Recent Experiences with Data Mining in Aviation Safety
Earl Harris Jr., Eric Bloedorn, Neal J. Rothleder
The MITRE Corporation
{esharris, bloedorn, neal}@mitre.org

Abstract
While many data miners want to find “interesting” patterns, the definition of an interesting pattern is often unclear. This is a case study of data mining applied to the problem of improving aviation safety. To address the problem, we applied several methods to pertinent data; these methods achieved varying degrees of success. During the project, our notion of an interesting pattern evolved and our appreciation for a good working definition for interestingness grew.

Introduction
Data mining researchers have identified “interestingness” as an important problem [Piatesky-Shapiro et al, 1994; Liu, 1997]. Unfortunately, methods for finding interesting results are not fully understood. Machine learning algorithms can search across a large number of attributes and find interesting patterns. However, when we applied these algorithms, our group found that we needed a better understanding of the term “interesting.”

This paper presents an overview of our project. It shows how we initially applied both supervised concept learning and association rule algorithms [Agrawal et al, 1993] to data related to aircraft accidents. This paper critiques the use of these popular tools in the field of aviation safety. It also motivates our latest interest in applying an alternate technique.

The Data Mining Problem
We are involved in a joint, aviation safety project with the Center for Advanced Aviation Systems Development (CAASD) at The MITRE Corporation. Our goal is to determine the feasibility and effectiveness of building an integrated data mining workbench. To measure effectiveness, we have examined how well some workbench tools help us to characterize dangerous situations in the aviation world. We hope this examination will lead us to develop strategies for preventing accidents.

Investigators file an accident report for a flight when there is either 1) substantial or greater damage to an aircraft or 2) serious or greater injuries during the flight. Otherwise, when there is only minor aircraft damage, minor injuries, or just a threat to safety, examiners file an incident report. Given accident and incident reports involving aircraft, we mine for patterns that depict hazardous situations. The International Civil Aviation Organization (ICAO) is one source of such reports. Accident and incident reports are written for several types of flights, such as commercial, cargo, or pleasure. These reports contain over 50 attributes and provide information such as aircraft descriptions, pilot age, pilot experience, operators, time of day, weather, and the factors contributing to the accident (or incident). One of our tasks is to use collections of these reports to characterize those situations in which accidents occur.
Applying Traditional Machine Learning Techniques

Initially, we believed that a rule was interesting if it had the following properties:

- The rule covered many examples (many domain members satisfy the conditions of the rule).
- It was relatively accurate (correctly classifies most domain members that satisfy the conditions).

This was a simple working definition that we thought would meet our needs.

Our group applied C4.5 [Quinlan, 1993] to the problem of finding interesting classification rules for aviation safety. The training sample is from the ICAO database; it consists of 1,256 aviation accident and incident reports from 1994 to 1995. The objective was to discriminate between four classes of injuries: none, minor, serious, and fatal. These classes refer to the worst degree of injury for any passenger or crew member involved in the accident. Consider an arbitrary supervised concept learning problem with arbitrary attributes \( p_1, \ldots, p_n \), arbitrary values \( q_1, \ldots, q_n \), and arbitrary class label \( c \). The formula “\( p_1 = q_1 \) \& \ldots \& p_n = q_n \rightarrow c \)” means “if attribute \( p_1 \) is equal to the value \( q_1 \) and ... and attribute \( p_n \) is equal to the value \( q_n \) then the class of the report is \( c \).” Following an arbitrary rule is a number pair \( (N, E) \). \( N \) is the number of training examples covered by the preceding rule and \( E \) is the number of training errors. Some of the derived rules are shown below.

\[
\begin{align*}
event1 & = \text{INJURIES\_FROM\_TURBULENCE} \rightarrow \text{class SERIOUS} \ (22, 3) \\
event1 & = \text{COLLISION\_WITH\_HILL\_MOUNTAIN} \& \text{damage = DESTROYED} \rightarrow \text{class FATAL} \ (19, 0) \\
event1 & = \text{LOSS\_OF\_CONTROL} \& \text{damage = DESTROYED} \rightarrow \text{class FATAL} \ (34, 2) \\
\text{phase3} & = \text{EMERGENCY\_UNCONTROLLED\_DESCENT} \& \text{damage = DESTROYED} \rightarrow \text{class FATAL} \ (57, 5)
\end{align*}
\]

The resulting rules achieved 81.8% predictive accuracy on our test sample. Unfortunately, most of these rules were obvious. Clearly, a plane that collides into a hill or mountain and is completely destroyed is likely to have fatalities. If a plane loses control and is eventually destroyed or if a plane has an emergency uncontrolled descent and is eventually destroyed, fatalities are expected.

The most interesting rule to the layperson was the first rule. During a flight, if there is mention of some injuries from turbulence, a non-expert might expect the class to be “minor.” Our aviation expert did not consider this rule to be unexpected. In his past experiences, he has usually seen serious, but not fatal, injuries from turbulence.

Though the preceding rules may help us predict the severity of the injuries during an accident, they do not help us gain new insight into preventing accidents. The rules merely validate our aviation expert’s previous findings.

We also used a variant of DIC, an algorithm for generating association rules [Brin et al, 1997]. This tool was applied to reports found in the ICAO database. Unlike the structured reports in the previous example, these reports are represented as a set of words from a finite vocabulary. The vocabulary consists of factors associated with the events during an accident. For arbitrary words \( p, \ldots, q, r \), “\( \{p, \ldots, q\} \rightarrow r \)” means “if \( p \) is present
and ... and $q$ is present then $r$ is also present.” In the number triple following the rules $(LHS, A, C)$, LHS is the number of records where the left-hand side words $p$, ..., $q$ are all present. $A$ is the number of records where all the words $p$, ..., $q$, $r$ are present and $C$, the confidence, is $\sqrt[3]{A / LHS}$. Confidence is analogous to predictive accuracy; both are real numbers between 0 and 1 and high values indicate stronger associations between the left-hand side and the right-hand side.

Some of the derived association rules are easy to explain.

$$\{DISORIENTATION/VERTIGO\} \rightarrow AIRCRAFT\_CONTROL\ (31,\ 16,\ 0.52)$$
$$\{FUEL\} \rightarrow FORCED\_LANDING\ (101,\ 64,\ 0.63)$$
$$\{CYLINDER\} \rightarrow FORCED\_LANDING\ (45,\ 30,\ 0.67)$$
$$\{STALL\} \rightarrow AIRSPEED\ (61,\ 13,\ 0.57)$$

Any mention of disorientation and/or vertigo would most likely refer to the pilot. Since this would affect his control of the aircraft, aircraft control is also mentioned. Often, a mention of fuel would mean there is a starvation or exhaustion of fuel. This would lead to a forced landing. Any mention of cylinder would probably mean there is a serious problem with the engine, which leads to a forced landing. The term “stall” refers to inadequate speed to maintain lift; therefore, if a factor mentions stall, it often mentions airspeed.

The next two derived rules may be interesting to the non-expert.

$$\{SPIN\} \rightarrow AIRSPEED\ (18,\ 13,\ 0.72)$$
$$\{AIRCRAFT\_CONTROL, DISORIENTATION/VERTIGO\} \rightarrow CLOUD\_BASE/CEILING\ (16,\ 11,\ 0.69)$$

When a plane’s airspeed drops too low, the airplane loses lift. This causes a stall that can also lead to a spin. Though the result was unexpected to us, it was not surprising to the aviation expert.

After reviewing the records that the last rule covered, we came to the following conclusion. During bad weather, there is often a low cloud base. For pilots not qualified to fly in bad weather (not instrument rated), the reduced visibility often causes disorientation. This disorientation may lead to a loss of control. Again, this rule did not surprise our aviation expert.

In summation, applying data mining to aviation safety had a unique problem. Investigators had examined aircraft accidents so thoroughly, there were few unknown patterns that could be found in the publicly accessible databases. So, conventional data mining tools seem to only discover known trends.

These results also showed us a problem with our strategy. Clearly, our working definition of “interesting” was inadequate. We had seen rules that covered many examples and were relatively accurate (or had achieved high confidence). Unfortunately, these rules provided no new information to the domain experts. To the pure machine learning enthusiast, it is interesting for rules to cover many examples and achieve high predictive accuracy. But in our notion of “interesting,” it is necessary that the rules are unexpected or surprising to the expert.
Applying Alternative Techniques

After our experience in applying machine learning techniques, we adopted a different working definition for an interesting rule: a rule is interesting if it represents an exception to some commonly accepted rule. Though this working definition did not completely capture the notion of “surprising to the domain expert,” we liked it for the following reasons.

1. Since most of the rules that we found were obvious to the domain expert, we felt that characterizing cases that refute these rules might be surprising.
2. Exceptions to rules seemed easier for an automated tool to identify than rules that are unexpected to the expert.

To find exceptions to commonly accepted rules, our group is developing a system, called Smithers, that uses a technique called attribute focusing. Smither’s algorithm is loosely based on the underlying strategy of Advanced Scout [Bhandari et al, 1996]. For a particular attribute, A, of some structured, flat-file database, an attribute focusing algorithm calculates the overall distribution of the values of A in the database. Then it compares this distribution with the distribution of A in various subsets of the database. (Smithers uses an arbitrary discrete attribute and one of the attribute’s possible values to define a subset of the given database.) If a certain subset has a statistically different distribution of A, then the condition that defines the subset is marked as interesting. Note that the overall distribution is our baseline rule and the distributions for the subsets are the potential exceptions.

We applied Smithers to a version of the Aviation Safety Reporting System (ASRS) database. This database is a source of anonymous, voluntary incident reports related to aircraft safety. This database contains information that is similar to the information in the ICAO database. We extracted the ASRS reports that were categorized as runway incursions occurring between 1988 and 1997. Smithers focused on an attribute of the database that denotes the consequences of a runway incursion. This attribute has four possible outcomes. (For simplification, we focused on records whose consequence field contained a single outcome.)

1. Damage: there was either damage to the aircraft, an injury to a passenger, or emotional trauma to a passenger.
2. Reprimand: the incident triggered one of the following: a Federal Aviation Administration (FAA) threat of penalties, assigned FAA penalties, an FAA investigatory follow up, or a flight crew/Air Traffic Controller (ATC) review.
3. None: nothing happened.
4. Other: something happened, but it was not damage or a reprimand.

The following frequency table presents an interesting result from Smithers.

<table>
<thead>
<tr>
<th></th>
<th>Damage</th>
<th>Reprimand</th>
<th>Other</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall</td>
<td>45</td>
<td>1350</td>
<td>165</td>
<td>2840</td>
</tr>
<tr>
<td>Adv. Display (exp.)</td>
<td>6.48</td>
<td>194.52</td>
<td>23.78</td>
<td>409.22</td>
</tr>
</tbody>
</table>
The first row is the overall frequency of the consequences of runway incursions in the database. The second row shows the expected frequency of consequences for aircraft with advanced displays (such as a glass cockpit, which uses LED displays in place of analog dials, or heads up display (HUD)). This interpretation of expected frequency makes an assumption about the probability distribution of consequences for the aircraft with advanced displays. It assumes the distribution is the same as the probability distribution of the consequences for all the aircraft. The third row is the actual frequency for aircraft with advanced displays. To test statistical difference, Smithers applies an multinomial test of hypothesis, which calculates chi-square statistics. Given a 0.01 significance level, we can claim that the actual frequency is not a natural deviation from the expected frequency. There is a statistically significant difference between the preceding expected and actual distributions.

Comparing the actual frequencies to the expected frequencies, we found that the number of cases where there are damages are less than expected. The number of cases where there are reprimands is also lower. The number of cases where nothing happens is greater than expected.

One could infer from this result that the presence of an advanced display helps to reduce the damage and reprimand consequences of a runway incursion. Our aviation expert found the result intriguing and felt it warranted further study, which we are pursuing.

Smithers works well in aviation accident databases, where known trends are easy to find. In these databases, there is a good chance that the exceptions to the general trends are interesting. And Smithers is suited for discovering these exceptions.

Future Directions and Related Work

We plan to continue the development of Smithers. One obvious enhancement is the ability to analyze subsets of the database whose description includes more than one attribute. In addition, we plan to explore other methods of finding rules that are interesting to our expert. Many of these methods incorporate expert preferences during learning. The goal of our approach is to find a method that balances the predicted accuracy of an attribute (as often estimated by information gain ratio) with the preferences of the expert.

Liu, Hsu, and Chen also developed a definition for “interesting” [Liu et al., 1997]. They want rules that have “unexpectedness” (they surprise the expert) and “actionability” (the expert can use the rules to their advantage). To find interesting rules, they built a learning system post-processor that accepts codified expert knowledge. Given classification rules, the post-processor uses a sophisticated matching system to rank the rules with respect to how well each rule matches with the expert’s general impression of his domain.

Silberschatz and Tuzhilin proposed using a belief system to help determine unexpectedness [Silberschatz et al., 1996]. This system uses bayesian probability to model an expert’s confidence in his or her beliefs. In this model, the confidence in a belief could change when the tool encounters several counter examples.
We are also looking at methods that find rules that are “intelligible,” which is another important property. If a rule is so unexpected that it violates some strong beliefs of the domain experts, they may not want to use the rule. Pazzani, Mani, and Shankle developed strategies for learning classification rules that were “intelligible” to the domain expert [Pazzani et al, 1997]. To produce intelligible rules, Pazzani built machine learning tools that analyze codified expert knowledge. The learning tool used the knowledge to show favoritism toward finding rules that are consistent with the expert’s beliefs. Clark and Matwin wanted to induce rules that are both accurate and “explainable” with respect to some qualitative model [Clark et al, 1993]. To derive explainable rules, the learning tools can only consider rules that are consistent with the qualitative theory of the expert.

Conclusions

This paper summarizes our search for patterns that will help to improve aviation safety. Our application of C4.5 and the variant of DIC was successful when we measured their ability to find simple, accurate rules. Their application was deficient when the aviation expert measured the interestingness of the results. Our latest work uses Smithers, an attribute focusing tool, to find exceptions to strong patterns. Initial results with Smithers have been promising.

Data miners often need an effective working definition for interesting. If data miners have an effective working definition, it helps them to find the appropriate tool for their job.

REFERENCES


