# A SEMI-LAZY FLIGHT TRAJECTORY PREDICTION FOR AIR TRAFFIC MANAGEMENT

**Group 10**


Ashwini Ravi (A0119980W)

Raghavendhra Balaraman (A0123443R)

Vishnu Gowthem Thangaraj (A0134525L)

# 1. INTRODUCTION

The Aviation industry is one of the largest and most essential service industries in the world. Stats[1] tell us that in the year 2014, approximately 3 billion people travelled by flights and that includes just passenger airlines. On an average, 102,465 flights in a single day! In several countries, the airlines industry is directly or indirectly responsible for contributing to a large part of the country's economic status. Clearly, the airlines industry is of paramount importance having social, economic and climatic implications on the world we live in. It is therefore imperative that the *safety of flight journeys* should be taken care of ,thus airline accidents should be kept to a minimum, if not totally avoided.

Amongst the various reasons for airline accidents, accidents due to weather are only behind human errors and mechanical failures [2]. Despite having extensive weather predictions from meteorological departments, approximately 15% of accidents every year are due to weather. To meet this growing demand and to predict the forthcoming trajectory dynamically is given a paramount importance to avoid airline accidents while ensuring safety and efficiency.

This raises to a very interesting machine learning problem, based on which we propose a semi lazy flight trajectory prediction system for in-flight air traffic management. The semi lazy paradigm combines the simplistic advantages of lazy methodologies such as clustering along with intense eager learning algorithms. Historical radar information is one of the most consistent record of the flight path. In this project, data mining algorithms are implemented to process the huge amount of historical radar data and to extract a typical trajectory. This typical trajectory and the dynamic weather information is used as the inferred intent information in the proposed algorithm to predict and recommend future trajectory points, thus avoids the hazardous area.

In this project, we propose a computationally effective analytical algorithm to predict and compute the flight trajectory points based on dynamic weather information.The rest of the explanation of this project is organized as follows: Section 6.2, describes how to acquire the typical trajectory after applying DBSCAN clustering algorithm.In Section 6.5,we introduce the concept of typical trajectory and explain the intent based algorithm to represent the improved flight intent.In section ,the performance of the proposed algorithm is tested , measured and compared against semi lazy paradigm and eager learning approach.

# 2. RELATED LITERATURE

In recent years, many scholars have launched the study of flight trajectory prediction. The prediction methods are categorized into three types based on the state of propagation, i.e., Nominal, Worst case and Probabilistic [3]. Nominal, a straightforward methods, predicts the future state based on the current state in single trajectory, whereas Worst case methods, predicts a range of movements and raises alarms if there is a conflict. Both these methods does not account to deviations that are caused by uncertainties. The trajectory points in probabilistic prediction method uses a probability density function considering the uncertainties and conflicts. Since, it is a non-intent based approach, prediction error could increase quadratically and does not guarantee accurate prediction when there are more than one flight maneuvers. An alternative method is an intent based prediction models [5] that uses aircraft intent and flight plans. However, the accuracy of the trajectory prediction relies on the dynamically changing nature of aircraft's intent and historical data which contains the information such as weather data, changes in flight plan and pilot's decision. By analyzing these information we could improve the accuracy of trajectory prediction in real time. Lots of recent studies analyze patterns in historical data using clustering. K-means method was used in aircraft monitoring [5] using the radar data. Frank R et al [6], used cluster analysis to perform comparison of trajectory points. ZengFu, used subtractive clustering [7] to propose dynamic track cluster algorithms. Fuzzy clustering method [8], was used by Taobo Wang to analyze the average centre trajectory during the arrival phase. All these methods are limited to using the clustering methods on historical data to analyze the flight data during arrival or departure phase rather than the complete trajectory. In this paper, we focus on predicting the trajectory points using a semi-lazy paradigm on the historical flight trajectory data.

# 3. PROBLEM STATEMENT

Despite the increasing technological advancement in the air traffic management, trajectory prediction still remains of high importance. There are still lots of ongoing research to use the real time big data to predict the future path of the trajectory.

Presently, all the variants of flight trajectory prediction algorithms are ground based , ie., the pilots still wait for the information to be transmitted from the Air Traffic Controller that are stationed at the airports. Advancements in real-time big data analysis are changing the course of flight as we know it. What if there exist a solution that could augment the pilot's decision-making process with the help of "Real time business intelligence"? A semi-lazy algorithm is proposed in this paper, which aims to deliver a real-time flight profile to the pilot, helping them make in-flight decisions more efficient and reliable on time.

In this paper, the semi-lazy data mining paradigm is studied and implemented to predict the trajectory of aircraft in-flight. A clustering algorithm is applied on the historical radar trajectory data to abstract a set of typical trajectories for the given source and destination airports. The typical trajectories, which are a subset of the historical data are now analysed using a intent-based model which includes dynamically changing weather conditions. The input flight plan is subjected to the given weather conditions and any conflicts are resolved by suggesting alternate route or deviation from the current flight path, obtained from the output of the intent based model.

| | |
|---|---|
| Data Source | Kaggle Flight Quest Challenge [9] |
| Data Size | 3.2 GB |
| # of historical radar flight trajectory information | 22 Million trajectory points |
| # of flight plan data | 0.9 Million |
| # of weather related information | 80,000 |

Table 1 : Dataset Details

# 4. DATASET

The dataset contains all the flight information & weather related information for domestic US flights during November 26, 2012 - December 9, 2012 (14 days). It is also important to note that the data is real-time, actual flight data collected by FlightStats for domestic US passenger airlines. Table 1 provides necessary insights about the data used in this project.

## 4.1. DATA ATTRIBUTES AND DESCRIPTION

The original data obtained from the data source contained separate flight and weather information per day. The raw data was preprocessed (discussed in the forthcoming topics) resulting in flight historical trajectory data, flight plan and weather data being consolidated and stored as a separate tables. These tables are used for our further analysis. A tabular representation of the schema diagram is given below.

| Flight History Table | Flight Plan Table | Air Sigmet Table |
|---|---|---|
| FlightHistoryID | FlightPlanID | AirSigmetID |
| UpdateTimeUTC | FlightHistoryID | TimeValidFromUTC |
| DepartureAirport | DepartureAirport | TimeValidToUTC |
| ArrialAirport | ArrialAirport | MovementDirectionDegrees |
| DepartureUTC | OriginalDepartureUTC | MovementSpeedKnots |
| ArrivalUTC | OriginalArrivalUTC | HazardType |
| Latitude | Latitude | HazardSeverity |
| Longitude | Longitude | AirSigmetType |
| | Ordinal | AirSigmetArea |

Table 2 : Table Schema

### 4.2. DATA STORAGE - MONGODB

To store the contents to a scalable database to facilitate easy querying , we loaded the dataset into SQLITE3, considering its light weight and easy portable nature. We then shifted to MYSQL - another popular open source relational DB choice.However the data was so large that querying took considerable time and reduced the performance. Hence, we chose **MongoDB,** a leading NoSQL databases, taking advantage of its JSON format documents, dynamic schemas and better performance due to indexing.

## 5. PROPOSED SOLUTION

### 5.1. FRAMEWORK ARCHITECTURE

The framework architecture of the semi-lazy paradigm is shown in the figure 1. The two components of the architecture includes DBScan Clustering and Intent-based model. The historical radar trajectory data along with the user input query are clustered using DBSCAN algorithm to abstract the typical trajectories. The abstracted typical trajectories used in the intent-based model to predict the trajectory points based on the AirSigmet region.

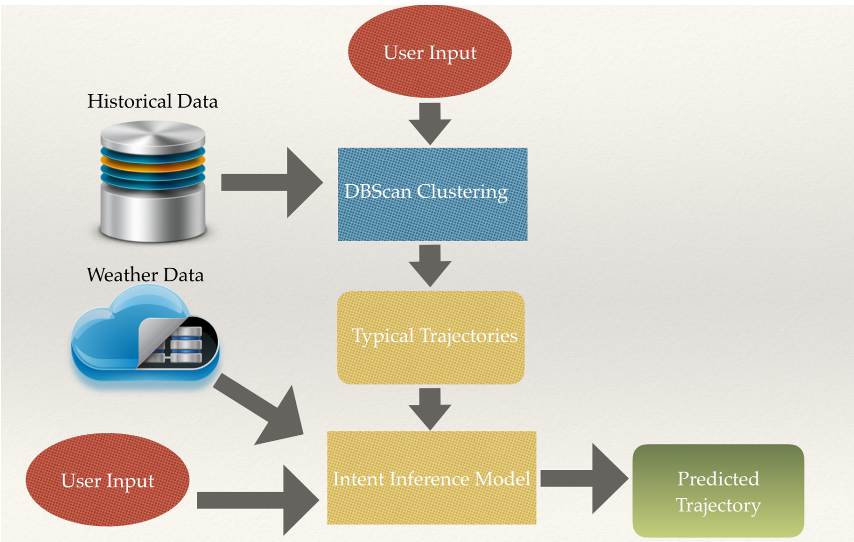



Figure 1 : Semi-lazy Framework Architecture

### 5.2. INPUT

The input for our semi-lazy paradigm is a set of latitude and longitude points in the below format representing a flight plan.

| Latitude | Longitude | Departure Airport | Arrival Airport | Start Time | End Time |
|---|---|---|---|---|---|

### 5.3. WEATHER INPUT - AIRSIGMET INFORMATION

There are various kinds of weather information available to passenger flights these days such as Airmets, Sigmets, PIREPS, TAF and METAR. Airmets (Airmen's Meteorological Information) and Sigmets (Significant Meteorological information), often known together as AirSigmets are predicted weather information that is available to aircrafts and are only valid for 4 hours. PIREPS (Pilot Reports) are weather forecasts transmitted to flights from radars during the journey whereas TAF (Terminal Aerodrome Forecast) and METAR (Meteorological Terminal Aviation Routine Weather report) are weather information at the ground level, that can often delay of affect landings and take offs. AirSigmets and PIREPS are both useful in our problem context to predict modified trajectories. Due to data insufficiency issues in PIREPS, we use AirSigmets. In a nutshell, the simplistic form of AirSigmets data considered are :

| AirSigmet Area | Hazard type | Hazard Severity |
|---|---|---|

### 5.4. OUTPUT

The output of the project involves a set of latitude and longitude points in the below format representing the updated flight plan trajectory.

| Original Trajectory | Predicted Trajectory | Speed Knots | Hazard Severity | Movement Direction Degrees |
|---|---|---|---|---|

# 6. IMPLEMENTATION

### 6.1. DATA PREPARATION

As mentioned earlier, the data being real-time flight data, needs to be preprocessed before it can be used for any type of analysis. As a part of data preprocessing, sampling should also be performed before computing the typical trajectories. Illegal historical records such as trajectories which have too few points and trajectories with unfilled flight number are removed to obtain a fairly cleaned and organized dataset.

### SAMPLING

Trajectories less than the input trajectory length are excluded from the dataset, to make use of clustering algorithm like DBSCAN, where each trajectory is signified as a vector. To compute the distance, all vectors should have the same number of points. Since all historical trajectories do not have the same length in dataset, it becomes essential to sample historical flight trajectory to have fixed length (equal to input trajectory length 'L') as well as the duration of the flight.

Sampling Formula

Pij' = Pij , j = round((j'*Li)/L), j'=1,2…,L

Sampling flight duration is performed by taking advantage of the time series functionality in a python library named ‘pandas’. Pandas are known for handling missing information by intelligent alignment and integration of data

### 6.2. CLUSTERING

DBSCAN (Density based Spatial Clustering of Application with Noise) algorithm is employed to cluster the historical flight trajectories that are closely connected to each other. DBSCAN algorithm is preferred over K-means for clustering historical trajectories as it fits well for high density spatio-temporal data and number of clusters for trajectories between source and destination airports are not known beforehand. A cluster satisfies the following properties:

1. Each trajectory inside the clusters are mutually “density connected”

2. If any historical trajectory is “density reachable” from any other trajectory within the cluster, it is also part of the cluster.

**DBSCAN clustering** algorithm requires two arguments: ε (eps) –neighborhood : Those points within a radius of e from an unvisited point are retrieved. The resultant number of objects retrieved should be at least “MinPts” (minimum number of points ) to form a dense region, else the point is categorized as noise point. In this project, since the historical trajectories for same route are closely associated to each other, the value for ‘MinPts’ and ‘eps’ are identified based on an iterative approach which results in a reasonable number of clusters being formed.

### 6.3. ABSTRACTING TYPICAL TRAJECTORY BY CLUSTER ANALYSIS

In this section, we will explain the method in attaining the typical trajectory and its illustration. After applying DBSCAN clustering algorithm, trajectories with similar spatial and temporal points based are grouped into one cluster. A typical trajectory is defined as a set of clusters containing historical trajectories which represents the similar pattern to our input flight plan. It must be prominent that, typical trajectory is now occupied only with those historical flight information correlated to input query and thus reduces the data size by large factor.

The below terminologies indicate that, the typical trajectory library holds ‘n’ trajectories, each trajectory Ti covers li points, and each point comprises x attributes.

***Typical Trajectory Library is represented as, T = {Ti , i=1,………,n)***

***In typical trajectory library, each trajectory record is represented as : Ti = {Pij, j=1,…….,li}***

***Each point : Pij = {Wjik, k=1,.....,x }***

Sample of an typical trajectory is shown in Table 3. Each record contains the attributes of flight number, latitude, longitude and time recorded for each point.

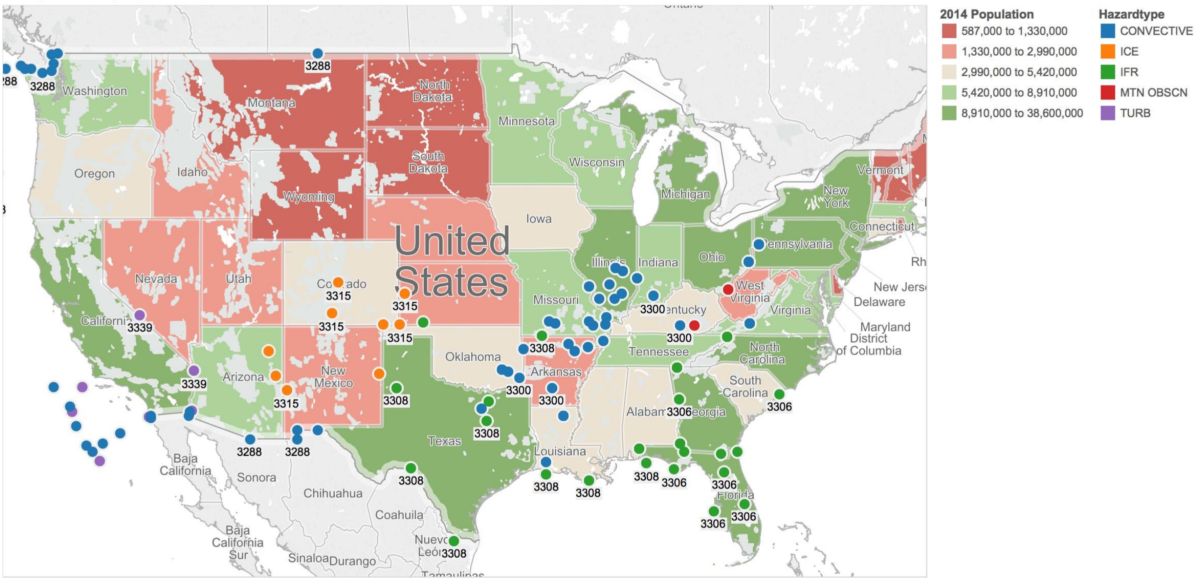
| Field | Example |
|---|---|
| **Flight Number** | AAR315 |
| **Time** | 2012-11-27 15:07:27 |
| **X Coordinate** | 36.066667484 |
| **Y Coordinate** | -115.150000123 |

Table 3 : Sample Typical Trajectory Point

### 6.4. HANDLING POLYGON SHAPED AIRSIGMETS

Weather data in the context of flights and aviation is available as AirSigmets in our data set. The problem with AirSigmets is that it contains predicted weather information such as hazard severity, hazard type for (lat,long) points that form a polygon area which for the purpose of analysis, we must first identify the flight trajectories that experienced or avoided these AIRSIGMETS from the historical data. This is not an easy task as flight trajectory information and weather information are from different sources. We use Tableau for this purpose to plot the AIRSIGMET information for any given time and day.

Airsigmets_Map_Plot_Time




Map based on Airsigmets across United States. Color shows details about Hazardtype. The numbers below are labeled by Airsigmetid. Map coloring shows populous areas. The data is filtered on Timevalidfrom and Timevalid for sigmets.

Figure 2 : Handling Airsigmet Polygons using Tableau

**Tableau** plots the weather information on a map, with different color for different severity such as icing and turbulence. This can be easily filtered based on time and day. For analysis, we need to consider trajectories between destinations that possibly encountered the plotted AIRSIGMETS. From the visualization, it is clear that flights from or to airports in the state of Illinois, Missouri, Arkansas, Oklahoma, Louisiana, Texas and Florida have a high probability of encountering troublesome weather.

### 6.5. INTENT BASED MODEL

In this project, we predict the flight path based on "Intent Inference Algorithm".An intent inference algorithm, implemented in [10], is to *validate the stated flight plan and infer the intended aircraft's intent.* Hence, the intent inference algorithm is applied in this project to better represent the intended flight route. As in figure 3. the intent based model block uses the airsigmet weather information and typical trajectory data to get an inferred intent and predict the flight trajectory dynamically.

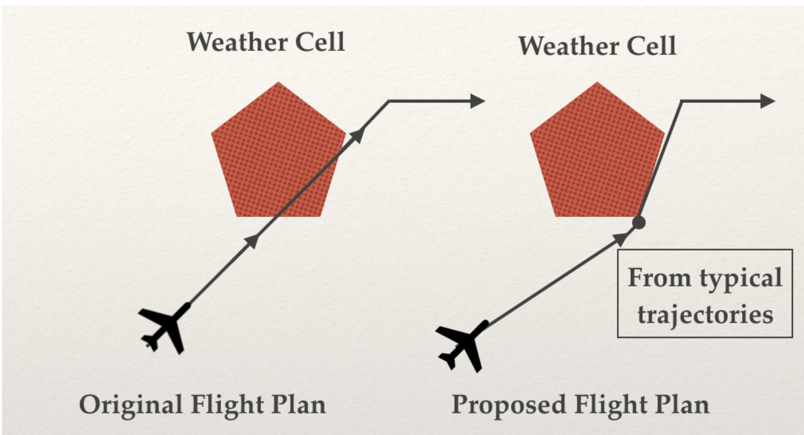



Figure 3 : Sample showing detecting and avoiding weather cell

As per semi lazy paradigm, the input query containing the initial flight plan and sigmet information is fed into the intent-based model. At any time, each point in the trajectory is subjected to comparison against the AIRSIGMET information provided. Those points in input flight plan that intend to cross a hazardous polygon-shaped AIRSIGMET area, is suggested/recommended with an alternate trajectory point. This alternate trajectory point is derived from the typical trajectory data that had evaded a similar AIRSIGMET area. The process is repeated until the flight reaches its destination.

# 7. EXPERIMENTAL SETUP

| | |
|---|---|
| Configurations | OS X El Capitan, 2.6 GHz Intel Core i5, 8 GB |
| Programming Language | Python |
| Database | MongoDB (DB Client - Robomongo) |
| Packages | Pandas, pymongo, numpy, matplotlib, networkx |

Table 4 : Experimental Setup Details

# 8. RESULTS

In this section, we present the results of trajectory predictions based on the semi-lazy paradigm on the real time US Domestic flight data for the following two queries.

**Query 1:** Trajectory Prediction Between Dallas/Fort Worth airport(DFW) - Chicago O'Hare International Airport (ORD)
**Query 2:** Trajectory Prediction Between San Francisco(SFO) - Los Angeles(LAX).

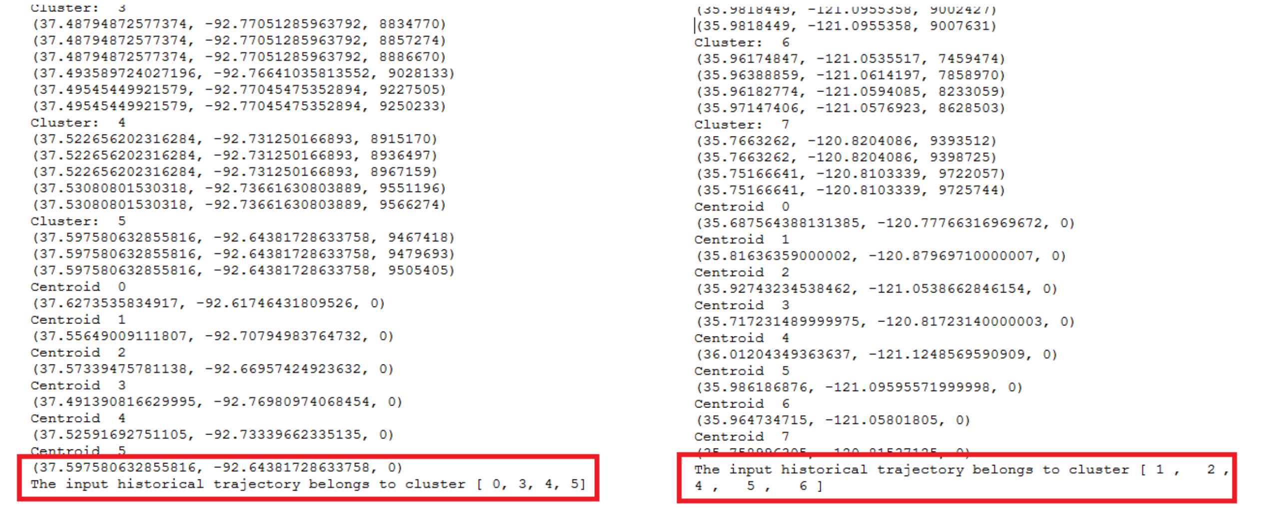
The results are presented for each phase of the model followed by the performance benchmark of the query, for better understanding and to appreciate the efficiency of semi-lazy paradigm.

## 8.1. CLUSTER ANALYSIS :

The data set pertaining to the source and destination airport is subjected to clustering and typical trajectories are obtained. Table 5 highlights the results of the clustering for Query 1 and Query 2

| Features | Query 1 | Query 2 |
|---|---|---|
| **Total Trajectory Points** | 22 Million | |
| **Total Clusters Formed** | 6 | 8 |
| **No of clusters, given input flight plan belongs to** | 4 | 5 |
| **Typical Trajectories Obtained** | 65 - Flight IDs from 4 clusters; 7157 trajectory points | 273 Flight IDs from 4 clusters 30030 trajectory points |
| **% of relevant data from the whole dataset** | **0.04%** | **0.13%** |

Table 5 : Cluster Analysis Results



```
Cluster:  3
(37.48794872577374, -92.77051285963792, 8834770)
(37.48794872577374, -92.77051285963792, 8857274)
(37.48794872577374, -92.77051285963792, 8886670)
(37.493589724027196, -92.76641035813552, 9028133)
(37.49545449921579, -92.77045475352894, 9227505)
(37.49545449921579, -92.77045475352894, 9250233)
Cluster:  4
(37.522656202316284, -92.731250166893, 8915170)
(37.522656202316284, -92.731250166893, 8936497)
(37.522656202316284, -92.731250166893, 8967159)
(37.53080801530318, -92.73661630803889, 9551196)
(37.53080801530318, -92.73661630803889, 9566274)
Cluster:  5
(37.597580632855816, -92.64381728633758, 9467418)
(37.597580632855816, -92.64381728633758, 9479693)
(37.597580632855816, -92.64381728633758, 9505405)
Centroid  0
(37.6273535834917, -92.61746431809526, 0)
Centroid  1
(37.55649009111807, -92.70794983764732, 0)
Centroid  2
(37.57339475781138, -92.66957424923632, 0)
Centroid  3
(37.491390816629995, -92.76980974068454, 0)
Centroid  4
(37.52591692751105, -92.73339662335135, 0)
Centroid  5
(37.597580632855816, -92.64381728633758, 0)
The input historical trajectory belongs to cluster [ 0, 3, 4, 5]

(35.9818449, -121.0955358, 9002427)
(35.9818449, -121.0955358, 9007631)
Cluster:  6
(35.96174847, -121.0535517, 7459474)
(35.96388859, -121.0614197, 7858970)
(35.96182774, -121.0594085, 8233059)
(35.97147406, -121.0576923, 8628503)
Cluster:  7
(35.7663262, -120.8204086, 9393512)
(35.7663262, -120.8204086, 9398725)
(35.75166641, -120.8103339, 9722057)
(35.75166641, -120.8103339, 9725744)
Centroid  0
(35.687564388131385, -120.77766316969672, 0)
Centroid  1
(35.81636359000002, -120.87969710000007, 0)
Centroid  2
(35.92743234538462, -121.0538662846154, 0)
Centroid  3
(35.717231489999975, -120.81723140000003, 0)
Centroid  4
(36.01204349363637, -121.1248569590909, 0)
Centroid  5
(35.986186876, -121.09595571999998, 0)
Centroid  6
(35.964734715, -121.05801805, 0)
Centroid  7
The input historical trajectory belongs to cluster [ 1 ,   2 ,
4 ,   5 ,   6 ]
```

Figure 4 : Screenshot depicting clusters formed for Query1 (DFW - ORD)

The percent of trajectory points between the DFW and ORD among the complete dataset constitutes around 0.04% , whereas for SFO-LAX, it is 0.13%. Thus, by the clustering we are able to retrieve the relevant data i.e.,0.04% data and 0.13% for DFW-ORD and SFO-LAX respectively required for modelling. The result of clustering is shown in figure 4

### 8.2. INTENT BASED MODEL :

The typical trajectories obtained from clustering are used to predict the flight trajectories for the given input flight plan using the proposed intent based model. The output of the model can be better explained using Figure 5. We see that the green line, representing the predicted trajectory against the given input flight plan path(red line) has deviated/ alternate route at the bubbled areas. This is because of the airsigmet forcing the pilot to take a detour to avoid hazardous weather.

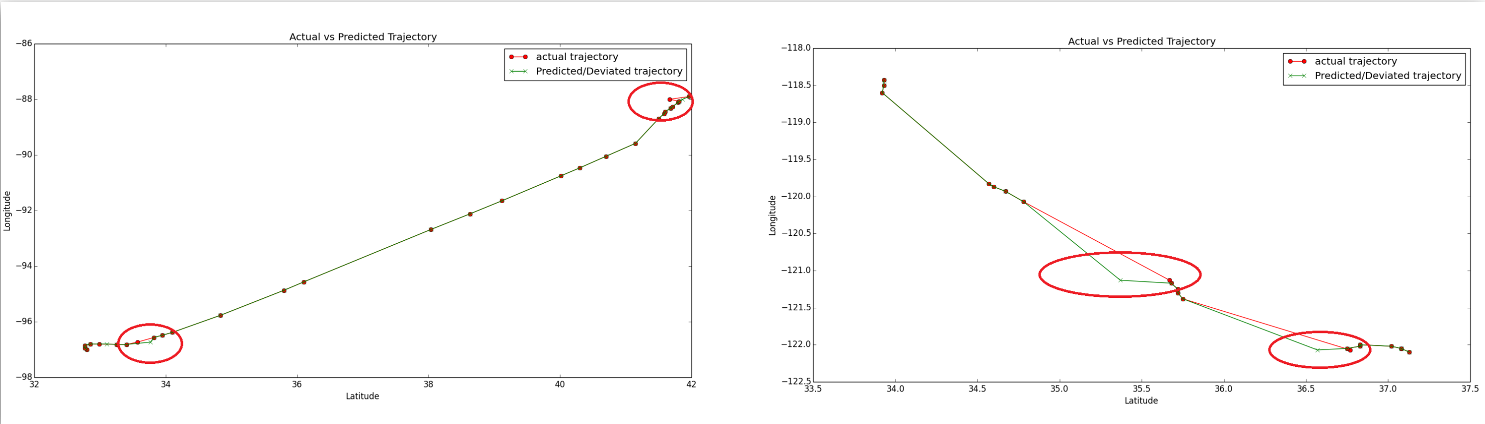



Figure 5: Deviation in trajectories against original trajectory for a) DFW-ORD b) SFO-LAX

| Input Flight Plan | | Predicted Flight Plan | |
|---|---|---|---|
| Latitude | Longitude | Latitude | Longitude |
| 32.79999924 | -97 | 32.79999924 | -97 |
| 32.76666641 | -96.94999695 | 32.76666641 | -96.94999695 |
| 32.76666641 | -96.84999847 | 32.76666641 | -96.84999847 |

| Input Flight Plan | | Predicted Flight Plan | |
|---|---|---|---|
| **Latitude** | **Longitude** | **Latitude** | **Longitude** |
| 32.84999847 | -96.80000305 | 32.84999847 | -96.80000305 |
| 32.98333359 | -96.80000305 | 32.98333359 | -96.80000305 |
| 33.25 | -96.81666565 | 33.25 | -96.81666565 |
| 33.4000015259 | -96.8166656494 | 33.4000015258789 | -96.8166656494141 |
| 33.5666656494 | -96.7333297729 | 33.8166656494141 | -96.5666656494141 |
| 33.9500007629 | -96.4833297729 | 34.0999984741211 | -96.3833312988281 |
| 34.8333320618 | -95.7666702271 | 34.9833335876465 | -95.6500015258789 |
| 35.7999992371 | -94.8666687012 | 35.6500015258789 | -95.0 |
| 36.0999984741 | -94.5666656494 | 36.533332824707 | -94.1333312988281 |
| 38.03333282 | -92.68333435 | 38.03333282 | -92.68333435 |
| 38.63333511 | -92.1166687 | 38.63333511 | -92.1166687 |

Table 6: Original trajectory vs Predicted flight trajectory points

### 8.3. PERFORMANCE BENCHMARK :

The time taken to predict the trajectory is measured for both the semi-lazy and eager model, and tabulated as shown in Table 7. We can clearly see that semi-lazy outperforms the eager learning model.

<table>
<tr><th>Benchmark</th><th>Semi-lazy Approach</th><th>Eager Learning</th></tr>
<tr><td>Querying all trajectories between Departure & Arrival</td><td>6m12.036s</td><td rowspan="4">Total time :<br>30 mins & Force-Quit</td></tr>
<tr><td>Sampling & identifying Typical trajectories :</td><td>4m0.494s</td></tr>
<tr><td>Trajectory Prediction :<br>(Intent Based Model)</td><td>1m20.165s</td></tr>
<tr><td>Total Time</td><td>11m32.695s</td></tr>
</table>

Table 7:Performance Benchmarking for Semi Lazy Approach vs Eager Learning

# 9. FUTURE EXTENSIONS

### Extended Weather Information

Given a framework for evaluating the current trajectory and predicting flight trajectories, if available further weather information can be factored in the model, such as PIREPS, TAFs and METARS. This will be helpful to extend the model for the entire flight trajectory including landings/takeoffs.

### Different Aircrafts

The model assumes all aircrafts are similar and considers trajectories with latitude, longitude, speed, altitude on a normalized level. Different aircrafts have various parameters distinguishing a typical flight trajectory for the aircraft. Another level of categorization while clustering could be to consider the historical trajectories of similar aircrafts.

### Conflict Detection

Weather is not the only factor that affects flight trajectories. The trajectories of other flights in the airspace also alters trajectories often as we have experienced. While extensive conflict detection mechanisms are already in place in the aviation industry, it would be largely beneficial to factor in such information in the prediction of trajectories.

### Airspace Health Management

An added measurement or exploration scope of the project is to measure the health of the airspace in which the trajectories actually take place. While this is a hugely difficult task, measuring the robustness of an airspace by removing various (lat,long) points in the trajectory gives an estimation of the current practices and policies in place.

Also, the model can be slightly modified to learn from the best practices. While flight plans are documented immediately when an altered flight journey takes place (this actually forms our historical trajectory data) further tuning could lead to the implementation of a fully supervised learning system.

# 10. CONCLUSION

In this project, using semi lazy paradigm aircraft's are likely to have improved flight path by avoiding hazardous weather area. We have illustrated data mining clustering algorithm to process the huge amount of historical radar data and to abstract typical trajectory .It is noted that the resultant typical trajectory is occupied only with those information correlated to input query and thus reduces the data size by large factor. We proposed an intent based algorithm that utilizes the dynamic airsigmet information and typical trajectory data as inferred intent and predict future trajectory point to avoid the weather cell. Through illustrative benchmarking ,our analysis shows that the proposed algorithm is computationally efficient and results in improved performance using semi lazy paradigm approach.

# REFERENCES

[1] http://www.garfors.com/2014/06/100000-flights-day.html
[2] http://www.planecrashinfo.com/cause.htm
[3] J. Kuchar and L. Yang, "A review of conflict detection and resolution modeling methods," IEEE Trans. Intell. Transp. Syst., vol. 1, no. 4, pp.178–189, Dec 2000.
[4] J. Yepes, I. Hwang, and M. Rotea, "An intent based trajectory prediction algorithm for air traffic control"
[5] Gariel M, Srivastava A N, Feron E, 2011, Trajectory Clustering and an Application to Airspace Monitoring, Intelligent Transportation Systems, IEEE Transactions on, pp. 1511-1524.
[6] Frank R, 2010, Clustering of Flight Tracks, AIAA Infotech @ Aerospace, American Institute of Aeronautics and Astronautics, Atlanta, pp. 20-22.
[7] Zengfu Wang, Quan Pan, Lin Lang, Yongmei CHENG, 2009, Dynamic Track Cluster Algorithm Based on Subtractive Clustering, Journal of System Simulation, pp. 5240-5246.
[8] Taobo Wang, Baojun Huang, 2013, Application of Improved Fuzzy Clustering Algorithm to Analysis of Trajectory, China Safety Science Journal, pp. 38-42.
[9] http://www.gequest.com/c/flight/data
[10] J. Yepes, I. Hwang, and M. Rotea, BAlgorithms for aircraft intent inference and trajectory prediction,[ AIAA J. Guidance, Contr. Dyn., vol. 30, no. 2, pp. 370–382, Mar./Apr. 2007