

Exploring Data Mining Representations of Track Data

Shawn R. Wolfe
NASA

Copyright © 2009 SAE International

ABSTRACT

Data mining is often used to analyze data that is too voluminous or complex to analyze by hand. However, most data mining algorithms require a fixed-length vector representation, in contrast to track data, which is naturally multi-dimensional and variable in length. We explore several methods for converting flight track data to a representation appropriate for data mining, and evaluate the performance of these representations in both clustering and classification tasks. Our results show that relevant features are captured in our representations, and describe the tradeoff in representational choices.

INTRODUCTION

Recognizing key signatures in flight track data is an important step towards understanding pilot behavior; such a capability could be used for safety and security applications. Pilots have been studied in detail in terms of cognitive ability and flight deck procedures, but not extensively on the level of short-range navigation. Our initial interest in recognizing important actions from track data stems from research into pilot avoidance of convective weather; though the Federal Aviation Administration (FAA) often directs flights around poor weather, the pilots themselves also have the option to deviate from their assigned flight plan to avoid difficult weather conditions. From a traffic flow management perspective, it would be useful to anticipate which weather cells pilots are likely to avoid and which ones will they knowingly penetrate.

Our exploration of track data is influenced by this particular application but is not restricted to it. More generally, we ask the following questions:

1. Is it possible to identify a reasonable and meaningful set of maneuvers that pilots use (where such maneuvers are more complex than just banking, ascending and descending)?
2. If such maneuvers exist, what representation would support their identification through automated means?
3. Provided there are such maneuvers and we can identify them, are any of these maneuvers correlated to meaningful actions in an application of interest, such as the avoidance of convective weather?
4. Given positive answers to the previous questions, what algorithms would be effective at identifying these maneuvers?

The focus of this paper is primarily on the second question, which representations are appropriate. We evaluate four candidate representations in both a clustering and a classification context. We evaluate the results in terms of weather avoidance and recognition of holding patterns; results may differ for other tasks with very different flight track properties. We conclude with discussion of some of the difficulties in using flight track data for these particular tasks.

The Engineering Meetings Board has approved this paper for publication. It has successfully completed SAE's peer review process under the supervision of the session organizer. This process requires a minimum of three (3) reviews by industry experts.

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of SAE.

ISSN 0148-7191

Positions and opinions advanced in this paper are those of the author(s) and not necessarily those of SAE. The author is solely responsible for the content of the paper.

SAE Customer Service: Tel: 877-606-7323 (inside USA and Canada)
Tel: 724-776-4970 (outside USA)
Fax: 724-776-0790
Email: CustomerService@sae.org

SAE Web Address: <http://www.sae.org>

Printed in USA

RELATED WORK

Many of the techniques and representational strategies developed for machine vision could be applied to our task as well. Significant research has been performed using grid-based (i.e., pixel-based) techniques, matching the original digital image representation; our vector-based input data could be transformed into a grid-based format in order to use these techniques. However, more natural representations of shapes have been developed; an extensive list is given by Loncaric (Loncaric, 1998). Loncaric categorizes the approaches along two binary-valued axes: approaches that represent the shape in a single vector versus those that use higher dimensional representations; and approaches that only represent the boundary of the object versus those that represent internal structure. Of these, our approach is the most similar to the single vector methods that focus on shape boundary. Loncaric also requires that shape representation schemes be invariant to rotation, translation and scale, but this may not be the right choice for our representation; in particular, scale invariance is undesirable because pilots will not deviate to avoid a storm hundreds of miles away.

Chain encoding of lines, originally developed by Freeman (Freeman & Glass, 1969) and later extended by Koplowitz and Touissant (Koplowitz & Touissant, 1976), are more directly applicable to our flight track and path representation problem. Chain encoding approximates a line by translating points on a line to points on a grid through a distance function. The approximation of the line is created by connecting the points in sequence. As the flight track also has a temporal sequence, the notion of sequence is appealing; furthermore, the chain encoding is translation invariant but not scale invariant. However, the chain encoding is not rotation invariant, which we currently believe to be a desirable property. Also, though the point sequence is sufficient to capture the ordering of events, it is not sufficient to encode the actual time between events.

Another interesting approach could be to use gesture recognition to identify pilot actions (see (Mitra & Acharya, 2007) for a review). Gesture recognition has primarily been applied to recognizing human movements, but nonetheless can be applied to aircraft movement under human control. Like flight track data, gestures also have a temporal aspect. However, gesture recognition techniques make use of a notion of state, which is currently ill defined in our domain. Future research may lead to identifiable states that can be utilized in gesture recognition; indeed, a broader goal of our research would be to identify common pilot maneuvers that could be the states of a later gesture recognition system.

Two prior efforts have used clustering techniques to identify the impact of convective weather on aviation traffic. Song et al. (Song, Wanke, & Greenbaum, 2007) used a self-organizing map to identify common flows of traffic. In their representation, flight tracks were represented in a grid structure based on transitions to and from adjoining sectors. Once the flows were identified, the potential impact of weather on these traffic flows was estimated. Callaham et al. (Callaham et al., 2001) used an unspecified clustering algorithm to group days with similar weather conditions together. A gridding procedure was used to transform weather polygons into a matrix representation. The weather was also rolled-up into four-hour segments. Analysis of the clusters showed that intra-cluster days had comparable overall delay statistics.

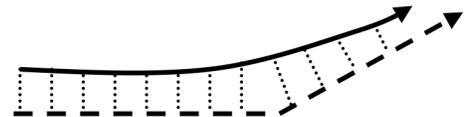


Figure 1. Correspondence between flight plan and track. For every point of the flight track (solid line), the closest point on the flight plan (dashed line) was identified as the corresponding point. In this figure, the corresponding points are connected with a dotted line. In most cases, this line was perpendicular to the flight plan.

APPROACH

We structured our data representation to be consistent with a previous study (Chan, Refai, & DeLaura, 2007), using the same dataset of approximately 90,000 instances and flight track subsetting procedure. In that study, flight tracks were divided into overlapping fifteen-minute segments, every five minutes. For example, a flight track from 10:00 am to 10:25 am would be separated into three overlapping fifteen-minute segments: 10:00 am to 10:15 am; 10:05 am to 10:20 am; and 10:10 am to 10:25 am. Since segments that were less than fifteen minutes in length were ignored, every flight segment had the same temporal extent. In the original study, these flight tracks were visually correlated with the flight plan.

In our study, such manual correlation was undesirable. Instead, a correspondence between the flight plan and the flight track was established. Our flight track data was divided into one-minute intervals. For every such flight track point, we used the closest point on the flight plan as the corresponding flight plan point (see Figure 1 for an example). This correspondence is a crucial concept in our representation, and the basis from which we derived all properties between the flight track and flight plan. We used three basic properties that were translated and combined in various ways in our representation:

Distance. For a given flight track point t_i and the corresponding flight plan point p_i , the distance $distance_{Spherical}(t_i, p_i)$ is the distance between the two points in spherical coordinates.

Heading differential. An instantaneous heading through the corresponding points to the next point on the chain is computed and their difference is calculated. Let $heading(x, y)$ be a function of two point arguments that gives the angle between the ray \overrightarrow{xy} and due north. For a given flight track point t_i and the corresponding flight plan point p_i , calculate the heading differential as

$$\begin{aligned} \Delta heading(p_i, t_i) \\ = heading(p_i, p_{i+1}) - heading(t_i, t_{i+1}) \end{aligned} \quad (1)$$

where t_{i+1} is the next flight track point in the sequence, and p_{i+1} is the corresponding flight plan point for t_{i+1} . Note that $\Delta heading$ is positive when the flight track veers to the right of flight plan and negative when the flight track veers to the left of the flight plan. $\Delta heading$ is also normalized such that $-180^\circ \leq \Delta heading \leq 180^\circ$. See Figure 2 for an example.

Progression on flight plan. For the progression on flight plan, a distance between successive flight plan points can be calculated. Let t_{i-1}, t_i be subsequent points on the flight track with p_{i-1}, p_i the corresponding flight plan points, respectively. We define the flight plan progression for point t_i as

$$progression(t_i) = distance_{Spherical}(p_{i-1}, p_i). \quad (2)$$

Furthermore, $progression(t_i)$ is defined as positive when p_{i-1} precedes p_i on the flight plan sequence, zero when p_{i-1} equals p_i , and negative when p_{i-1} occurs after p_i on the flight plan sequence.

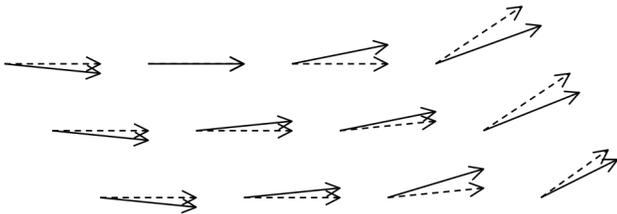


Figure 2. Translating correspondence to feature. The flight track (solid arrows) and flight plan (dashed arrows) from Figure 1 expressed as heading differentials. The arrows have been placed on three rows solely for increased readability. The heading differential is the angle between each pair of vectors.

We processed and combined these properties to create the following representations:

nonorm: In this representation, each feature is the absolute distance between the corresponding flight track and flight plan points.

norm: Like *nonorm*, this representation uses the distance between corresponding points as the only features. However, to prevent the overall distance from the flight plan from dominating, the minimum distance between the corresponding points of the instance was subtracted from all the distances. As a logical consequence, the value of at least one feature for every instance of *norm* was zero. (*norm* stands for normalized.)

heading: Every feature in this representation was the differential heading between the corresponding points, as defined above.

retro: This representation captured the progression along the flight plan by defining every feature to the progression on the flight plan for consecutive points as defined above. (*Retro* stands for retrograde, the opposite of progression.)

All our representations share common properties. First, they are rotation and translation invariant, but dependent on scale. This matched the intuition of the subject matter experts we consulted with for our evaluation tasks. Some differences with respect to orientation may be observed, as wind and weather follow certain patterns, but these differences are expected to be slight and therefore rotation dependence is not desirable. Likewise, though certain positions in the airspace will have different tendencies with respect to congestion, flight patterns (e.g., at an airport) and weather, the additional complication of factoring in position does not appear to be worthwhile at present. Scale, on the other hand, is important; we are focused on short-term maneuvers, so this temporal quality should be preserved. It would not make sense to regard a fifteen-minute maneuver as identical to a seventy-five minute maneuver for this reason.

Second, each representation translates the four-dimensional flight track and flight path information (three spatial dimensions, plus time) into a single one-dimensional vector. This is desirable because this sort of representation is suitable for many machine learning algorithms. Third, our translation is “lossy” in the sense that the original flight plan and flight track information cannot be reliably recovered from the single one-dimensional vector representation. Finally, different flight track/flight plan combinations will map unto the same single vector representation. This is a critical feature for the learning algorithms to be able to effectively generalize.

EVALUATION

We evaluated the suitability of these representations in both a clustering and classification context. Some of our metrics required a labeling of the data; we used two

different sets of labels on the same dataset (described below). The Weka software package (Witten & Frank, 2005) provided the implementation of all algorithms used.

ALGORITHMMS - For clustering, we used the k-means clustering algorithm, which iteratively refines clusters by putting instances into their closest centroid (MacQueen, 1967). The number of clusters k was arbitrarily chosen to be 100 in our case and purposely set to be not too low: initial efforts with $k = 20$ yielded clusters with too many seemingly dissimilar instances grouped together. Presumably, high-quality clusters that correctly capture an important commonality of cluster members can be identified when k is too large, but may not be visible when k is too low. On the other hand, evaluation becomes more difficult with larger k . We used the k-means clustering algorithm instead of other clustering algorithms as it is well-known, relatively fast and easy to understand. More complex algorithms may yield better clusters, but presumably a simple algorithm such as k-means would be sufficient to evaluate the usefulness of the representations. It is worthwhile to note that this implementation automatically standardizes the data (i.e., transforms the attributes to have zero mean and unit variance), thus transforming it before invoking the clustering algorithm.

We used metrics with and without class labels to evaluate quality of the clusters. We compared our results with the same metric applied to a random clustering of the data. To create the random clustering, we created clusters of approximately the same size as were generated by k-means, but with randomly selected cluster members. This was repeated ten times for each original set of clusters to account for the variability caused by the random selection. Comparison with random clusters avoids issues with metrics that are affected by the number of clusters or are difficult to interpret independently.

For the classification tasks, we utilized several simple algorithms: Decision tables (Kohavi, 1995), which use build simple majority-class rules on a subset of attributes; decision trees (specifically C4.5 (Quinlan, 1993)), which splits the data into increasingly smaller groups by attribute until a classification can be inferred; nearest neighbors (Aha & Kibler, 1991), which classifies based on classifications in the local neighborhood (in our case, only the closest single neighbor); and a naïve Bayes classifier (John & Langley, 1995), which makes a probabilistic class prediction under the assumption of probabilistic independence.

METRICS - Recall that k-means attempts to minimize the distance from the cluster centroids to each cluster member. The sum of these distances for a particular cluster can be thought of as the sum of error for that

particular cluster; the total sum of these errors would be the sum of these sums, namely

$$TSSE = \sum_{i=1}^k \sum_{x \in C_i} \text{distance}_{kmeans}(x, c_i) \quad (3)$$

where c_i is the centroid of the i^{th} cluster, x is an element of the i^{th} cluster C_i , and distance_{kmeans} is the distance measure used by k-means. In our case, this distance measure is simply Euclidian distance (as none of our features have retained the spherical coordinates of the original space). As is commonly done, we use TSSE as a measure of how well the k-means algorithm has identified reasonable clusters and an indication of how naturally the data can be separated into clusters given the particular representation. To evaluate whether or not those clusters are meaningful for a particular application, different metrics must be used.

We used the standard metrics of purity and entropy to evaluate the quality of the clusters with respect to the various target attribute labels. Purity with respect to a target attribute X is defined as

$$\begin{aligned} \text{purity}(X) &= \sum_{i=1}^k \left(\frac{|C_i|}{n} \max_j \left(\frac{m_{ij}}{|C_i|} \right) \right) \\ &= \frac{1}{n} \sum_{i=1}^k \max_j (m_{ij}) \end{aligned} \quad (4)$$

where n is the total number of instances, $|C_i|$ is the number of instances in cluster C_i , and m_{ij} is the number of instances of C_i with label $x_j \in X$. Larger values for purity represent a better clustering with respect to the class.

Similarly, entropy with respect to a target attribute X is defined as

$$\begin{aligned} \text{entropy}(X) &= - \sum_{i=1}^k \frac{|C_i|}{n} \sum_{j=1}^{|X|} \left(\left(\frac{m_{ij}}{|C_i|} \right) \log_2 \left(\frac{m_{ij}}{|C_i|} \right) \right) \\ &= \frac{1}{n} \sum_{i=1}^k \sum_{j=1}^{|X|} \left(m_{ij} (\log_2 |C_i| - \log_2 m_{ij}) \right) \end{aligned} \quad (5)$$

where n is the total number of instances, $|C_i|$ is the number of instances in cluster C_i , $|X|$ is the number of labels for target attribute X , and m_{ij} is the number of instances of C_i with label $x_j \in X$. Smaller numbers for entropy represent a better clustering with respect to the class.

For highly skewed class distributions (i.e., nearly all instances are of the same class), however, both the purity and entropy measures can give misleadingly favorable scores on random clusters. Consider a clustering that produces a single cluster (all instances in the same cluster) for the *Holding* target attribute (as described below). Such a cluster does not capture anything interesting about the domain, but has a purity of 0.989 and an entropy of 0.087, both which are reasonably close to the optimal values of 1 and 0, respectively. To compensate, we also calculated adjusted versions of purity and entropy that accounted for the class distribution by dividing by the frequency of the label on the labeled instances (equivalently, multiply by its reciprocal). By doing so, each target attribute label is given equal weighting in the computation of the adjusted metric. Such a metric can be thought of as class-specific (evaluating the clusters with respect to the classes) rather than instance-specific.

The re-weighting is pervasive since it is part of any term that involves an instance count, leading to a deceptively complex-looking formula. We define the adjusted purity, $purity_A$, as

$$\begin{aligned} purity_A(X) &= \sum_{i=1}^k \left(\frac{|C'_i|}{n'} \max_j \left(\left(\frac{|X|}{w_j} \right) \frac{m_{ij}}{|C'_i|} \right) \right) \\ &= \frac{1}{n'} \sum_{i=1}^k \left(\max_j \left(\frac{m_{ij}}{w_j} \right) \right) \end{aligned} \quad (6)$$

Similarly, we define the adjusted entropy, $entropy_A$, with respect to a target attribute X as

$$\begin{aligned} entropy_A(X) &= - \sum_{i=1}^k \frac{|C'_i|}{n'} \sum_{j=1}^{|X|} \left(\left(\frac{m_{ij}}{w_j |C'_i|} \right) \log_2 \left(\frac{m_{ij}}{w_j |C'_i|} \right) \right) \\ &= - \frac{1}{n'} \sum_{i=1}^k \sum_{j=1}^{|X|} \left(\frac{m_{ij}}{w_j} \log_2 \left(\frac{m_{ij}}{w_j |C'_i|} \right) \right) \end{aligned} \quad (7)$$

with the terms defined as above.

Surprisingly, such a weight-adjusted version of purity and entropy has not gained widespread adoption, for we could not find a mention of it in the literature, though similar approaches do exist. The V-measure (Rosenberg & Hirschberg, 2007) is an entropy-based cluster evaluation metric that normalizes for the class distribution and also incorporates a factor for the number of clusters, which we have not done. Weighted-adjusted versions of several other popular evaluation metrics have also been developed to account for skewed distributions (Tan, Steinbach, & Kumar, 2006).

Finally, we used accuracy as our metric for the classification task, defined as follows:

$$accuracy(X) = \frac{\sum_{i=1}^n I_{eq}(p_i, t_i)}{n} \quad (8)$$

where n is the total number of instances, p_i is the predicted class of instance i , t_i is the actual class of instance i , and I_{eq} is an indicator function that is 1 when its arguments are equal, zero otherwise.

	Holding	Deviation
Positive instances	1071	187
Negative instances	163133	60
Percent Positive	0.7%	75.7%

Table 1. Number of positive and negative instances for each labeling scheme.

CLASSES- As mentioned previously, our original impetus for exploring track data was understanding how pilots react to convective weather, specifically, when they choose to deviate. Research into this issue is being incorporated into the Convective Weather Avoidance Model (CWAM) (DeLaura & Evans, 2006; DeLaura, Robinson, Pawlak, & Evans, 2008). CWAM uses meteorological products such as echo top (a measure of the height of the storm) and vertical integrated liquid (a measure of precipitation) to calculate the likelihood that a pilot will avoid a given region of the storm. The storm is divided into polygons at different flight levels, and various levels of probability. A limited validation study (Chan et al., 2007) showed that the CWAM model was reasonably accurate, but the validation method was limited by the manual identification of weather-caused deviations on a flight-by-flight basis. This validation study provided class data for us to test against, and also provided our motivation- to provide automated methods that generate similar class labels. Instances were labeled as *Deviation* positive if the flight plan intersected a CWAM polygon that the flight track avoided through a lateral deviation (i.e., by going left or right of the storm, but not over or below it).

However, only a very small subset of the approximately 160 thousand instances were classified in the CWAM validation study (see Table 1). To compensate, we created a new dataset that labeled all instances in the dataset as either holding patterns or not. This also gave us a different application to evaluate our representations on. Given the number of instances, we chose to label the instances automatically instead of manually. Obviously, with an automatic method of creating such labels, it is not necessary to use data mining techniques to recreate the same labels. Regardless, the ability to recognize such features would show some capability of the

representation and data mining algorithm to identify salient features in the domain. An instance was labeled as *Holding* positive if at any point, the flight track segment included a heading that was offset 360° or more from some other heading in the segment. As would be expected, a very small percentage of the instances displayed holding patterns, as can be seen in Table 1.

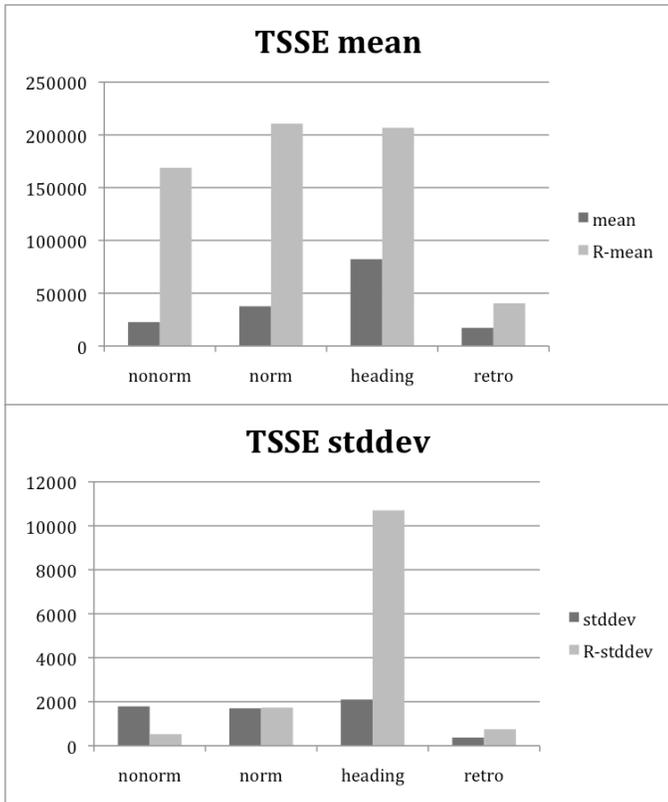


Figure 3. Total Sum of Squared Error (TSSE) for each representation. The mean and standard deviation (stddev) are given over all clusters. The R-mean and R-stddev are the mean and standard deviation from randomly generated clusters, respectively, created for comparison purposes; they should be markedly higher than the non-random clusters.

RESULTS

CLUSTER EVALUATION- The k-means algorithm was run with $k = 100$ ten times for each representation, each time with a different random seed, to account for the variation caused by different choices for initial centroids. To create the random clusterings, each single clustering produced by k-means was randomized by creating clusters of approximately the same size, but with randomly chosen members. This was repeated ten times per k-means clustering, resulting in 100 random clusterings for every representation. These random clusters have the same structure in terms of the number of clusters and their sizes, but presumably capture no useful information. By contrast, a clustering that captures useful information about the domain would presumably produce better results, in terms of our chosen metrics.

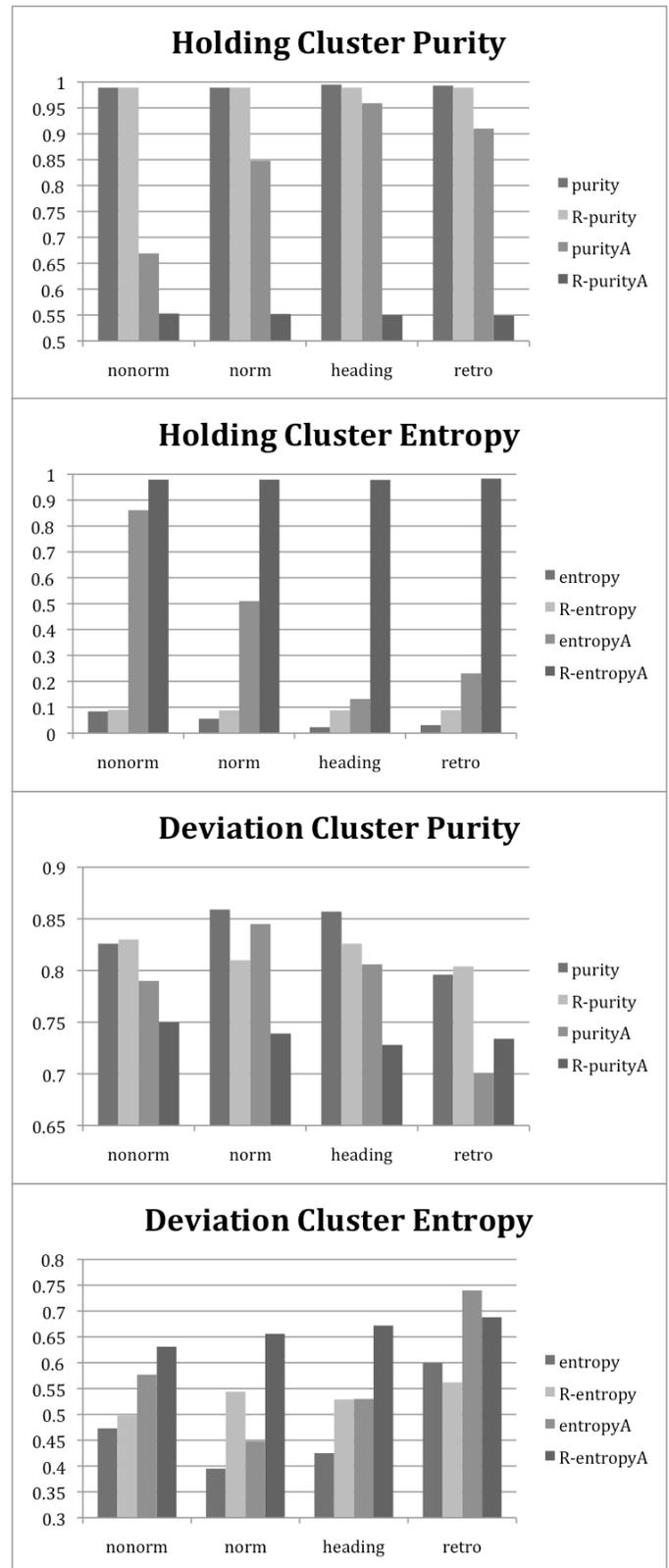


Figure 4. Purity and Entropy cluster metrics. Purity and entropy range from zero to one, with large purity values and small entropy values indicating good clusters. PurityA and EntropyA are the distribution-adjusted metrics, and R represents the metrics on the randomly generated clusters.

Figure 3 shows the results on the total sum of squared error metric (see section 4.1) for each representation. Each representation had a much lower TSSE scores than the corresponding random clusters. This shows that the k-means algorithm is able to pick out some commonalities in the data, but does not tell us much about the data itself: we would expect to k-means to outperform a random clustering on random data (i.e., pure noise), as well. However, the *Holding* representation has a noticeably higher TSSE than the others, perhaps indicating that clusters are not as distinct in this domain. On the other hand, the TSSE was very low for the *retro* clusters, particularly for the randomly generated clusters, indicating that this there was less relative difference between instances in this representation, a potential problem.

Figure 4 shows the purity and entropy for the clusters on both sets of labels, in terms of the original metrics and our adjusted versions. For the *Holding* experiment, the predominance of the non-*Holding* label makes the non-adjusted metrics difficult to interpret. Using the adjusted metrics, it is clear the *heading* and *retro* representations are doing a reasonable job of separating the classes, with *heading* performing the best. In contrast, the distance offset measures of *norm* and *nonorm* are not separating as well, with *nonorm* apparently not finding much difference between the *Holding* on non-*Holding* instances. This matches our expectation, as *heading* and *retro* should have distinctive characteristics for holding patterns (consistent angle of change and retrogression on the flight plan, respectively). In contrast, distance from the flight plan is a subtler signal for holding.

The results for the *Deviation* labels are not as easily interpreted. The metrics show clearly poor results for the *retro* representation, so it is unlikely that meaningful separation can be achieved with this representation for *Deviation*. The *heading* and *norm* representations are able to achieve some separation, but clearly less so than was possible with the *Holding* labels. Though superior to *retro*, it is not clear that the *nonorm* is a suitable choice for the *Deviation* application. Of course, there is one key feature for *Deviation* in all these representations- the location of weather.

CLASSIFICATION EVALUATION – Though the cluster evaluation with *Holding* relatively promising, our cluster evaluation for the *Deviation* labels failed to establish any of our representations as adequate. However, the training phase in classification can isolate structure that might be less dominant in a clustering context. Therefore, we wanted to try to classify according to the *Deviation* label, as was our original goal. We also added a simple additional set of weather features: for each minute of data of flight track and flight plan, a corresponding binary feature was set to one if the point

was within a CWAM polygon, zero otherwise. This resulted in a tripling in the number of features (the original set plus weather data for the flight track and weather data for the flight plan).

Figure 5 shows the classification results for various algorithmic and representational combinations (see the algorithm subsection for a brief description of each algorithm). For comparison, an additional algorithm, ZeroR, was used. ZeroR does not use any of the features, instead returning the majority class. A good representation/algorithm combination should outperform ZeroR.

Unfortunately, none of the algorithm/representation pairs performed particularly well on the full dataset when compared to the ZeroR baseline. The addition of the weather information appears to be important, indeed, all algorithms outperformed ZeroR when using weather information only, though sometimes only slightly. Overall, the simple DecisionTable performed best, while the Naïve Bayes classifier had generally unacceptable performance. For the most part, it is not clear that the flight track/flight plan representations add anything useful when evaluated on the full dataset, though the decision tree and decision table had somewhat better results when the *norm* or *heading* representations were coupled with weather information than when only weather was used.

Further examination of the *Deviation* labels showed that they did not fit our expectations. We had expected that every instance labeled should have a flight plan intersecting weather, according to the description of the original study; in particular, according to the stated conditions a deviation could only be defined when the flight plan intersected weather. (Situations where pilots avoided weather off of their flight plan were not included in the original study). However, in an overwhelming number of instances (139 out of 247), the flight plan did not intersect a CWAM polygon. A much smaller number of instances violated other assumptions, such as a flight track labeled as intersecting weather when it did not, and flights labeled as deviating that did not drift more than 1.5 nautical miles from their flight plan.

We would not expect our representations to be able to correctly classify instances when our assumptions about the data are violated. To test this theory, we divided the labeled data into two subsets: one that matched the assumptions above and one that did not. Running the experiments again, we found that no learning occurred on the subset that violated our assumptions above, as we had expected. Unfortunately, only 100 of the original 247 instances fit our assumptions, and of these 63 were labeled positive for *Deviation*. The small size is problematic because training is generally less effective on small datasets.

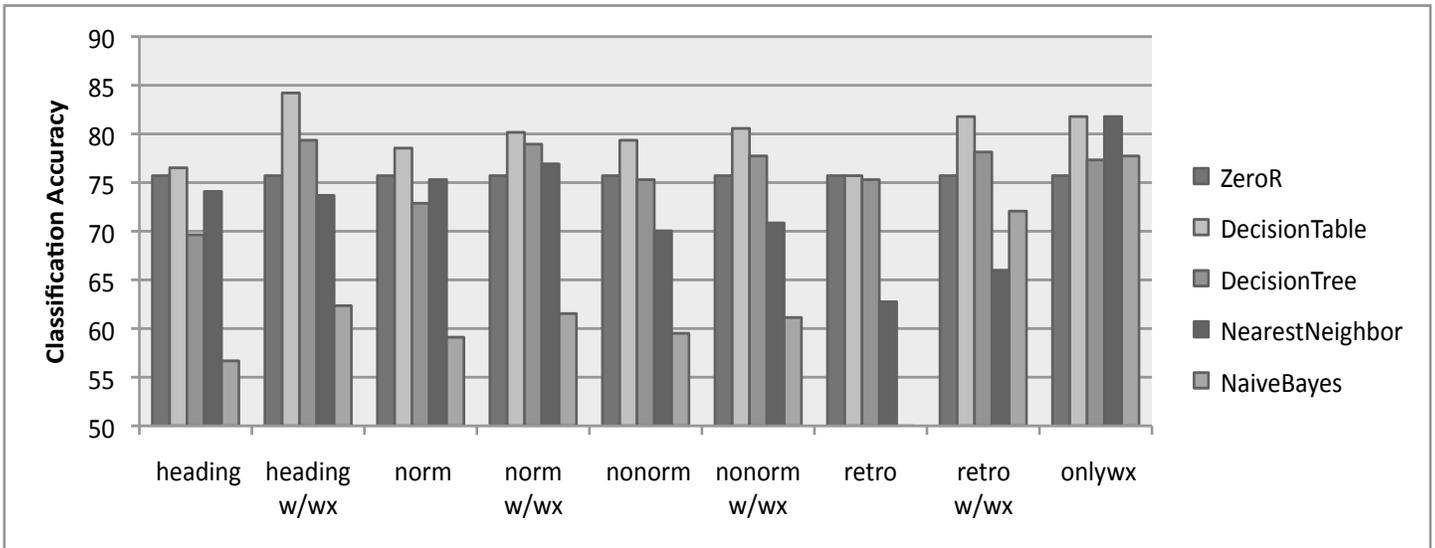


Figure 5. Classification on all labeled instances for *Deviation*. The w/wx variants had additional features showing if the corresponding flight track and flight plan points were within a CWAM polygon. The onlywx representation had only weather information and no other feature.

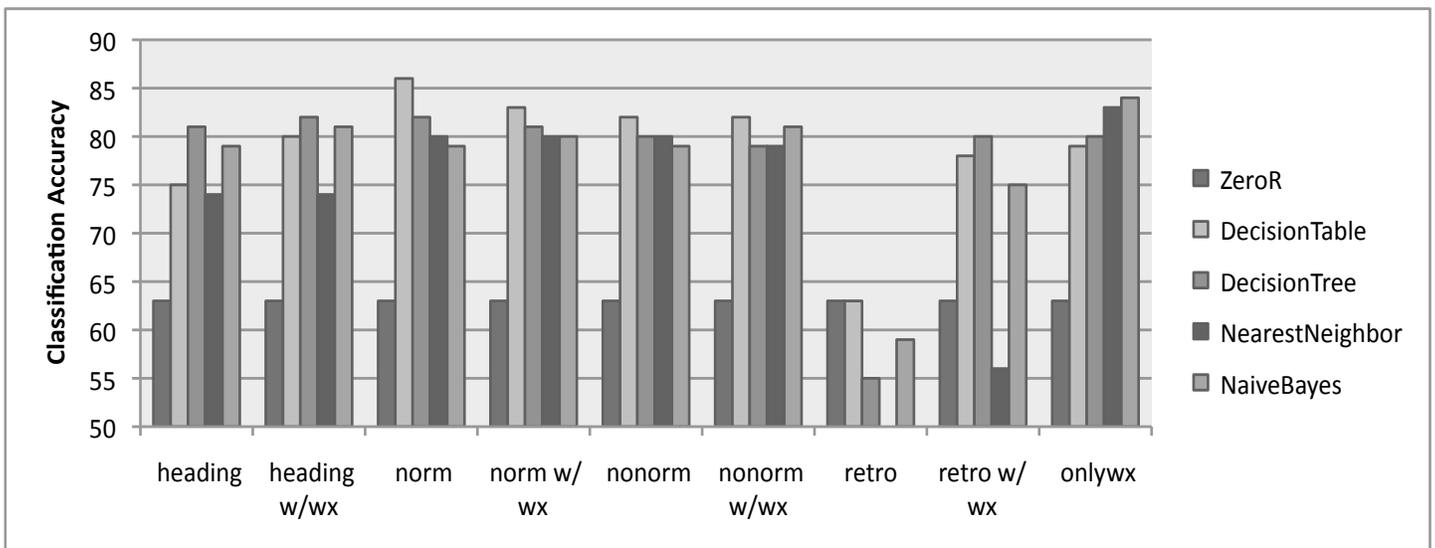


Figure 6. Classification on *Deviation* subset where flight plan intersects weather. The w/wx variants had additional features showing if the corresponding flight track and flight plan points were within a CWAM polygon. The onlywx representation had only weather information and no other feature.

Figure 6 shows the classification results on the subset of data that matched our previously stated assumptions. Consistent with our earlier findings, the *retro* representation does not seem to be a good choice. Though the weather features appear to be important, the difference is not as clear as on the full dataset. Indeed, the best classification rate overall came on the *norm* representation without weather, and for the *heading* and *nonorm* representations, some algorithm on a representation without weather performed comparably to the best performer with weather. Given the small size of

the dataset, small percentage differences are not presumed to be significant, and perhaps more importantly, some of the relative rankings of representation/algorithm combinations could change with more training data.

DISCUSSION

Of the two sets of labels, the tasks associated with the *Holding* labels were better suited to our representations. Even though we did not use these labels in our

classification task, one reason *Holding* may have been easier to identify is because the dataset was fully labeled. Initial experiments showed that a fairly large number of clusters were needed to separate the data into reasonable-looking sets. However, this may have worked against the *Deviation* related tasks, since fewer labels meant that even random clusters would do a good job of separating the data (since fewer instances of any label would be likely to share a cluster). Also, the definition we used for *Holding* was simple and concise. In contrast, the *Deviation* labels were made by hand and had no concise definition. Though such manual labels are presumably of higher quality than a simple automatic definition, differences in judgment mean that they may not be globally consistent (same for all persons) or even locally consistent (instances labeled consistently by the same person).

Even with the complete set of labels for the *Deviation* task, the small size of the dataset is challenging. There is a relationship between the number of features and the amount of data required to adequately learn, and for most if not all of our representations, the small dataset size hurt performance. In nearly all cases, a condition known as overfitting occurred, where the classification performance on the training set (which we did not report here) far exceeds that on the testing set. In such cases, the learning algorithm mistakes random differences between instances for meaningful ones and actually creates a worse classifier. Another issue is that most algorithms have some parameters that should be tuned, but again this was difficult with such a small dataset. A larger dataset or eliminating or combining features can address these issues.

However, the biggest concern from an experimental standpoint was the apparent mismatch between our expectation and the actual *Deviation* labels. A small number of disagreements are to be expected, either from mis-labeling or from differences of opinion on difficult cases, but the number of flight plans erroneously marked as intersecting weather far exceeded a tolerable number. Visual analysis of the flight plans showed close agreement with the plots of the original study and were mostly unambiguously clear of weather, ruling out the possibility of different flight plan interpretations or a prevalence of difficult to judge cases. When limited to the fifteen minute window, it is unlikely that we could have done better classifying for *Deviation* by hand.

When we looked beyond the fifteen minute time window, however, it became clear that many of the flight plans did indeed intersect weather at some point. In the subset of cases we examined, we observed two ways in which the data was labeled other than advertised: either the deviation occurred in a later fifteen minute window than marked (which would result in two incorrect labels, one for the early mis-labeling and one for the later missed

label); or the deviation occurred during a time window greater than fifteen minutes (including both the initial deviation and the time the flight plan intersected weather). Neither one of these cases would have had any adverse effect on the original study (as it was not sensitive to such alterations), but was a problem with our re-purposing of the data.

In any case, when our assumptions for the data are met, we were able to classify the instances with a high degree of accuracy. At the very least, then, there is a certain subset of the data that can be automatically identified and reasonably classified. The fact that this subset can be classified well without using information about the weather indicates that the weather interpretation is probably not an issue.

Though it seems clear that our representations are capturing useful information, creating a system to classify deviations is not yet feasible. A subject matter expert involved in the original study has tentatively validated our analysis of the discrepancies of the *Deviation* labels, but a more extensive examination is warranted, as is a larger set of correct labels. Also, our sample of labels is biased by the instances the analysts chose to label. The accuracy on instances outside of this subset should be evaluated. Finally, even if these issues are satisfactorily resolved, there is the near certainty of less than perfect classification accuracy. This could be handled in one of two ways. As the first choice, if the classifier can be tuned so that it classifies correctly in a well-defined majority of cases, with the ambiguous minority checked by hand. For instance, if it always correctly classifies non-deviations, but occasionally misclassifies deviations, the much smaller set of instances classified as deviations could be checked by hand. Alternatively, within the original CWAM validation study, the classification error can be interpreted as an error bound in the study.

Even without these improvements, the ability to correctly identify deviations in a subset of the data may lead to new insight on when and how pilots deviate when encountering convective weather.

CONCLUSION

The focus of our study is to evaluate several candidate representations for data mining of flight track and flight plan information. We evaluated four representations on clustering and classification tasks, and used two different sets of labels. Which representation performed best was not always consistent for all tasks. However, the *heading* representation was perhaps the most consistently beneficial representation, followed by *norm*. In contrast, the *retro* representation did consistently poorly. Between *nonorm* and *norm*, *nonorm* rarely provided better results than *norm* but often did worse, so relative distance to

flight plan is probably more important than absolute distance.

Further examination of the flight track data shows that it is fairly noisy, in the sense that the positional information for the aircraft is not very precise in some cases. This would manifest itself in several ways; aircraft rapidly accelerating and then coming to a near-stop; aircraft occupying the same point for consecutive minutes (freezing in midair); aircraft doing an immediate 180 and then back again (or similarly impossible maneuvers); or in extreme cases, teleporting backwards to repeat some large portion of the flight plan, only to teleport further ahead. By no means is it unusual to encounter noise in data mining, and the flight track noise is not necessarily worse than usual. However, the type of noise does not necessarily affect all our representational choices equally, in particular *retro* would be more affected by such noise than *norm* or *nonorm*. Without correcting for noise (which we did not do), this may be enough reason to prefer one representation over another.

ACKNOWLEDGMENTS

The author would like to thank William Chan, Mohamad Refai, Deepak Kulkarni, Yao Wang, and Professor Yi Zhang for their feedback and assistance with interpreting the dataset.

REFERENCES

1. Aha, D., & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning*, 6, 37-66.
2. Callahan, M. B., DeArmon, J. S., Cooper, A. M., Goodfriend, J. H., Moch-Mooney, D., & Solomos, G. H. (2001). Assessing NAS Performance: Normalizing for the Effects of Weather Paper presented at the The 3rd USA/Europe Air Traffic Management Research and Development Symposium.
3. Chan, W. N., Refai, M., & DeLaura, R. (2007). An Approach to Verify a Model for Translating Convective Weather Information to Air Traffic Management Impact. Paper presented at the 7th AIAA Aviation Technology, Integration and Operations Conference (ATIO).
4. DeLaura, R., & Evans, J. (2006). An Exploratory Study of Modeling Enroute Pilot Convective Storm Flight Deviation Behavior. Paper presented at the 12th Conference on Aviation, Range and Aerospace Meteorology.
5. DeLaura, R., Robinson, M., Pawlak, M., & Evans, J. (2008). Modeling convective weather avoidance in enroute airspace Paper presented at the 13th Conference on Aviation, Range and Aerospace Meteorology.
6. Freeman, H., & Glass, J. M. (1969). On the Quantization of Line-Drawing Data. *IEEE Transactions on Systems Science and Cybernetics*, 5(1), 70-79.
7. John, G. H., & Langley, P. (1995). Estimating Continuous Distributions in Bayesian Classifiers. Paper presented at the Estimating Continuous Distributions in Bayesian Classifiers.
8. Kohavi, R. (1995). The Power of Decision Tables. Paper presented at the European Conference on Machine Learning.
9. Koplowitz, J., & Toussaint, G. T. (1976). A unified theory of coding schemes for the efficient transmission of line drawings. Paper presented at the IEEE Conference on Communications and Power.
10. Loncaric, S. (1998). A Survey of Shape Analysis Techniques. *Pattern Recognition*, 31(8), 983-1001.
11. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. Paper presented at the Fifth Berkeley Symposium on Mathematical Statistics and Probability.
12. Mitra, S., & Acharya, T. (2007). Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(3), 311-324.
13. Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers.
14. Rosenberg, A., & Hirschberg, J. (2007). V-Measure: A conditional entropy-based external cluster evaluation measure. Paper presented at the The 2007 Joint Meeting of the Conference on Empirical Methods on Natural Language Processing (EMNLP).
15. Song, L., Wanke, C., & Greenbaum, D. (2007). Predicting Sector Capacity for TFM. Paper presented at the The 7th USA/Europe Air Traffic Management Research and Development Seminar.
16. Tan, P.-N., Steinbach, M., & Kumar, V. (2006). Class Imbalance Problem. In *Introduction to Data Mining* (pp. 294-298): Addison-Wesley.
17. Witten, I. H., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques* (2nd ed.). San Francisco: Morgan Kaufmann.