Trajectory classification based on machine-learning techniques over tracking data

Jesus García
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
jgherrer@inf.uc3m.es

Oscar Pérez Concha
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
oscar.perez.concha.uc3m.es

José M. Molina
Computer Science
Department-GIAA
Universidad Carlos III de Madrid
Colmenarejo, Spain
molina@ia.uc3m.es

Gonzalo de Miguel
Signal Systems and Radio-communications
Department-GPDS
Universidad Politécnica de Madrid
Madrid, Spain
gonzalo@grpss.ssr.upm.es

Abstract - This work addresses the application of a machine-learning approach to classify ATC trajectory segments from recorded opportunity traffic. It is based on the mode probabilities estimated by an IMM tracking filter operating forward and backward over available data. A learning algorithm creates a rule base for classification from these data, once they have been properly prepared. Performance of this data-driven classification system is compared with a more conventional approach based on transition detection on simulated and real data of representative situations. The offline processing of real data allows an accurate classification of manoeuvring segments, with the possibility of synthesizing ground truth lines for performance evaluation.

Keywords: Trajectory classification and reconstruction, data mining, artificial intelligence.

1 Introduction

Trajectory classification and reconstruction over recorded real data (known as opportunity traffic) is an essential process to validate and evaluate data processing systems operating in critical applications such as air traffic control [1,2]. This process is applied off-line over recorded files, so it basically works as an especial multi-sensor fusion system in which we take advantage of knowledge of both past and future target position reports to improve the performance of classification and estimation algorithms.

A reconstruction system typically transforms multi-sensor plots to a common coordinated system (e.g. stereographic projection), associates them to trajectories representing real targets and corrects systematic errors. Then, for each trajectory, segments of different modes of flight (MOF) are identified. Each segment represents a time interval in which the aircraft is flying in a different type of motion (uniform, turns, decelerations, etc.).

The segments available after MOF classification are a valuable description of real data, providing information to analyse the behaviour of interesting objects (where uniform motion flight and manoeuvres are performed, statistics of magnitudes, durations, etc.).

In order to take advantage of the off-line processing, an appealing strategy is performing a double tracking loop in the forward and backward directions to identify transitions [7]. In this work, we propose an IMM tracking filter whose modes are matched to different manoeuvring MOFs. The basic idea is that this kind of filters is optimized to detect start edges but may have problems to detect ends of manoeuvre segments (slow convergence). The running of forward and backward filters may potentially solve this problem, although it is needed an analysis of the evolution of probability sets to clearly identify the MOF transitions.

In this work it is explored a classification system to automatically extract the MOF information from IMM mode probabilities, based on a data mining strategy. Data mining techniques derive from Statistical and Machine-learning fields, and are typically used to "discover" hidden patterns and relations or build useful prediction models [3]. In this case, it is evident that MOF information is in the IMM mode transitions, but there is not an exact model of that relation to accurately perform the classification, and this learning paradigm can be useful to succeed in this task. One of the key aspects in this process is the preparation of data, including the selection of input data, problem formulation and model representation (rules, decision/regression trees, etc.). After this preparation, the algorithms search in the space of model parameters for those most suitable to the training data sets. We add also a post-processing step to have a smooth output from classified samples, avoiding short-term jumps which make no sense in this application since categories correspond to motion states.

1 Funded by projects CICYT TEC2005-07 186/TCM and CAM MADRINET S-0505/TIC/0255
The goal of this work is to create a system to automatically exploit the available data and search an accurate classification model. The results are compared with a more conventional segmentation with edge detections to show the out-performance and capability to obtain general models from training data.

Next section outlines the tracking structure to process the available data and states the MOF classification problem, with a first potential solution based on edge detection. Section 3 presents the approach based on machine learning for this application, in the phases of preparation and training, algorithm and post-processing. Section 4 presents the results obtained with simulated and real trajectories with both approaches, and some conclusions are summarized at the end.

2 IMM filter for segment classification

2.1 Filter structure

The IMM tracking filter for this application is composed of three Kalman filters matched to uniform motion, transversal and longitudinal manoeuvres (Figure 1). The three Kalman filters operate in parallel, each one matched to a specific type of motion (parameters in plant noise characteristics). All modes in this structure share a common state representation and dynamic model, simplified to have 2 states (position and velocity) in 2D stereographic plane and constant-velocity prediction. The state vector is

\[ \mathbf{x}[k] = \begin{bmatrix} x[k] \\ \dot{x}[k] \end{bmatrix} \]

with the corresponding 4x4 error covariance matrix.

The difference among the three modes is in the plant-noise covariance matrix. There is a uniform-motion mode with zero-matrix (K=1), and two modes with a directional covariance matrix projected along the transversal and longitudinal (K=2,3) directions. These two last modes are intended to detect manoeuvres aligned with their noise directions, as indicated in the figure.

\[ \text{transversal plant noise} \]
\[ \text{longitudinal plant noise} \]
\[ (\hat{s}_x[k], \hat{s}_y[k]) \]
\[ \text{filtered} \]

Figure 2. Directional Kalman filter with oriented plant-noise covariance matrix

So, the filter structure is thought to capture the transitions among the different types of manoeuvres and identify the most probable MOF in each sample. Although the manoeuvring modes (K=2,3) are not optimized to provide high-accuracy estimates during manoeuvres, the idea is that they should be activated in the appropriate circumstances in order to reconstruct the MOF segments.

2.2 MOF identification problem

The classification problem can be stated as an accurate indication of current mode of flight, taking the output of IMM mode probabilities in the forward and backward directions as input. The input variables are:

- Forward Uniform Mode: \( \mu_{f1}[k] \)
- Forward Transversal Mode: \( \mu_{f2}[k] \)
- Forward Longitudinal Mode: \( \mu_{f3}[k] \)
- Backward Uniform Mode: \( \mu_{b1}[k] \)
- Backward Transversal Mode: \( \mu_{b2}[k] \)
- Backward Longitudinal Mode: \( \mu_{b3}[k] \)

The output is the MOF label, with the categories:

- Uniform motion
- Transversal manoeuvre
- Longitudinal manoeuvre
- Combined manoeuvre

Besides, this classification problem for MOF identification in the trajectory has two characteristics to be taken into account:

- Information is not only in the current set of probabilities for a time instant, but also in the samples within a certain time interval. For instance, a transition is identified as a sudden drop after stable samples. So, the output should depend not only in the current samples but consider also their neighbourhood.
- MOF cannot freely change from sample to sample, but it extends over time intervals corresponding to real aircraft modes of motion. So, the output is correlated and classification decisions over different samples are not independent.

2.3 A classifier based on edge detections
A method to derive the MOF segments, employed as a benchmark in this analysis is summarized here. It is based on the identification of edges corresponding to starts and ends of typical manoeuvres analysing drops in uniform-mode probability. Transitions from uniform motion to manoeuvres are quick (high residual) while transitions of manoeuvres coming back to uniform motion are slow (especially in this IMM structure with modes differing only in the plant-noise covariance). So, the forward and backward runs are used to separately identify manoeuvres starts and ends. It is illustrated in the following figure:

After the edges have been detected, the decision about the type of segment is taken considering probabilities for each mode in the two directions, $\mu_f[k]$ and $\mu_b[k]$. The mode with majority number of samples is assigned to this time interval. For instance, the application of this process to a short manoeuvre is depicted in the following figures, with the resulting classification below.

Forward probabilities are represented solid lines and backward probabilities with dashed lines. The uniform motion probability, in blue, clearly falls in the time instant 220, with a slow transition back to uniform motion from $t=240$ until $t=300$. The backward run shows that it drops at $t=240$ and then comes back from $t=230$ until $t=120$ s. In both runs, the dominant mode while uniform mode probability has drop is turn (in red). Therefore, the central segment is classified as turn mode, and the two adjacent intervals, from beginning to start of manoeuvre and from end of manoeuvre to end of data, are classified as uniform-motion segments, with the transition times given by the drop of forward and backward runs.

3 Machine-learning approach for MOF trajectory classification

An objective of the MOF classification is to achieve high reliability about the description generated of available data. The design of classification system has been carried out based on the paradigm of data-driven modelling, learning from examples an unknown relation. In this case, the selected algorithm was PART algorithm (Partial Rules from Decision Trees) [3], a model tree induction algorithm for predicting categorical variables using both categorical and numeric attributes.

3.1 Data preparation

The input data for model induction is prepared in a tabular form with the forward and backward probabilities in each sample, and the known MOF category in a set of training simulated trajectories. In order to take into account not only the current sample but also the samples in a close time interval where information is available, the values corresponding to several samples before and after current time are also included. The table prepared has the following aspect:
Table 1. Data preparation for MOF classification

<table>
<thead>
<tr>
<th>Sample M</th>
<th>µf[k]</th>
<th>µb[k]</th>
<th>µf,k-1[k]</th>
<th>µb,k-1[k]</th>
<th>...</th>
<th>µf,k-1[k-M]</th>
<th>µb,k-1[k-M]</th>
<th>...</th>
<th>µf,k,M</th>
<th>µb,k,M</th>
<th>MOF</th>
</tr>
</thead>
</table>

The number of neighbour samples tried in this study varied from M=0 to M=5 so the total number of input parameters is 6*(2M+1). In the worst case (M=5), a model depending on 66 input variables was used. The impact of M on the classification performance is included at the end. In any case, it is important to notice that the edge-detector classifier also uses context information since transitions are detected after five samples at uniform motion. All the input parameters are mode probabilities and so they are in the interval [0,1], while the MOF category takes the values {1, 2, 3, 4}, corresponding to the labels {Uniform, “Transversal Manoeuvre”, “Longitudinal Manoeuvre”, “Combined Manoeuvre”}.

3.2 Training trajectories

With this disposition of data, a set of 16 simulated trajectories were generated to train the system and identify the MOF segments in representative situations. The following cases were considered:

Simple trajectories (8)
- Two radial trajectories with speed 150 m/s, turn of 45° with accelerations of 2.5 and 6 m/s², at 100 Km from radar (the second one is quite faster).
- Two tangential trajectories with speed 150 m/s, turn of 45° with accelerations of 2.5 and 6 m/s², at 100 Km from radar.
- Four trajectories analogous to the previous ones, separated now 220 Km from radar in the manoeuvring time instant.

Combined turns (2)
- Double turn of 45° to right and 45° to left, with speed of 150 m/s, acceleration of 2.5 m/s² and distance of 100 Km to radar.
- Analogous to previous trajectory, with shorter turns of acceleration 6 m/s².  

Circular trajectory (1)
- Three whole circles with acceleration of 4 m/s², after uniform segment. Speed of 150 m/s and distance 100 Km to radar.

Hippodrome trajectory (1)
- A hippodrome with turns of acceleration of 4 m/s², speed of 150 m/s and distance 100 Km to radar. In the hippodromes, segments of uniform motion last 100 seconds, and turns 300 seconds, approximately.

3.3 Learning algorithm

The PART algorithm is a variation from algorithm C4.5 proposed by Quinlan [4], one of the most commonly used data mining algorithms and available in many commercial products. Algorithms of this family induce decision trees to classify instances into different categories. The goal is searching which attributes distinguish each category from another, done in a recurrent way to develop the tree. Besides, a heuristic-pruning method is used to avoid over-fitting. It makes the tree less complex and also more general by replacing subtrees with the most common branches or leaves.

The PART algorithm forms rules from pruned partial decision trees built using C4.5’s heuristics. The rules are formed by writing a rule for each path in the tree and then eliminating any unnecessary antecedents and rules. According to Witten and Frank [5], the main advantage of PART over C4.5, illustrated with some results, is that the rule learner algorithm does not need to perform global optimization to produce accurate rule sets. To make a single rule, a pruned decision tree is built and the leaf with the largest coverage is made into a rule.

3.4 Post-processing

A last step in the classification of trajectories applies the condition that MOF variable must not freely change from sample to sample. The categorical values of this variable,
corresponding to different types of motion, are extended over time intervals longer than a minimum length.

This condition has been implemented by means of a post-processing filter applied over the output sequence, a median filter. The median filter considers each sample in the sequence and looks at its nearby neighbours (N samples) to decide which is the most representative value of its surroundings to generate the output. Next Figure illustrates an example calculation with three segments of classes 1, 2, 1, with some noisy samples. The median filter applied has an extent of N=2 samples at both sides. Obviously the effect of the number N on the behaviour is a trade-off between smoothing (large N) and capability to pass short manoeuvres (small N). In this work N=2 at both sides was used.

![Original sequence vs Filtered sequence](image)

Figure 6: Median filter applied to a classification sequence

An example of its application to a situation of MOF classification is in the following figures. Figure 7 depicts the forward-backward IMM probabilities for a trajectory with alternating straight and turn segments (hippodrome). The direct output of classifier is in figure 8, in thin red line, while the filtered output is in thick blue line. The short variations of the output of one or two samples are removed to leave a stable output.

![Mode probabilities in a hippodrome](image)

Figure 7: Mode probabilities in a hippodrome

![MOF classes in a hippodrome and median filter](image)

Figure 8: MOF classes in a hippodrome and median filter

The use of this post-processing phase was needed only in the machine-learning algorithm since it takes independent decisions over each sample. In the case of edge-detector classifier, decisions are taken about manoeuvre starts and ends, and it was not necessary to do this post-processing.

4 Results

With the trajectories described in section 3.2 the PART algorithm was trained to obtain a classification model (rule base). This model was evaluated on three data sets: on the original trajectory data used for training (obtaining the optimistic “re-substitution” error), on independent simulated data corresponding to a different random seed and analogous trajectories, and on real data sets from recorded trajectories. The experiments were conducted on different using WEKA software system [6] to evaluate the classification accuracy over the data sets.

4.1 Sample Results

Some sample results from the whole set of trajectories presented in 3.2 are commented here. For each one, the trajectories is depicted first, then a figure with the IMM mode probabilities, and then a figure with the classification generated both with learning algorithm and edge detection, together with the ideal values. The results presented have been obtained with the test data with a different random seed that data used for training. Although all results were obtained from a varying number of context samples (M=0...5), here we only depict the best case with M=5 samples after and before current one.

4.1.1 Medium and short manoeuvres

In this case we can notice the effect of a sharp manoeuvre (6 vs. 2.5 m/s²) on the transition detector. With a manoeuvre of 2.5 m/s² and duration of 24s both algorithms succeed in the classification (Figure 11), although the edge detector has higher deviations in the start and end of manoeuvres, due to detection delays. In the case of 6 m/s² and duration 12 s, the start and end of manoeuvre are very close (Figure 12) and the edge detector misses the curvilinear segment, while the learning algorithm is able to segment the situations (Figure 13). In the case of a long circular manoeuvre (trajectory 3), both algorithms had a very similar performance.

![Turn with 2.5 m/s2 acceleration](image)

Figure 9: Turn with 2.5 m/s² acceleration
Figure 10: Fwd-Bkd IMM mode probabilities for turn at 2.5 m/s²

Figure 11: MOF output for turn at 2.5 m/s². Blue: PART, Red: edge detector, Black: ideal

Figure 12: Fwd-Bkd IMM mode probabilities for turn at 6 m/s²

Figure 13: MOF output for turn at 6 m/s². Blue: PART, Red: edge detector, Black: ideal

4.1.2 Combined manoeuvres

The presence of alternated segments presents problems for the edge detector when they are very short. In this case we have a hippodrome trajectory (two rounds) where the straight segments are very short (30 s) and detections are delayed, so the whole hippodrome is wrongly classified as turn using the edge-detector classifier. The learning algorithm had capability to distinguish also the alternation of MOF segments for this trajectory. In fact, its resolution would be limited by the extent of median filter in pos-processing phase. In this case it requires that segments extend at least over three samples, with 4 radars of T=12 seconds means manoeuvres as short as 3 seconds. With a mono-radar situation, it would require at least 36 seconds. A similar conclusion is obtained with the analysis of trajectory with alternated longitudinal manoeuvre and straight segments with result in Figure 17.

Figure 14: Simulated hippodrome trajectory

Figure 15: Fwd-Bkd IMM mode probabilities
4.1.3.3 real trajectories

Finally, the classification algorithms were tested with a set of six real trajectories using available recordings. Here, only a uniform trajectory (figures 18-20) and hippodrome (figures 21-23) are depicted. Notice that in this case we have not ideal values for categories (black thick lines in figures 20, 23). In the first case, the presence of additional noise in the IMM mode probabilities does not produce disruption in the classification (both algorithms are robust against noise). In the case of hippodrome, the duration of segments is enough to avoid problems to the edge detector, and here the effect of additional noise produces some isolated errors in the categories generated by the learning algorithm.
4.2 Global Results

The effect of the number of samples used for training on the global error (all data in the 16 simulated trajectories) is shown in Figure 24. The use of training data and independent data has been separated, since the error on training data is always optimistic. We can see a clear out-performance over edge detector and a continuous improvement in the capability to distinguish categories as the number of context samples is increased.

5 Conclusions

Machine-learning for classification over tracking output is an effective tool in the analysis and reconstruction of recorded data. Since accurate models relating the available variables and final state is not available, the use of machine-learning paradigm allows a significant improvement in the system performance. Although a conventional segmentation with edge detection provides acceptable performance in most situations, the machine-learning power allows the resolution in problematic cases of identifying very short MOF segments.

The use of a representative algorithm, PART, is complemented in this work with the proposal for data preparation, collecting context information to decide the category at each moment, and a post-processing to filter out variations of unreal duration. The trained model has been tested with simulated and real trajectories with satisfactory results. A certain degradation due to presence of noise in real data suggest a following step using real data in the training phase to derive a more robust model.

References