Aviation Data Mining

David A. Pagels
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA 56267
pagel093@morris.umn.edu

ABSTRACT

We explore different methods of data mining in the field of aviation and their effectiveness. The field of aviation is always searching for new ways to improve safety. However, due to the large amounts of aviation data collected daily, parsing through it all by hand would be impossible. Because of this, problems were found by investigating accidents. With the relatively new field of data mining we are able to parse through an otherwise unmanageable amount of data to find patterns and anomalies that indicate potential incidents before they happen. The data mining methods outlined in this paper include Multiple Kernel Learning algorithms, Hidden Markov Models, Hidden Semi-Markov Models, and Natural Language Processing.

Keywords

Aviation, Data Mining, Multiple Kernel Learning, Hidden Markov Model, Hidden Semi-Markov Model, Natural Language Processing

1. INTRODUCTION

On January 31st, 2000 a plane travelling from Puerto Vallarta, Mexico to Seattle, Washington dove from 18,000 feet into the Pacific Ocean, losing 89 lives. The cause of this accident was found to be "a loss of airplane pitch control resulting from the in-flight failure of the horizontal stabilizer trim system jackscrew assembly's acme nut threads. The thread failure was caused by excessive wear resulting from Alaska Airlines' insufficient lubrication of the jackscrew assembly"[2]. The cause of this accident was predictable through analysis of flight data recordings. There are many other incidents that would also be preventable through analysis of the flight data recordings.

Data mining is a broad field of data science that was developed to make predictions on future data based on patterns found in collected data. Finding patterns in aviation data manually is impracticable due to the mass amount of data produced every day. Data mining has been able to start addressing this problem. Although they are not yet optimized for mining of aviation data in their current state, some common data mining methods, such as kernel methods, text classification, and Hidden Semi-Markov Models, are being explored. Kernel techniques have been largely developed around either discrete or continuous data. This limitation makes it unsuited for use on the combined discrete and continuous data collected in aviation. Hidden Markov Models are limited to analyzing sequences without the ability to take into account the duration of actions. Aviation incident reports often contain a small amount of information per report, while current methods of text classification requires large amounts of descriptive data. Although these approaches are not optimal for aviation data, we can use these concepts to produce new approaches for data mining this data.

In section 2 of this paper we will be explaining the concepts necessary to understand the approaches to aviation data mining outlined in sections 3 and 4. In section 3, three different methods of data mining in aviation will be introduced, and the methods explained. The first of these three methods is data mining using Multiple Kernel Learning, which finds patterns in combined discrete and continuous data. The second of the three methods compares the effectiveness of the Hidden Markov Model versus the Hidden Semi-Markov Model in detecting anomalies. The third method analyzes the effectiveness of a text classification algorithm. Section 4 will discuss the relative success found in the results of these methods, and section 5 will summarize the effectiveness of the methods.

2. BACKGROUND

To discuss several methods of data mining used in aviation today, we need to understand several data mining concepts. These concepts include supervised, semi-supervised, and unsupervised learning; text classification; Natural Language Processing (NLP); Support Vector Machines (SVMs); Hidden Markov Models (HMMs); Hidden Semi-Markov Models (HSMMs); and kernels. To summarize, we classify the data by searching for key words in text (NLP), by finding clusters of data (kernels), or by observing the probability of a sequence of events (HMMs and HSMMs).

2.1 Aviation Data

We show implementation of these methods on three types of aviation data in this paper. The first is Flight Recording Data collected by the flight data recorder. The flight data recorder is informally known as the black box. Planes

UMM CSci Senior Seminar Conference, December 2014 Morris, MN.

equipped with flight recording data typically record up to 500 variables of data per second for the duration the plane is being operated [2]. Some of the variables described in these flight data recordings are time, altitude, vertical acceleration, and heading [1]. Some of these variables are discrete and some are continuous.

The second type of data is synthetic data. This is data generated with flight anomalies intentionally placed in the data to test the abilities of the algorithms to recognize the anomaly. These anomalies are referred to as dispersed anomalies. These anomalies may be an unconventional sequence of events, an unusual duration between events, etc. Some of the synthetic data used in this paper is data generated from a robust flight simulator, FlightGear. The FlightGear simulator is often used in the aviation industry and in academia due to its accuracy [6].

The third type of data is aviation incident reports. These reports do not have any strict conventions, do not require pilots to use specific terms, and include narratives. Since this data is not uniform, we must find a method to determine the relevant and important data.

2.2 Labels and Labeled Data

A label is a descriptive word assigned to data based on some property of the data. The labels in this paper are called shaping factors, or **shapers**, of an aviation incident. Examples of shapers in an aviation incident might include illness, hazardous environment, a distracted pilot, etc.

2.3 Supervised, Semi-Supervised, and Unsupervised Learning

There are many methods of finding a function to describe data. This function is commonly called the model, as it is made to model some set of data. Three such methods include supervised, semi-supervised, and unsupervised learning. Supervised learning uses labeled data to form the function. Semi-supervised learning uses some labeled data along with some unlabeled data to form the function. Unsupervised learning uses no labeled data to form the function. The term supervised in this context means that the labels for the data have already been found and are being used to construct the new model in a somewhat predictable way. The set of data used in supervised and semi-supervised learning is called the training set.

2.4 Natural Language Processing

Natural Language Processing (NLP) is a field of computer science focused on gathering meaningful data from text generated by humans. Aviation incident reports are not uniform, as they are filled out by humans. To get meaningful data from these reports, we first have to identify the overall picture of the data. This process is called text classification. Text classification is a general term and there are several different methods of text classification. The research outlined in this paper classifies text by using some prelabeled incident reports. Using the reports and the shapers associated with these reports, we can then find words in the reports that are commonly associated with a shaper. These words are referred to as **expanders**. While these expanders are being found, we can label unlabeled reports that are likely to be associated with a shaper if it contains a minimum number of expanders.

2.5 Kernels and Support Vector Machines

A Support Vector Machine (SVM) classifies new data into one of two categories. This data is represented by vectors which are denoted by an arrow over a variable. It does this by separating the data with a hyperplane. A hyperplane is a line/plane of regression that best separates the two categories of data. This hyperplane is constructed by the SVM. For example, the hyperplane in Figure 1 is the line separating the two clusters. Sometimes, an SVM is unable to produce a hyperplane. When this is the case, a kernel trick is used. A kernel trick maps the plane into a higher dimension so that a hyperplane may be found by the SVM [Figure 2]. The hyperplane in the left image of Figure 3 is the plane separating the two clusters, it is then shown mapped back into two dimensions in the right image of Figure 3. These clusters are considered labeled after the hyperplane is constructed. The label is determined by the location of the data point relative to the hyperplane. A kernel is a function used to find the similarity between unlabeled data and the data points, and label it accordingly.

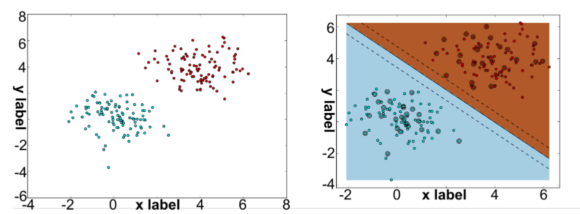

Figure 1: Linearly separable data [4].

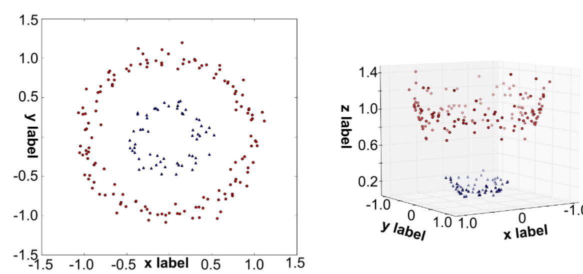

Figure 2: In the case of non-linearly separable data, we can use a kernel trick to map the data to a higher dimension [4].

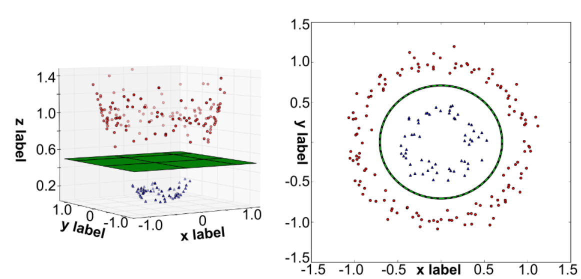

Figure 3: Once the data is mapped to a higher dimension, we find a hyperplane to separate it [4].

2.6 Hidden Markov Models (HMM) and Hidden Semi-Markov Models (HSMM)

To understand Hidden Markov Models we must first understand Markov chains. A Markov chain is a sequence of states such that the probability of a transition from one state to the next is dependent on the current state. A Markov chain in aviation would be the probability of going from one maneuver to the next (e.g. the pilot is more likely to stop the aircraft after touching down than stopping the aircraft after taking off.)

An HMM is a hidden Markov chain (the x level in Figure 4) with observable states (the o level in Figure 4) that allow us to infer the most probable state of the Markov chain. In the instance of aviation, the Markov chain is hidden as we are measuring the maneuvers of the aircraft, x_t, based on the inputs of the pilot, o_t. For instance, the pilot turning the yoke to the left, pulling the yoke back slightly, and applying left rudder would be the observable state, o_t, and the hidden state would be the aircraft turning to the left, x_t.

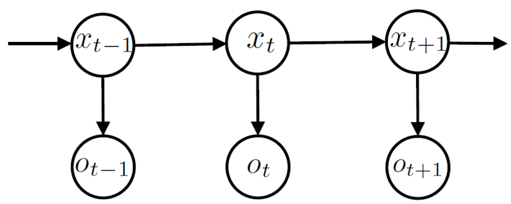

Figure 4: An example of a Hidden Markov Model with state transition probabilities ommitted [6].

An HSMM is an HMM that accounts for the change of transition probabilities between states over the duration of a state. This is necessary since the duration between actions could possibly classify the pattern as anomalous. An abnormally long time between a plane touching down and the plane stopping is an example of a sequence that is anomalous due to duration.

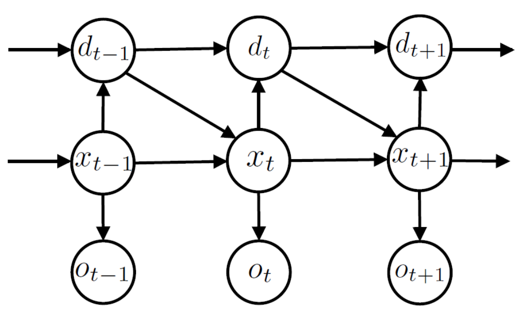

Figure 5: An example of a Hidden Semi-Markov Model with state transition probabilities ommitted [6].

3. METHODS

Next, we will explore different methods of data mining being used in the aviation industry today. Two of the three methods search for anomalies in flight recording data. Anomalous data in this context is defined as "unusual order of actions, unusual duration between actions, forgotten actions, etc"[6].

3.1 Data Mining Flight Recording Data Using Multiple Kernel Learning

As discussed earlier in the background section 2.1, the data to be analyzed can be a mixture of discrete and continuous data. To find anomalies in such mixed data, we must introduce Multiple Kernel Learning (MKL) methods. Using multiple kernel learning allows us to analyze discrete and continuous data simultaneously.

The kernels for discrete and continuous data are combined using the function:

$$k(\vec{x}_i, \vec{x}_j) = \eta K_d(\vec{x}_i, \vec{x}_j) + (1 - \eta) K_c(\vec{x}_i, \vec{x}_j)$$

In this function K_d is the kernel over discrete sequences, and K_c is the kernel over continuous sequences. The constant η is the weight of the kernel which was 0.5 for this research [2]. We use the following function to determine the discrete kernel:

$$K_d(\vec{x}_i, \vec{x}_j) = \frac{|LCS(\vec{x}_i, \vec{x}_j)|}{\sqrt{l_{\vec{x}_i} l_{\vec{x}_j}}}$$

The value of K_d is equal to the length of the Longest Common Subsequence (LCS) of two sequences divided by the square of the product of the two sequences length (l_x). Consider an example of an LCS made from nonsensical data:

$$ABB\,CBB\,AC$$

$$AB\,A\,BA\,A\,C\,B$$

$$ABBAC$$

The LCS is found by using the Hunt-Szymanski algorithm, which is explained in greater detail in [3].

We use the same function to find the continuous kernel, but we first preprocess the continuous data to make it discrete. To do this, a variant of the Sample Aggregate approXimation (SAX) representation was used on the data. To get the SAX representation, we find the averages of set intervals along the time series. An example of this is shown in Figure 6.

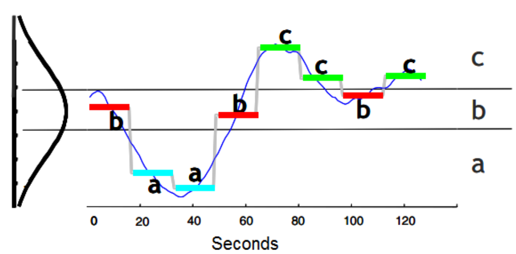

Figure 6: The SAX representation of data. This continuous time series was translated to the sequence baabccbc [5].

To be able to test the effectiveness of the MKL approach, the Orca and SequenceMiner algorithms were used as a baseline. The Orca algorithm specializes mainly in the finding of patterns in continuous data. SequenceMiner finds patterns in discrete data [2]. To initially test the Multiple Kernel Anomaly Detection (MKAD) algorithm, data was generated with 12 pre-determined anomalies. Of these 12 pre-determined anomalies, 3 were continuous, and 9 were discrete. On the generated data, Orca was unable to detect any discrete faults, but did detect every continuous fault; SequenceMiner was able to detect 89% of discrete faults, but no continuous faults; and the MKL method was able to detect all of both types of faults.

The real world data analyzed using the Multiple Kernel Anomaly Detection method was a set of 3500 flights consisting of 160 parameters, sampled once per second for an average flight duration of 1.7 hours [2].

3.2 Data Mining Flight Recorder Data using HMM and HSMM

Hidden Markov Models and Hidden Semi-Markov Models are analyzed in this paper to gauge their effectiveness in finding anomalous patterns in flight recording data. As discussed earlier, HMMs have a distinct disadvantage versus HSMMs, as HSMMs have the ability to account for duration of states, whereas HMMs do not.

The two methods use a dataset of "110 landings under regular operating conditions" from a flight simulator to define normal operation. This data came from a flight simulator called FlightGear, which is introduced in Section 2.1 [6]. For these simulations, there were 12 discrete pilot commands being recorded. Five different types of anomalous landings were then created using FlightGear:

1. Throttle is kept constant and flaps are not put down. The rest of flight is the same as in normal case.

2. No initial throttle increase, the rest of operation is normal.

3. The flight is similar to normal, except that the flaps are not put down.

4. At the end of the flight the brakes are not applied, the rest of operation is normal.

5. Pilot overshoots the airport runway and lands somewhere behind it.

Each of those scenarios were repeated 10 times for 50 anomalous scenarios.

The log of the probability of a sequence divided by the length of the sequence was then found to determine the likelihood of a the sequence. If a sequence of states were found to be anomalous, the probability of each state, given the sequence of states before it, was used to find the anomalous state [6].

A simple set of synthetic data was used to check that the HSMM was able to detect anomalous state durations and HMM was not. This data set had 25 sequences with normal duration between states, and 25 of the same sequences, but with abnormal duration between states. The ability of HSMM to detect anomalous state durations can be seen in Figure 7.

To interpret a Receiving Operating Characteristic (ROC) curve, one must know that as the line is followed from (0, 0) to (1, 1), the threshold is being changed for how the data is classified. The knotted line depicts the ability of HSMM to detect anomalous state durations. Since the area under the knotted line approaches the coordinate (0, 1), we can see that HSMM has a threshold value that will produce minimal false positives and catch most true positives. However, the solid line shows that HMM is fairly unreliable at any threshold level.

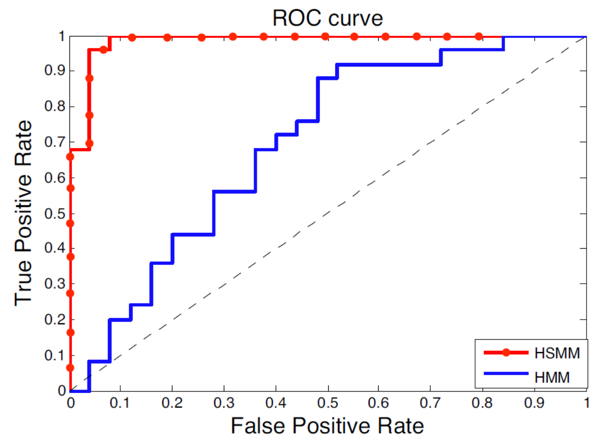

Figure 7: Detection of anomalous state duration of HMM and HSMM [6].

3.3 Data Mining Aviation Incident Reports

Conventional methods of text classification are found ineffective when used on aviation incident reports [8]. This is due to the limited amount of information given on each incident report. An especially difficult task is the classification of minority classes due to the limited number of samples available for training. A minority class is a cause that accounts for less than 10% of the incidents. This paper explores the use of a semi-supervised bootstrapping algorithm produced to more effectively label causes in aviation incident reports with an emphasis on minority classes.

The aviation incident reports for this section came from the Aviation Safety Reporting System (ASRS) database [8]. There are 14 potential labels (or causes of the incident) for the data. These labels are called shapers. Examples of these shapers are familiarity, physical environment, physical factors, preoccupation, and psychological pressure [8]. To assign shapers to incident reports, algorithms commonly search for words, referred to as expanders, which are indicative of certain labels.

The data taken from the ASRS database includes descriptions and opinions of the persons filing the incident report. These descriptions and opinions are written out in abbreviated words and often contain improper grammar. This data must be preprocessed to be able to be processed by the algorithms. To do this, the abbreviations are mapped to their expanded forms. For example, "HAD BEEN CLRED FOR APCH BY ZOA AND HAD BEEN HANDED OFF TO SANTA ROSA TWR" expanded to "had been cleared for approach by ZOA and had been handed off to santa rosa tower" [8].

Once the data is finished being preprocessed, it can be run through the baseline software and the bootstrapping algorithm. This bootstrapping algorithm is able to classify the text by adding key words (expanders) from pre-labeled data to a set. It then checks to see if the reports being labeled have more than 3 words from the set of key words. If the report being labeled has 3 or more words in common with that set, it is labeled with the shaper associated with the expanders. For example, if the bootstrapping algorithm

Table 1: Positive and Negative Expanders [8].

Shaping Factor	Positive Expanders	Negative Expanders
Familiarity	unfamiliar, layout, unfamiliarity, rely	
Physical Environment	cloud, snow, ice, wind	
Physical Factors	fatigue, tire, night, rest, hotel, awake, sleep, sick	declare, emergency, advisory, separation
Preoccupation	distract, preoccupied, awareness, situational, task, interrupt, focus, eye, configure, sleep	declare, ice, snow, crash, fire, rescue, anti, smoke
Pressure	bad, decision, extend, fuel, calculate, reserve, diversion, alternate	

is labeling reports with the Preoccupation label using the set of positive expanders in Table 1, and an unlabeled report contains the words "awareness", "task", "eye", and "smoke", it would label this report with the Preoccupation shaper regardless of any negative expander.

$Train(P, N, U, k)$
Inputs:
  P: positively labeled training examples of shaper x
  N: negatively labeled training examples of shaper x
  U: set of unlabeled narratives in corpus
  k: number of bootstrapping iterations
$PW \leftarrow \emptyset$
$NW \leftarrow \emptyset$
for $i = 0$ to $k-1$ **do**
  if $|P| > |N|$ **then**
    $[P, PW] \leftarrow ExpandTrainingSet(P, N, U, PW)$
  else
    $[N, NW] \leftarrow ExpandTrainingSet(N, P, U, NW)$
  end if
end for

$ExpandTrainingSet(A, B, U, W)$
Inputs:
  A, B, U: narrative sets
  W: unigram feature set
for $j = 1$ to 4 **do**
  $t \leftarrow \arg\max_{t \notin W} \left(\log(\frac{C(t,A)}{C(t,B)+1}) \right)$
  // $C(t, X)$: number of narratives in X containing t
  $W \leftarrow W \cup \{t\}$
end for
return $[A \cup S(W, U), W]$
// $S(W, U)$: narratives in U containing ≥ 3 words in W

Figure 8: The bootstrapping algorithm [8].

The bootstrapping algorithm [Figure 8] takes several inputs. These inputs consist of a set of positively labeled training examples of a shaper, a set of negatively labeled training examples of a shaper, a set of unlabeled narratives, and the number of bootstrapping iterations (k). Positive examples of a shaper are narratives which contain words that indicate that shaper, negative examples of a shaper are narratives that include words that indicate that the shaper is not appropriate for said narratives. To find more expanders, two empty data sets are initialized, one for positive expanders, one for negative expanders. The algorithm then iterates the ExpandTrainingSet function k times. In these iterations, if the size of the set of positively labeled training examples is larger than the size of the set of negatively labeled training examples, a new positive expander is found and vice versa.

In the second function of the algorithm, ExpandTrainingSet, 4 new expanders are found. The inputs for this function are the narrative sets of the positive and negative shaper examples in their respective variable (A or B dependent on the size of the narrative sets), the set of narratives not assigned a shaper, and a unigram feature set: a set of positive or negative expanders (dependent on the sizes of the positive and negative narrative sets). Expanders are found by finding the log of the number of narratives in one set (P or N) containing a word, t, divided by the number of narratives in the other set containing the word, t, for every word. The maximum value of these values is then found. If expanding the positively labeled training examples, the positive narrative set is used in the numerator, and the negative narrative set is used in the denominator, and vice versa for expanding the negatively labeled training examples. If this word is not already an expander, it is added to the set of expanders. After the 4 new expanders are found, all of the unlabeled narratives that contain more than 3 words in the list of relevant expanders are added to the relevant list of labeled training examples.

To use the semi-supervised bootstrapping algorithm, some data must already be labeled. To initially label a test set of incident reports, two graduate students not affiliated with the research labeled 1,333 randomly selected reports from the ASRS database.

The reports are then classified by a pre-existing software package called LIBSVM to have a baseline to which we may compare the bootstrapping algorithm.

4. RESULTS

In this section we discuss the outcomes of the three methods compared to their respective baselines.

4.1 Multiple Kernel Learning

The data of concern in the set of 3500 flights were the points below 10,000 feet mean sea level (MSL). The flight data of these flights was passed through an algorithm to rid the set of flights where the sensors or sensor values were unreasonable, likely due to noise or other malfunctions. This left 2500 of the original 3500 flights for analysis. To find a training set for the algorithm from these 2500 flights, "an aggressive data quality filter was applied to the remaining flights", which returned "approximately 500 flights" [2]. Of the 2500 flights, 227 flights were found to be anomalous by the MKAD method. Of these 227 anomalous flights, 19 were discrete, 94 were continuous, and 114 were heterogeneous (discrete and continuous). Table 2 shows the results from the multiple kernel method research, the overlap between this multiple kernel method and the baselines. *Multiple kernel learning for heterogeneous anomaly detection* suggest the

MKAD approach was able to detect anomalies indicated by both discrete and continuous data more effectively than the baseline methods based on this overlap.

Algorithms	Overlap of anomalous flights (with MKAD)		
	Discrete	Continuous	Heterogeneous
O	21%	59%	34%
S	53%	0%	54%
O & S	58%	59%	67%
MKAD	**19**	**94**	**114**

Table 2: Overlap between MKAD approach and baselines. The baselines are represented by O for Orca and S for SequenceMiner. The values of O & S are the union of their anomalous sets [2].

4.2 HMMs and HSMMs

Overall, of the scenarios listed in section 3.2, HSMM performed better on scenarios 1 and 2, and performed similarly to HMM on scenarios 3, 4, and 5. While the authors of the paper discussed possible methods to further improve anomaly detection using a HSMM, the result confirms the relevance and importance of an algorithm that takes duration of states into account.

4.3 Semi-Supervised Bootstrapping Algorithm

A sample of the words indicative of certain labels, or expanders, found when the bootstrapping algorithm was run on the set of incident reports may be found in Table 1. We can get an idea of the effectiveness of the bootstrapping algorithm based on the expanders. In a table from *Semi-Supervised Cause Identification from Aviation Safety Reports*, 1.8% of the reports were annotated with the 'Pressure' shaper. Even with the small percentage of the data set having a cause of pressure it is easy to see how the positive expanders shown in the table can indicate pressure as a cause leading to the incident.

The bootstrapping algorithm's effectiveness was measured by F-measure. An F-measure is the combination of precision and recall. Precision is the fraction of reports accurately assigned a shaper, recall is the fraction of the reports for a shaper that were properly labeled. The bootstrapping Algorithms' F-measure yielded "a relative error reduction of 6.3% in F-measure over a purely supervised baseline when applied to the minority classes" [8].

5. CONCLUSION

Techniques in data mining show signs of improving the ability to detect anomalies in aviation data. We are now able to detect heterogeneous anomalies in data, where before we were only able to find either discrete or continuous anomalies. To do this we use Multiple Kernel Learning. We have learned that a Hidden Semi-Markov Model approach to detecting anomalies is favorable over a Hidden Markov Model approach. This is due to Hidden Semi-Markov Models having the ability to model the probability of sequences with the duration of states having significance. Lastly, we have looked at a new text classification approach to effectively identifying causes in aviation incident reports with an emphasis on minority causes. To accomplish this, the bootstrapping algorithm was used to find causes based on key words contained in the aviation reports. Some continuing problems which have yet to be addressed in the field of data mining of aviation data include:

- Overly generalized data in incident reports, making cause identification a difficult task
- Providing a simple way for these methods to be deployed
- Linking reports between other data (e.g. linking incident report to aircraft maintenance records) [7].

6. ACKNOWLEDGEMENTS

Many thanks to Peter Dolan, Elena Machkasova, and Andrew Latterner for their invaluable feedback.

7. REFERENCES

[1] *14 Code of Federal Regulations 121.344*. 2011.

[2] S. Das, B. L. Matthews, A. N. Srivastava, and N. C. Oza. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 47–56. ACM, 2010.

[3] J. W. Hunt and T. G. Szymanski. A fast algorithm for computing longest common subsequences. In *Communications of the ACM: Volume 20-Number 5*, pages 350–353. ACM, 1997.

[4] E. Kim. Everything you wanted to know about the kernel trick (but were too afraid to ask). `http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html`, 2013.

[5] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*, 15(2):107–144, 2007.

[6] I. Melnyk, P. Yadav, M. Steinbach, J. Srivastava, V. Kumar, and A. Banerjee. Detection of precursors to aviation safety incidents due to human factors. In *Data Mining Workshops (ICDMW), 2013 IEEE 13th International Conference on*, pages 407–412. IEEE, 2013.

[7] Z. Nazeri, E. Bloedorn, and P. Ostwald. Experiences in mining aviation safety data. In *ACM SIGMOD Record*, volume 30, pages 562–566. ACM, 2001.

[8] I. Persing and V. Ng. Semi-supervised cause identification from aviation safety reports. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 843–851. Association for Computational Linguistics, 2009.