In *Proceedings of the Ninth ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS'16)*, N. Ronald, G. Thakur, and S. Winter, eds., pages 25–30, Burlingame, CA, October 2016.

Time Series Clustering of Weather Observations in Predicting Climb Phase of Aircraft Trajectories

Samet Ayhan * Hanan Samet
Department of Computer Science
University of Maryland
College Park, MD 20742
{sayhan, hjs} @cs.umd.edu

ABSTRACT

Reliable trajectory prediction is paramount in Air Traffic Management (ATM) as it can increase safety, capacity, and efficiency, and lead to commensurate fuel savings and emission reductions. Inherent inaccuracies in forecasting winds and temperatures often result in large prediction errors when a deterministic approach is used. A stochastic approach can address the trajectory prediction problem by taking environmental uncertainties into account and training a model using historical trajectory data along with weather observations. With this approach, weather observations are assumed to be realizations of hidden aircraft positions and the transitions between the hidden segments follow a Markov model. However, this approach requires input observations, which are unknown, although the weather parameters overall are known for the pertinent airspace. We address this problem by performing time series clustering on the current weather observations for the relevant sections of the airspace.

In this paper, we present a novel time series clustering algorithm that generates an optimal sequence of weather observations used for accurate trajectory prediction in the climb phase of the flight. Our experiments use a real trajectory dataset with pertinent weather observations and demonstrate the effectiveness of our algorithm over time series clustering with a *k-Nearest Neighbors (k-NN)* algorithm that uses *Dynamic Time Warping (DTW)* Euclidean distance.

CCS Concepts

•Information systems → Data analytics; Clustering;
•Computing methodologies → Dynamic programming for Markov decision processes;

Keywords

Time Series Clustering; Aircraft Trajectory Prediction; Predictive Analytics; Hidden Markov Model

*Senior Engineer at Boeing Research & Technology

IWCTS'16, October 31-November 03 2016, Burlingame, CA, USA

DOI: http://dx.doi.org/10.1145/3003965.3003968

1. INTRODUCTION

Trajectories are usually discussed for cars along roads [28] with an emphasis on queries (e.g., [20, 21, 24, 27]), Here we are interested in aircraft trajectory prediction which is a crucial process for Air Traffic Control (ATC) as the more accurate and reliable the predicted trajectories the more safe and efficient the deconfliction of flights. In order for ATC to attain this goal, the trajectory prediction tools should be able to provide high levels of accuracy for an adequate horizon of time. Although prior trajectory prediction research and development activities have been able to meet the challenge to some extent, dealing with congested airspace and environmental factors still remains the challenge when a deterministic approach is used in the trajectory prediction process [2, 4, 6, 14]. Hence due to the uncertainties contributing to trajectory prediction errors, we take a stochastic approach to address these issues. The past efforts addressing aircraft trajectory prediction using probabilistic methods are similar to our approach in a way that they aim at modeling uncertainties to describe potential changes in the future trajectory of an aircraft. However, many of them [17, 18, 19, 29] either lack empirical validation or use a simulated set of aircraft trajectories instead of actual track data in their evaluations.

We define airspace as a set of spatio-temporal cuboids where each cuboid is considered as an atomic unit. Weather parameters such as temperature, wind speed, wind direction, and humidity considered homogeneous within the cuboid during a period of time. Other parameters that describe the cuboid include its center coordinates along with a time stamp. These spatio-temporal cuboids form the overall airspace. Next, we adapt historic raw trajectories to the cuboids. Using adapted trajectories in the form of 4D joint cuboids, we train our model with the historical data and choose a state sequence that best explains the current weather observations. Due to the nature of interconnected cuboid centroids forming a trellis, we use the Viterbi algorithm [32] to efficiently generate the optimal trajectory by joining the multiple segments together, where one segment is only dependent on the previous segment. However, the Viterbi algorithm requires input observations, which are unknown, although the weather parameters overall are known for the airspace.

We use a time series clustering approach to this problem. We basically perform time series clustering on the current weather observations for the cuboid centroids that were traversed by historic trajectories. Our time series clustering algorithm computes a cost value for each weather observation based on its frequency per time period, and ranks them. The process yields a matrix, where each matrix element corresponds to a weather observation with its cost. Using DP, our algorithm computes the optimal sequence of weather observations, a set with the minimum cumulative cost that satisfies the *continuity* constraint. We finally feed the output cluster into the Viterbi process to probabilistically generate the best sequence of trajectory segments.

Our experiments use a real trajectory dataset with pertinent weather observations and demonstrate that our algorithm outperforms time series clustering with a *k-Nearest Neighbors (k-NN)* algorithm that uses DTW Euclidean distance [13]. The results suggest that our algorithm can be of significant value for predicting aircraft trajectories even during the climb phase of the flight. Trajectory prediction for the climb and descent phases of flight is difficult because of the uncertainty of rates of climb and descent due to the effects of outside air temperature on engine power and therefore climb/descent rates.

In summary, the contributions of this paper are as follows:

- We propose a novel time series clustering algorithm that is based on a cost value computed by the frequency of weather observations at each time period. Next, we feed the series of cluster centroids as observations into our aircraft trajectory prediction system that is based on a stochastic model, HMM to predict trajectories for the climb phase of a flight.
- We conduct experiments based on actual track data and weather observations and demonstrate that our time series clustering algorithm outperforms time series clustering with *k-NN* algorithm that uses DTW Euclidean distance by yielding cluster centroids that result in more accurate trajectory predictions for the climb phase of a flight.

The rest of the paper is organized as follows. Section 2 introduces the aircraft trajectory prediction system. Section 3 presents generic time series clustering and discusses our algorithm. Section 4 details experiments, and Section 5 concludes the paper and outlines some future work.

2. AIRCRAFT TRAJECTORY PREDICTION SYSTEM

This section presents the system that enables testing of our time series clustering algorithm. The prediction system used in this paper is an advanced version of [3] and distinguished from the previous version in the following respects:

- The new system uses a more granular reference grid and defines airspace as a set of spatio-temporal cuboids.
- The new system utilizes more extensive weather observations by including *humidity* to the existing set of *temperature, wind speed*, and *wind direction.*
- It employs our novel time series clustering algorithm to aid in the prediction process.

The system predicts aircraft trajectories in three steps: *i*) Training data processing, *ii*) Test data processing, and *iii*) HMM and Viterbi processing, detailed in [3].

In training data processing step, the system accurately fuses weather parameters for each sample point of a raw trajectory. To achieve this, the system uses the National Oceanic and Atmospheric Administration (NOAA) GFS Rapid Refresh (RAP) weather model's 3D grid network as the reference system. Due to fact that the lateral resolution of GFS RAP weather model's 3D grid network is $13km$, the system densifies the lateral resolution by splitting the original grids in half in east-west direction, resulting in a new 3D grid network formed by rectangles of $13km$ x $6.5km$. This new grid network allows the system to define an airspace of cuboids.

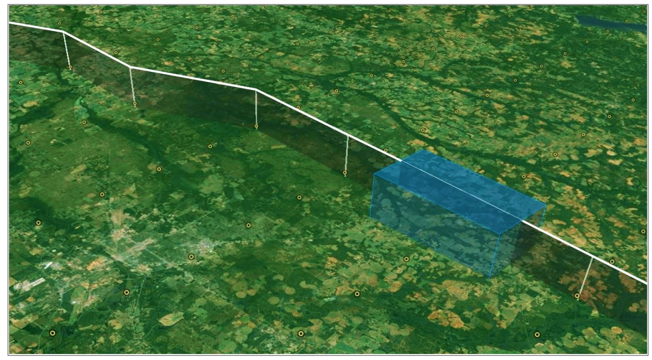

Figure 1: Illustration of a spatio-temporal cuboid formed around an aligned trajectory segment on the reference system in Google Earth.

To compute additional weather values for the derived grid points, the system performs *Sinusoidal* interpolation [1] on the original weather data. The weather parameters in resampled distinct buckets are fused with spatial data for each grid point. The process yields training data where each historical raw trajectory becomes a set of 4D joint spatio-temporal cuboids as illustrated in Figure 1.

Test data processing generates the input observations for the Viterbi process. Note that the observation sequence needs to be known in order for the system to compute the maximum probability of the optimal state sequence being generated by HMM. Although the reference points covering the entire airspace volume of interest are known, it is not known which of these should be fed into the Viterbi process. To identify the observation sequence, time series clustering is performed on the extensive observation set. Section 3 presents our novel time series clustering algorithm in detail.

In the final step, the system performs HMM and Viterbi processing. To achieve this, the system computes HMM parameters; states, transition probabilities, emission probabilities, and initial probabilities, denoted by $\lambda = \{S, A, B, \pi\}$, respectively. The next step in the process is to choose a corresponding state sequence that best explains the observation sequence using the Viterbi algorithm [32].

3. TIME SERIES CLUSTERING

Clustering is the general problem of partitioning a set of observations into a number of groups where the similarity is minimized within the group and the dissimilarity is maximized between the groups [16]. Han et al. classified clustering methods into five categories: partitioning, hierarchical, density-based, grid-based, and model-based methods [11].

A time series is a sequence of real numbers collected at regular intervals over a period of time. Time series clustering is a critical analysis technique which can be used as a preprocessing step for further data mining and it mostly relies on classic clustering methods, either by replacing the default distance measure with an alternative one or by transforming time series into static data so that classic clustering

Algorithm 1 OBSERVATIONPATHS

```
Input: observation matrix Om
Output: observationPaths op
Coc ← {}, Cmoc ← {}, Cm ← {}, Pm ← {} // columnwise observation counts, columnwise max observation counts, costs matrix, and pointers matrix, respectively
/* STEP 1: Compute counts */
for c ∈ Om.cols do
    Coc.o ← COUNT(o ∈ c)
end for
for c ∈ Om.cols do
    Cmoc.c ← MAX(Coc)
end for
/* STEP 2: Build pointers and costs matrices */
while c ≠ 0 (c ← |Om.cols|-1) do
    for r ∈ Om.rows do
        Cm[r][c] ← Cmoc[c] - COUNT[Om[c]]
        if c ≠ |c_{n−1}| then
            if r = 0 then
                mn ← MIN(Cm[r][c+1], Cm[r+1][c+1])
                Cm[r][c] ← Cm[r][c] + mn
                if mn = Cm[r][c+1] then
                    Pm[r][c].concat(r, Om[r][c+1])
                end if
                if mn = Cm[r+1][c+1] then
                    Pm[r][c].concat(r+1, Om[r+1][c+1])
                end if
            else if r = |Om.rows|-1 then
                mn ← MIN(Cm[r-1][c+1], Cm[r][c+1])
                Cm[r][c] ← Cm[r][c] + mn
                if mn = Cm[r-1][c+1] then
                    Pm[r][c].concat(r-1, Om[r-1][c+1])
                end if
                if mn = Cm[r][c+1] then
                    Pm[r][c].concat(r, Om[r][c+1])
                end if
            else
                mn ← MIN(Cm[r-1][c+1], Cm[r][c+1], Cm[r+1][c+1])
                Cm[r][c] ← Cm[r][c] + mn
                if mn = Cm[r-1][c+1] then
                    Pm[r][c].concat(r-1, Om[r-1][c+1])
                end if
                if mn = Cm[r][c+1] then
                    Pm[r][c].concat(r, Om[r][c+1])
                end if
                if mn = Cm[r+1][c+1] then
                    Pm[r][c].concat(r+1, Om[r+1][c+1])
                end if
            end if
        end if
    end for
    c ← c-1
end while
/* STEP3: Compute observation paths */
procedure ObsPathsDp((psf, r, c, cc))
    Procedure is detailed in the next algorithm
end procedure
ic ← s ∈ MIN(Cm.cols[0])
for ic ∈ ic[0] do
    p ← [Om[ic][0]]
    ObsPathsDp(p, ic, 0, Cm[ic][0])
end for
```

Algorithm 2 OBSERVATIONPATHS DP

```
procedure ObsPathsDp((psf, r, c, cc))
Input: pathSoFar psf, row r, colmn c, cumulativeCount cc
Output: observationPaths op
    if c = |Om.cols|-1 then
        return op ← psf, (c + Cm[r][c])
    end if
    for p ∈ P_m[r][c] do
        psf.concat(p[1])
        ObsPathsDp(psf, p[0], c+1, cc + Cm[r][c])
        psf.pop(-1)
    end for
end procedure
```

methods can be directly used. Liao et al. [16] classified time series clustering approaches into three major categories: *i)* raw-based [15, 23], *ii)* feature-based [10, 31], and *iii)* model-based [9] approaches, depending on whether they work directly with raw data, indirectly with features extracted from the raw data, or indirectly with models built from the raw data. In this paper, we follow a raw-based approach.

Although many clustering criteria to identify the similarity and dissimilarity have been proposed, the minimum within the cluster sum of squared distances is the most commonly used measure [12]. To represent a cluster, a centroid is used that is defined as the data point that minimizes the sum of squared distances to all other points. Hence, using the distance measure is critical in computing centroids. Finding such a centroid is known as the Steiner's sequence [25]. The centroid is computed with the arithmetic mean property, when Euclidean distance is used [7]. In the realm of time series clustering, DTW is the most widely used measure to compare time-series sequences with alignment [23]. Hence, we evaluate our time series clustering algorithm by comparing it with the state-of-the-art *k-NN* algorithm that uses DTW Euclidean distance. Unfortunately, using an arithmetic mean to find a single representative data point for each time interval along the time series often doesn't generate an accurate centroid as we present in Section 4.

Now, we propose a novel time series clustering algorithm for multivariate weather observations of the same length to aid in the aircraft trajectory prediction process.

Suppose we have a time series of weather observations between departure and arrival airports for the same flight over a period of time. We can represent this time series with an n x m matrix, where each row contains a set of periodic weather observations recorded along the aircraft's trajectory as the flight progresses. As the flight takes place the next time, a new row is added to the matrix. Assuming that the flight occurs once a day, each matrix cell corresponds to an observation recorded at a particular time interval during the flight in a particular day. Note that each observation is composed of the major weather parameters affecting the aircraft trajectory, that is *temperature, wind speed, wind direction*, and *humidity*. Given those observations, we want to trace an optimal weather observation path that best represents the underlying observation set. This objective translates to finding a single representative centroid for each time interval and optimally concatenating them as we move forward.

Formally, given n observation sets, each with length of m, $O_1 = [o_{1,1}, o_{1,2}, ..., o_{1,m}]$, $O_2 = [o_{2,1}, o_{2,2}, ..., o_{2,m}]$, ... $O_n = [o_{n,1}, o_{n,2}, ..., o_{n,m}]$, we want to come up with the op-

In *Proceedings of the Ninth ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS'16)*, N. Ronald, G. Thakur, and S. Winter, eds., pages 25–30, Burlingame, CA, October 2016.

timal observation path, $OOP = [o_{i,1}, o_{j,2}, ..., o_{k,m}]$, which is subject to *continuity* constraint that is defined as:

Continuity: Given $o_{i,j} = (a, b)$, then $o_{i,j-1} = (a', b')$, where $a - a' \geq 0$ and $b - b' \geq 0$. This restricts the allowable steps in the observation path to adjacent cells (including diagonal adjacent cells). There are exponentially many observation paths that satisfy the *continuity* constraint. However, we are only interested in the path, OOP that minimizes the cumulative cost denoted by

$$OOP = \arg\min_{c} \sum_{i=1}^{m} c_i$$

where cost of a cell $c_{i,j}$ is computed based on the pertinent observation's frequency at that time period.

To solve the optimization problem, we perform preprocessing by computing: *i*) Column-wise observation counts Coc, and column-wise maximum observation counts $Cmoc$, *ii*) A costs matrix Cm of size n x m where each matrix cell corresponds to a cost value, and a pointers matrix Pm of size n x m where each matrix cell corresponds to the adjacent observation (including diagonally adjacent), the original observation can connect to. Next, we perform Dynamic Programming (DP) in reverse order *i.e. beginning with the last column* to iteratively select the next weather observation that satisfies both conditions, *continuity* and *minimum cumulative cost* at each time period of the series.

Algorithm 1 illustrates the preprocessing steps line 1–46. In line 49–53, Algorithm 1 iteratively calls the ObsPathsDp procedure detailed in Algorithm 2 to dynamically find the next optimal sequence. Feeding the output cluster centroids into the Viterbi algorithm finalizes the process to generate the best sequence of trajectory segments.

4. EXPERIMENTS

In this section, we describe the experiments performed using our algorithm versus *k-NN* clustering with DTW on the weather observations along the time series.

Our experiments used real trajectory and weather data: The Delta Airlines' flight DAL2173, departing from Atlanta International Airport and arriving at Miami International Airport was studied for the period of May 2010 through December 2015. The dataset has a total of 1624 trajectories and 183797 points. Figure 2 shows 3D raw trajectories.

The main data sources for our Aircraft Trajectory Prediction System are the FAA's Aircraft Situation Display to Industry (ASDI) and NOAA's RAP data, detailed in [3]. Before training data processing, time series data interpolation and filtering processes took place that reduced the number of historical recordings from 183797 to 137689. In the training data processing, we split the weather parameters into buckets as shown in Table 2. Next, the following HMM parameters were computed:

- 7292 distinct states, S were generated.
- A sparse transition matrix, A of size 7292 x 7292 = 53173264 was generated.
- An emission matrix, B of size 495 x 7292 = 3609540 was generated.
- An initial matrix, π of size 38 x 1 was generated.

To evaluate our prediction system, we performed bootstrapping by drawing many trajectory samples with replacement from the historical trajectories. The trajectory sample

Figure 2: 3D raw trajectories of flight DAL2173 for the period of May 2010 through December 2015 in Google Earth. Climb phase is colored white, cruise and descent phases are colored gray.

for May 18, 2015 was chosen at random to be used as an example of the process for trajectory samples.

In test data processing, we performed time series clustering, using DTW and our algorithm. The input to the process was 1206 time series of 78 weather observations, where each observation contained *temperature, wind speed, wind direction*, and *humidity* parameters. Our first round evaluation tested the *k-NN* clustering with k=1 that used DTW Euclidean distance for the similarity measure. The resulting set of cluster centroids for the first round evaluation identified by weather parameters defined the first observation sequence Y_{s1}. The second round evaluation tested our time series clustering algorithm with the identical input. This defined the second observation sequence Y_{s2}. As the final step of the process, we fed the first and second observation sequences, Y_{s1}, Y_{s2}, respectively along with pertinent HMM parameters into the Viterbi algorithm, one at a time. The algorithm returned the optimal state sequence with the maximum probability per observation sequence.

Here, we present our evaluation results. Our evaluation was based on bootstrapping by drawing 7 trajectory samples with replacement from the historical trajectories. This way, we performed two series of comparisons:

- We compared the cluster centroids generated by both time series algorithms against the ground truth, the weather observations along the sample trajectories.
- We compared climb phases of the predicted trajectories aided by both algorithms against the ground truth, flight DAL2173's aligned sample trajectories.

Then, we rank ordered the means to estimate the 2.5 and 97.5 percentile values for 95% CI. Time series clustering with *k-NN* algorithm that uses DTW Euclidean distance correctly found 34 centroids on the average out of full 78 original centroids formed by weather observations *temperature, wind speed, wind direction,* and *humidity* of ground truth. This corresponds to 43.6% accuracy. Time series clustering with our algorithm correctly found 55 centroids on the average out of full 78 original centroids formed by

Table 1: Mean and standard deviation values for cross-track and vertical errors for flight DAL2173 on trajectory samples.

k-NN		Our algorithm	
Mean error (μ)			
$\mu(e_{cross})$	$\mu(e_{vert})$	$\mu(e_{cross})$	$\mu(e_{vert})$
$1.490nm$	$1697.042ft$	$0.982nm$	$259.588ft$
Standard deviation (σ)			
$\sigma(e_{cross})$	$\sigma(e_{vert})$	$\sigma(e_{cross})$	$\sigma(e_{vert})$
$3.100nm$	$1299.931ft$	$3.296nm$	$653.612ft$

weather observations of ground truth, which corresponds to 70.5% accuracy. Hence, our time series clustering algorithm outperformed time series clustering with a *k-NN* algorithm that uses DTW Euclidean distance by 61.8%, in accuracy.

Figure 3 shows a lateral view of the actual aligned trajectories in white overlaid on top of spatio-temporal cuboids for the flight's climb phase on May 18, 2015. On the left, cuboids in red were generated upon time series clustering using *k-NN* with DTW algorithm. On the right, cuboids in blue were generated upon time series clustering using our own algorithm. The predicted trajectory in the form of spatio-temporal cuboids in red on the left substantially deviates from the aligned trajectory in white as soon as the aircraft departs. The deviation from the aligned trajectory during the departure is relatively less significant for the spatio-temporal cuboids in blue on the right. Both sets of spatio-temporal cuboids in red and blue align well with the actual trajectories in white as the flight progresses.

Our quantitative evaluation is based on trajectory prediction accuracy metrics as outlined in [22]. The cross-track error, denoted by e_{cross} is computed based on the actual position of the aircraft AC, predicted trajectory segment containing the previous predicted position TR_1, current predicted position TR_2, and the angle θ between the two vectors, $|\overrightarrow{TR_1AC}|$ and $|\overrightarrow{TR_1TR_2}|$. Hence, $e_{cross} = |\overrightarrow{TR_1AC}| sin\theta$. The vertical error represents the difference between the actual altitude of the aircraft AA and predicted altitude of the aircraft PA. Hence, $e_{vert} = AA - PA$.

Table 1 captures mean error μ and standard deviation σ values for the entire trajectory samples. The first two columns correspond to cross-track and vertical errors generated upon time series clustering using *k-NN* with DTW algorithm. The last two columns correspond to cross-track and vertical errors generated upon time series clustering using our own algorithm. Note that both the cross-track and vertical errors are signed errors. However, the signs of the errors are omitted in the computation of the mean values along the climb phase of the trajectory.

Figure 4 illustrates the histograms for cross-track errors for flight DAL2173's climb phase on May 18, 2015. On the left, the graph in red illustrates the histogram generated upon time series clustering of *k-NN* with DTW algorithm. On the right, the graph in blue illustrates the histogram generated upon our own algorithm. The area of the histograms is an indication of overall performance.

5. CONCLUSIONS

In this paper, we presented a novel time series clustering algorithm that generates an optimal sequence of weather observations used in accurate trajectory prediction for the climb phase of the flight. Our algorithm computes a cost value for each weather observation and ranks them. The process generates a matrix, where each matrix element corresponds to a weather observation with its cost value. Using *DP*, the algorithm computes the optimal sequence of weather observations that satisfy the continuity constraint. We evaluated our algorithm and demonstrated its effectiveness over time series clustering with *k-NN* algorithm.

Table 2: Buckets for weather parameters

No	Bucket	Value
Temperature ($temp$)		
No	Bucket	Value ($kelvin$)
1	$temp \leq 220$	220
2	$220 < temp \leq 240$	240
3	$240 < temp \leq 260$	260
4	$260 < temp \leq 280$	280
5	$280 < temp \leq 300$	300
6	$300 < temp \leq 350$	350
Wind speed (ws)		
No	Bucket	Value ($knots$)
1	$ws \leq 30$	30
2	$30 < ws \leq 60$	60
3	$60 < ws \leq 90$	90
4	$90 < ws \leq 120$	120
5	$120 < ws \leq 150$	150
Wind direction (wd)		
No	Bucket	Value ($degrees$)
1	$wd \leq 45$	45
2	$45 < wd \leq 90$	90
3	$90 < wd \leq 135$	135
4	$135 < wd \leq 180$	180
5	$180 < wd \leq 225$	225
6	$225 < wd \leq 270$	270
7	$270 < wd \leq 315$	315
8	$315 < wd \leq 360$	360
Humidity ($hmdty$)		
No	Bucket	Value ($percent$)
1	$hmdty \leq 20$	20
2	$20 < hmdty \leq 40$	40
3	$40 < hmdty \leq 60$	60
4	$60 < hmdty \leq 80$	80
5	$80 < hmdty \leq 100$	100

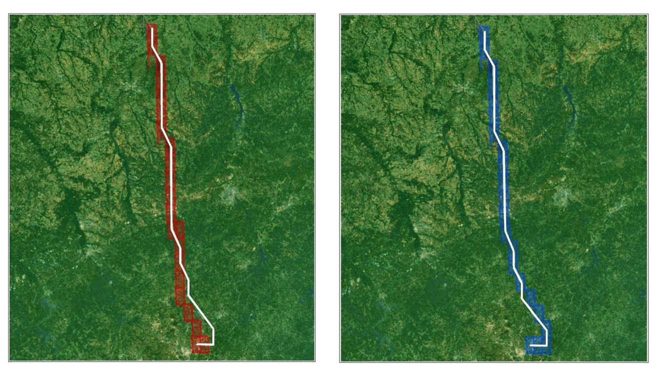

Figure 3: Actual flown lateral trajectory in white overlaid on top of spatio-temporal cuboids for flight DAL2173's climb phase on May 18, 2015. (left) k-NN with DTW, (right) our algorithm.

In *Proceedings of the Ninth ACM SIGSPATIAL International Workshop on Computational Transportation Science (IWCTS'16)*, N. Ronald, G. Thakur, and S. Winter, eds., pages 25–30, Burlingame, CA, October 2016.

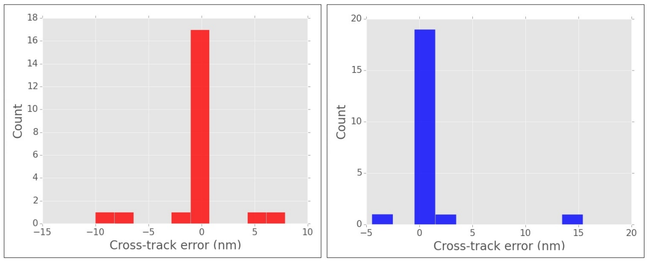

Figure 4: Histograms for flight DAL2173's climb phase on May 18, 2015. (left) k-NN with DTW, (right) our algorithm.

In the future, we plan to handle time series of weather observations with unequal length, utilize our algorithm in predicting full trajectories including cruise and descent phases of the flight, as well as look into adding a spatial browsing capability (e.g., [5, 8, 26]) and operating in a distributed environment [30].

6. ACKNOWLEDGMENTS

The authors would like to thank Srijan Kumar for his assistance with this research. This work was supported in part by the NSF under Grants IIS-12-19023 and IIS-13-20791.

7. REFERENCES

[1] K. E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, 1978.

[2] G. Avanzini. Frenet-based algorithm for trajectory prediction. *Journal of Guidance, Control, and Dynamics*, 27(1):127–135, January 2004.

[3] S. Ayhan and H. Samet. Aircraft trajectory prediction made easy with predictive analytics. In *Proc. of the 22nd Intl' Conf. on Knowledge Discovery and Data Mining*, San Francisco, August 2016.

[4] J. Benavides, J. Kaneshige, S. Sharma, R. Panda, and M. Steglinski. Implementation of trajectory prediction function for trajectory based operations. In *AIAA Atmospheric Flight Mechanics Conf.*, Atlanta, GA, 2014.

[5] F. Brabec and H. Samet. Client-based spatial browsing on the world wide web. *IEEE Internet Computing*, 11(1):52–59, January/February 2007.

[6] J. Bronsvoort, G. McDonald, J. Lopez-Leones, and H. Visser. Improved trajectory prediction for air traffic management by simulation of guidance logic and inferred aircraft intent using existing data-link technology. In *AIAA GNC Conf. and Exhibit*, Minneapolis, MN, August 2012.

[7] E. Dimitriadou, A. Weingessel, and K. Hornik. A combination scheme for fuzzy clustering. In *Proc. of the 2002 AFSS Intl' Conf. on Fuzzy Systems. Calcutta: Advances in Soft Computing*, London, UK, February 2002.

[8] C. Esperança and H. Samet. Experience with SAND/Tcl: a scripting tool for spatial databases. *Journal of Visual Languages and Computing*, 13(2):229–255, April 2002.

[9] S. Gao, Y. He, and H. Chen. Wind speed forecast for wind farms based on arma-arch model. In *Proc. Int'l Conf. on Sustainable Power Generation and Supply*, Nanjing, China, April 2009.

[10] C. Guo, H. Jia, and N. Zhang. Time series clustering based on ica for stock data analysis. In *Proc. 4th Intl' Conf. on Wireless Communications, Networking and Mobile Computing*, China, October 2008.

[11] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco, 2011.

[12] P. Hansen and B. Jaumard. Cluster analysis and mathematical programming. *Mathematical Programming*, 79(1-3):191–215, October 1997.

[13] E. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, March 2005.

[14] J. Krozel and D. Andrisani. Intent inference and strategic path prediction. In *AIAA GNC Conf. and Exhibit*, San Francisco, August 2005.

[15] M. Kumar, N. R. Patel, and J. Woo. Clustering seasonality patterns in the presence of errors. In *Proc. of the Eighth ACM SIGKDD Intl' Conf. on Knowledge Discovery and Data Mining*, Edmonton, Alberta, August 2002.

[16] T. W. Liao. Clustering of time series data-a survey. *Pattern Recognition*, 38(11):1857–1874, November 2005.

[17] Y. Liu and X. R. Li. Intent based trajectory prediction by multiple model prediction and smoothing. In *AIAA GNC Conf. and Exhibit*, New Orleans, LA, January 2015.

[18] I. Lymperopoulos, J. Lygeros, and A. Lecchini. Model based aircraft trajectory prediction during takeoff. In *AIAA GNC Conf. and Exhibit*, Keystone, CO, August 2006.

[19] S. Mondoloni. A multiple-scale model of wind-prediction uncertainty and application to trajectory prediction. In *6th AIAA Aviation Technology, Integration and Operations Conf.*, Wichita, KS, September 2006.

[20] S. Nutanong, E. H. Jacox, and H. Samet. An incremental Hausdorff distance calculation algorithm. *PVLDB*, 4(8):506–517, August 2011.

[21] S. Nutanong and H. Samet. Memory-efficient algorithms for spatial network queries. In *Proc. of the 29th IEEE Int'l Conf. on Data Engineering*, pages 649–660, Brisbane, Australia, April 2013.

[22] M. Paglione and R. Oaks. Implementation and metrics for a trajectory prediction validation methodology. In *AIAA GNC Conf. and Exhibit*, Hilton Head, SC, August 2007.

[23] J. Paparrizos and L. Gravano. k-shape: Efficient and accurate clustering of time series. *SIGMOD Rec.*, 45(1):69–76, June 2016.

[24] S. Peng, J. Sankaranarayanan, and H. Samet. SPDO: High-throughput road distance computations on spark using distance oracles. In *Proc. of the 32nd IEEE Int'l Conf. on Data Engineering*, pages 1239–1250, Helsinki, Finland, May 2016.

[25] F. Petitjean and P. Gançarski. Summarizing a set of time series by averaging: From steiner sequence to compact multiple alignment. *Theory Computer Science*, 414(1):76–91, January 2012.

[26] H. Samet, H. Alborzi, F. Brabec, C. Esperança, G. R. Hjaltason, F. Morgan, and E. Tanin. Use of the SAND spatial browser for digital government applications. *Communications of the ACM*, 46(1):63–66, January 2003.

[27] J. Sankaranarayanan, H. Alborzi, and H. Samet. Distance join queries on spatial networks. In *Proc. of the 14th ACM Int'l Symposium on Advances in Geographic Information Systems*, pages 211–218, Arlington, VA, November 2006.

[28] J. Sankaranarayanan and H. Samet. Roads belong in databases. *IEEE Data Engineering Bulletin*, 33(2):4–11, June 2010.

[29] C. Schultz, D. Thipphavong, and H. Erzberger. Adaptive trajectory prediction algorithm for climbing flights. In *AIAA GNC Conf. and Exhibit*, Minneapolis, MN, August 2012.

[30] E. Tanin, A. Harwood, and H. Samet. A distributed quadtree index for peer-to-peer settings. In *Proc. of the 21st IEEE Int'l Conf. on Data Engineering*, pages 254–255, Tokyo, Japan, Apr. 2005.

[31] G. Verdoolaege and Y. Rosseel. Activation detection in event-related fmri through clustering of wavelet distributions. In *IEEE Intl' Conf. on Image Processing*, Hong Kong, September 2010.

[32] A. J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. on Information Theory*, 13(2):260–269, April 1967.