in any region. We have used Python programming language together with scikit-learn and matplotlib libraries for machine learning and visualization. In the end we try to predict several steps of the movement of all objects on the map.**
This work is the result of Master Thesis research.

Keywords—Movement Behavior; Movement Prediction; Trajectory; Machine Learning; Polynomial Regression

I. Introduction

Understanding animals' movement behavior may be crucial when studying spatial population processes. It is of no doubt that predicting animal behavior may help animal population and sometimes save human's lives (when dealing with roads). Among those who are interested in such research are the zoologists, who investigate the impact of vehicular traffic on the movement and distribution of animals. There are different approaches in working with movement data - State–space models [1], algorithmic approaches to combine trajectories into sub-trajectories, that we have tried before, and other.

The old studies were mainly based on point data. Progress in satellite systems and tracking devices allow collecting large databases of moving object trajectories, such as traffic data, hurricane tracking data and animal movement data. For example, to predict future hurricane trajectories, the most similar hurricanes to a target hurricane are detected by comparing the directions of hurricanes. Then the first and second difference of positions during their life time is used. [2]

For many years, weather researchers have used statistical analysis to study associations between weather events and environmental observations. The motivation was to be able to predict, as far as possible, the amount and intensity of hurricanes that may occur during a given hurricane season. Scientists often used basic graphic data, such as scatter diagrams and histograms. However, the inability of the basic data graphics to transmit complex data associations is well understood by the visualization community. [3]

Interest in processing such data is growing strongly in recent times. A typical task for data analysis is to find objects that moved the same way. Therefore, an effective clustering algorithm is necessary for the analysis of such problems. The work on the development of clustering algorithms for streaming and online settings is actively used at this time. [4]

Previously we have considered a "split-and-group" approach for clustering trajectories, that divides the trajectory into a set of line segments, and then groups similar segments into a single cluster. That algorithm was relatively slow and sophisticated.

At the moment, there is a new algorithm for cluster analysis that does not explicitly create clustering of the data set; but instead creates an extended database ordering that represents its density-based clustering structure. [5-6] This clustering contains information that is equivalent to clusters based on density, corresponding to a wide range of parameters. This is the universal basis for automatic and interactive cluster analysis.

This time we decided to use standard machine learning techniques, such as polynomial regression, to process our data for finding the common movement behavior of observed objects.

Scikit-learn is a well-known software module. The Python language, which contains a wide variety of various algorithms machine learning, allowing to solve the most non-standard applied tasks. This module is intended, first of all, for people, who are not specialists in the field of data analysis and machine training, as evidenced by the relative simplicity of "talking" names of all the structural elements of the module, the convenience of using them in aggregates, as well as their combinations with elements (objects and methods) of other Python program modules. [7]

II. Literature review

Trajectory analysis aroused great interest, including trajectories (migrating) animals, hurricane paths, and from an indexation point of view. The latter studied indexing, but not the detection of the pattern; among the first, attention was focused on clustering and distance functions on trajectories. In [8] they seek to identify so-called templates for identifying anomalies and helping the taxi operating system with the ultimate goals of fuel maximization, efficiency and passenger satisfaction. Disclosure of the rules governing collective taxi actions is a difficult task due to the many factors that influence a person's decision to take a specific action. In this paper, 10,000 trajectories generated by anonymous taxi drivers are used to measure social and economic activity. Given the set of GPS coordinates for each taxi, every few minutes and its status ("full" or "empty") reveals a general trend towards a taxi.

Another important drawback of the current forecasting methods is the fact that most of these methods do not use historical data. Such methods are considered obsolete and not suitable for forecasting the location on the Internet due to the long processing time and computational cost of these methods. In [9] a probabilistic model of the unknown position of a moving object is constructed on the basis of historical data collected from other objects moving in the same area. Thus, frequent trajectories of objects representing popular traffic routes are detected, and then on frequent trajectories of movements, regulations. To predict the location of a moving object for which only a part of the motion history is known, the history of the movement of the object from the motion rules database is evaluated to find possible locations of an object.

When clustering of continuously moving objects, it is necessary to consider not only the current movements but also their expected movements. As it is noted in [10], it is also important to track the changes in clustering structure, and they propose a new scheme to maintain data bubbles incrementally.

III. Algorithmic approach

Our previous work was focused on the results of Jae-Gil Lee where they suggest viewing the drawing. The figure in [8] clearly shows that there is a general behavior indicated a thick arrow, in a dotted rectangle. But, grouping these trajectories as a whole cannot detect the normal behavior, since directions vary completely. To eliminate this problem, it is suggested to break the trajectory into a set of line segments and then group similar segments. As a result, we could calculate the general trajectory (vector) for the group. [8]

We have proposed an algorithm for handing particular case of circular movement, helping to find the center and the radius of a round trajectory.

IV. The main idea of our research

Most of the studies for trajectories clustering handle moving objects separately. We do the same for preprocessing, but in the end, we come up with the split data for smaller parts of trajectories that would be feed to a machine learning model.

During data preprocessing we convert single points into tiny lines that represent trajectories, where the length of the line corresponds to object's speed. Next, we add extra columns for building polynomials.

We are given coordinates of each object at each time moment, so the input table looks as follows:

1. Object id
2. Time (integer field, autoincrement)
3. X coordinate
4. Y coordinate

The picture below (Fig. 1) shows us a simple visualization of each object's trajectory.

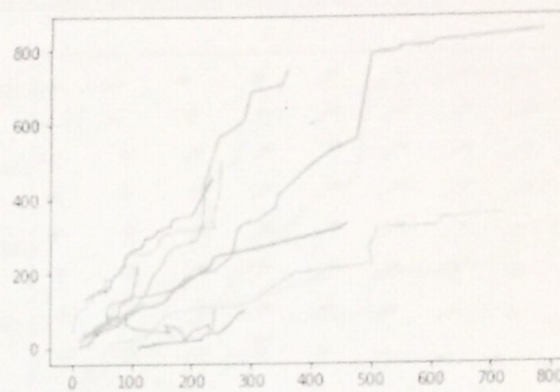

Figure 1. Input data: ten separate objects trajectories.

As mentioned before, we add extra columns for data movement and polynomials, so after preprocessing we have the following additional fields:

1. Next x (nx): x coordinate at next moment
2. Next y (ny): y coordinate at next moment
3. dx: delta X = nx - x
4. dy: delta Y = ny - y

```
dx = []; dy = []; nx = []; ny = []
for index, point in df.iterrows():
    next_point = df[(df.object == point["object"]) &
                    (df.time == point["time"] + 1)]
    if len(next_point) > 0:
        next_point=next_point.iloc[0]
        dx.append(next_point["x"]-point["x"])
        dy.append(next_point["y"]-point["y"])
        nx.append(next_point["x"])
        ny.append(next_point["y"])
    else:
        dx.append(0)
        dy.append(0)
        nx.append(point["x"])
        ny.append(point["y"])
df["nx"] = nx; df["ny"] = ny; df["dx"] = dx; df["dy"] = dy
```

Figure 2. The listing for data preprocessing

We may observe the splitting of each step on Fig. 3.

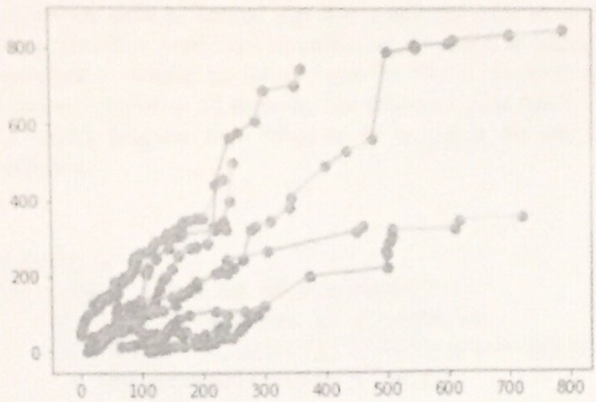

Figure 3. Splitting point data into smaller trajectories.

So, the data is almost ready for feeding to a linear regression model of scikit-learn, the last step is to add polynomial fields and run the learning process.

Regression model should return two fields: `x` and `y`. Our attempts to combine these two fields failed, so we have split our learning process into two parts:

```
1. lin_x = LinearRegression()
2. lin_y = LinearRegression()
3. lin_x.fit(df[["x","y","x2","y2",
   "xy"]],df["dx"])
4. lin_y.fit(df[["x","y","x2","y2",
   "xy"]],df["dy"])
```

that for our case gave us the following coefficients:

```
1. lin_x: [0.05523343, 0.05208924,
   -0.00024617, -0.00024446, 0.00044677]
2. lin_y: [0.08220028, 0.11533574,
   -0.00035998, -0.00032135, 0.0004776 ]
```

Here it is important to observe that all model is decreased to th ese 10 coefficients together with two intercepts (12 total). Wh en dealing with a large amount of historical data, the latter can be processed and represented by these numbers.

Next, we try to predict the common behavior movement by building a matrix of arrows that show the common behavior of all objects in each rectangular region. Since we have reconstructed the data into smaller line segments, we did not provide the object id and time fields to our linear regression model. This means we have a common behavior in each point regardless of the object.

We can see on Fig. 4 that our regression model learned the common behavior quite well. At the regions with missing data the movement is inverted, but since there was no data at all it is hard to predict the movement.

So, we have tried (Fig. 5) to check how the movement behavior changes with adding more data at the missing regions. Not only

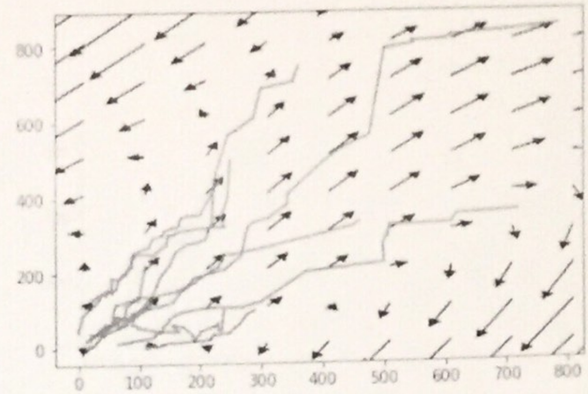

Figure 4. The results of polynomial regression.

we have filled the lower region of our map, we decided to see what happens if that object stays for longer time (20 steps) at one point (680,240) and then slowly moves forward.

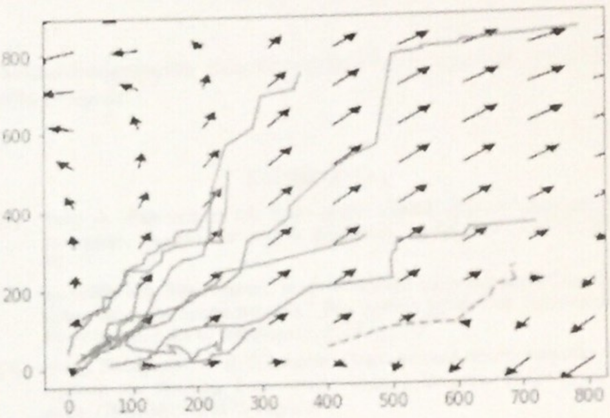

Figure 5. Adding a relatively slow object on a map.

We can observe much shorter lines, as the common speed went down. Also, we can see that the arrows at the position of a new trajectory point in the correct direction. Even at the top-left corner, arrows do not look that inverted any more.

V. Movement Prediction

Our next goal is to predict the movement of all objects once we have our model fit. In order to do so, we need to capture each object (currently we have 11) and make prediction step by step remembering the last coordinate (Fig. 6). Once we fixed the last positions of each object on a map, we try to simulate the steps, adding the predicted `dx` and `dy` values. On Fig. 7, we can clearly see that the prediction follows the common behavior of all objects.

VI. Why This Approach?

Why is that important to predict the movement based on common behavior?

Assume we have an animal that has submitted only 10 to 20 points. This data would not be sufficient for machine learning algorithms to predict its future behavior. But if we have the common behavior of all animals, that becomes more feasible – our model projects that behavior to a single animal for prediction.

```
objects = {}
for index, point in df.iterrows():
    if point.object not in objects or \
    objects[point.object]["time"]<point.time:
        objects[point.object] = {
            "x": point.x,
            "y": point.y,
            "time": point.time,
        }
```

Figure 6. Finding the last positions of each object.

Also, if the common behavior is repetitive by days, one could try collect data for one year and try our model for simplifying daily data down to 12 float numbers (5 for x, 5 for y coordinates, two intercepts). Given simplified model, it may be possible to make predictions of common behavior by days. That may be animal movement behavior by days, though hurricane movement would be much more difficult to predict. This approach would not work on urban traffic or any other kind of complex data, since it is absolutely impossible to describe cars movement using a polynomial.

Conclusion

Simple polynomial regression model gave us better results than we had before, when used a pure algorithmic approach. Though algorithmic approach can deal with more complex behavior, smarter machine learning models, like Random Forest Regressor, can also be used, increasing the degree of the polynomial.

Machine Learning approach wins when we need to predict movement based on historical data, and this can be crucial when selecting a tool.

The consequence of this will be that it is pointless to chase these data beyond accuracy with complex, massive methods. That is, everything that could be caught in this data, and so it was caught already by a polynomial model. This tells us that sometimes the data is not as complex as it seems, and very quickly it is possible to come to the actual maximum of what can be squeezed out of them. Moreover, from the greatly reduced data by means of the selection of characteristics, our accuracy has not actually suffered.

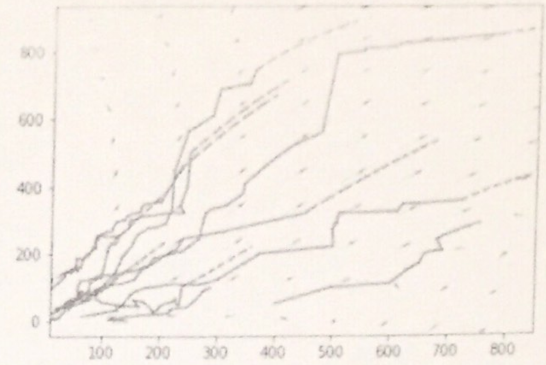

Figure 7. Movement prediction.

So, our solution for these data turned out to be not only simple, but also compact. As a result, we got a general direction of the movement of objects (animals).

Acknowledgements. *Research is funded by national grant No. BR05236644.*

References

[1] Toby A. Patterson et al. State–space models of individual animal movement. Trends in Ecology & Evolution , Volume 23 , Issue 2 , 87 – 94.

[2] Kordmahalleh, Mina Moradi, et al. "Hurricane trajectory prediction via a sparse recurrent neural network." *Proceedings of the 5th International Workshop on Climate Informatics.* 2015, pp 2-3.

[3] Hong, Seungkyun, et al. "Globenet: Convolutional neural networks for typhoon eye tracking from remote sensing imagery." *arXiv preprint arXiv*:1708.03417 (2017). Pages 2-4

[4] Gaffney, S. and Smyth P. Trajectory Clustering with Mixtures of Regression Models. ACM SIGKDD *International Conference on Knowledge Discovery and Data Mining.* 1999, pp. 267-272.

[5] Ankerst, M., Breuning M. M., Kriegel, H. -P и Sander, J. OPTICS: Ordering Points to Identify the Clustering Structure. *ACM SIGMOD International Conference on Management of Data.* 1999, pp. 102-123.

[6] Chen, L., Ă Ozsu, M. T., and Oria, V. Robust and Fast Similarity Search for Moving Object Trajectories. In Proc. *ACM SIGMOD International Conference on Management of Data.* 2005, pp 45-87.

[7] Raschka, Sebastian. *Python machine learning.* Packt Publishing Ltd, 2015. Pages 313- 314

[8] Matsubara, Yasuko, et al. "F-trail: Finding patterns in taxi trajectories."*Pacific-Asia Conference on Knowledge Discovery and Data Mining.* Springer, Berlin, Heidelberg, 2013. Pages 3-4.

[9] Morzy, Mikołaj. "Mining frequent trajectories of moving objects for location prediction."*International Workshop on Machine Learning and Data Mining in Pattern Recognition.* Springer, Berlin, Heidelberg, 2007. Pages 7-9.

[10] S. Nassar, J. Sander, and C. Cheng, "*Incremental and Effective Data Summarization for Dynamic Hierarchical Clustering,*" *Proc. ACM SIGMOD Int'l Conf. Management of Data* (SIGMOD '04), 2004. Pages 467-478

[11] Jae-Gil Lee, Jiawei Han et al. Trajectory Clustering: A Partition-and-Group Framework. Proceedings of the 2007 ACM SIGMOD international conference on Management of data. Pages 593-604