Trajectory Clustering and Coastal Surveillance

Anders Dahlbom, Lars Niklasson

School of Humanities and Informatics, University of Skövde
P.O. Box 408, SE-541 28 Skövde, Sweden
firstname.lastname@his.se

Abstract: In this paper we explore trajectory clustering as a means for representing the normal behavior of vessels in a coastal surveillance scenario. Trajectory clustering however suffers from some drawbacks in this type of setting and we therefore propose a new approach, spline-based clustering, with a potential for solving the task of representing the normal course of events.

1 Introduction

Decision support systems can be vital for performing situation analysis more efficiently. Situation analysis is used to create and maintain a state of situation awareness in a decision maker [Ro01], which is of high importance to make good decisions. Automated systems that can discern interesting information in a continuous stream of data are of great interest. These can limit the information gap often experienced by decision makers [En00]. This entails systems that need to be aware of the *normal course of events*, to detect uncommon or interesting patterns of activity. This motivates the interest for the task of representing normal behavior, as well as detecting anomalous behavior.

2 Trajectory clustering for anomaly detection

In a coastal surveillance setting, anomaly detection is concerned with representing the normal course of events occurring at sea. Automatic Identification Systems, radars, and other surveillance equipment supply tracks of individual entities. The task for an anomaly detection system is to maintain a knowledge base of normalcy, to find entities behaving anomalous. Trajectory clustering [PFS05, SPF06] is a spatio-temporal clustering algorithm, which maintains a forest of trees to represent the normal course of events. Each tree can represent several similar trajectories. Trajectory clustering is based on prefix matching, i.e. the initial parts of trajectories need to match for them to be clustered. A trajectory t is represented by a list of vectors, where each vector v_i describes the spatial position of an observation. Similarly, a cluster c is represented by a main trajectory consisting of a list of vectors, where each main trajectory point w_j consists of physical coordinates and a variance. The distance between a cluster c and a trajectory t, is calculated as the mean of the normalized Euclidean distance between each trajectory point v_i and its nearest cluster point w_j, found within a sliding temporal window of increasing size. The authors suggesting trajectory clustering [PFS05] state that by increasing the size of the temporal window, matching is allowed under accumulating temporal differences. Trajectory clustering can be divided in three phases: matching, creation, and updating. Figure 1 illustrates an example of the algorithm.

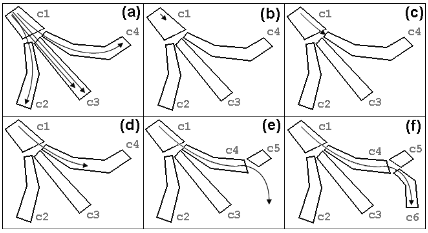

Figure 1: Illustration of how the trajectory clustering algorithm works (adapted from [PFS05]).

In Figure 1a, a number of trajectories have been used to establish a tree with four nodes. The observed trajectories share the initial parts, whilst the latter parts are more specific. In Figure 1b, a new trajectory is detected and matched against node c1. The new trajectory is associated with c1 and the algorithm proceeds to the update phase, where it remains until the trajectory at some point no longer match c1 (Figure 1c). This occurs at the end of c1 and the algorithm directly enters the matching phase. The trajectory is now matched with the children of c1, i.e. c2, c3, and c4. In Figure 1d, cluster c4 has been found to match the trajectory, which is associated with c4. The algorithm again enters the updating phase and remains there until it no longer match c4 (Figure 1e). This time the mismatch is detected in the middle of c4, which is split. The unmatched part is used to create a new cluster c5, which is inserted as a child to c4. The algorithm enters the matching phase and the trajectory is matched against the children of c4. It can be seen in Figure 1f that no matches are found, and therefore, a new cluster (c6) is created and inserted as child to c4. Cluster c6 is from here on created with data from the trajectory.

In the matching phase, a trajectory t is defined to match a cluster c if the distance falls bellow a match cut-off value ε. In the creation phase one main cluster point w_j is created for each new trajectory point v_i, and the variance for each point is set to a predefined value σ_0. In the update phase, the main trajectory of a cluster is updated for each observation, by moving the best matching main cluster point towards it and updating the variance accordingly. The variance and position of the best matching cluster point is updated according to a learning rate α, determining at which speed clusters approach new trajectories. Each cluster becomes a dynamic approximation of the mean and variance of trajectories that have matched the cluster, and there is an exponentially decreasing weight on older trajectories [PFS05].

The authors suggesting trajectory clustering claim that the algorithm offers a natural way of detecting anomalies by assigning counters to each individual cluster [PFS05]. The counters track how many times each cluster has been matched with trajectories, and can be used to assign probabilities to each cluster, by dividing its respective value with the total number of times a trajectory has matched any child of its parent. The probability for a complete trajectory can be derived by multiplying the probabilities for each cluster that the trajectory has been matched with. In other words, any given path through the forest can be assigned a probability by multiplying the probabilities of the clusters that the path consist of. Anomaly detection can be achieved by looking at the derived probability for the path that matches a given trajectory. If the derived probability is low, then the trajectory could be considered anomalous as this path has not been seen too frequently. A trajectory that ends up in the creation phase would of course also be considered anomalous, as this path has never been seen at all.

3 Coastal surveillance

A coastal surveillance scenario has been implemented in a ground target simulator, briefly described in [WP04]. A number of routes have been established along the southern coast of Sweden. These routes are intended to represent the normal course of events for commercial traffic. Simulations have been carried out in which a number of vessels have been created and attached to the routes during a 24h period. The trajectory clustering algorithm has been applied on this scenario.

3.1 Algorithmic problems

To test the matching capabilities of the algorithm, we have applied the training data to the learned clusters. This gives some interesting results, as the training data cannot be matched for more than the initial parts on any trajectory. Analysis of this problem reveals that root clusters are matched perfectly, but child clusters cannot be matched at all. This problem can be traced to the distance equations, which try to match the complete trajectory with the cluster under consideration. However, the part of the trajectory which matched a parent cluster does of course not match the child cluster, resulting in that no matches are found amongst children. Furthermore, an inspection reveals that several similar clusters exist after split points. We have identified the reason for this as follows. Upon reaching the end of a parent, a child that matches a new trajectory should be found. Again, no children can be found since the matching criteria involve matching the complete trajectory, and not only the latter part.

Moreover, some cluster points appear very far away from their neighboring cluster points. This problem has been tracked to two occurring situations in the creation phase of the algorithm: (1) a trajectory creating a cluster is being outrun by a trajectory which is updating the very same cluster and (2) a trajectory updating a cluster forces a split of the cluster, while the cluster is still being created with data from another trajectory. Both cases are concerned with vessels being matched with an incomplete cluster.

3.2 Gaps in the data stream and redundancies in the representation

A more serious problem has however been observed, as there at some points are huge gaps in the clusters. This problem can be tracked to long gaps appearing in the output from the simulation. Even though ground truth has been considered, the simulator still simulates real-world sensors to a high fidelity. Radars exploit differences in Doppler frequency/velocity between the target and the background surface to detect targets. This means that a target needs to have a radial velocity larger than some cut-off value, compared to its background. This is not the case in all real-world situations, and hence, this is reflected in the simulations. Even though this problem can be removed in the simulated data, it raised an important question: is there always a complete coverage in the real-world? In real-world applications the coverage is not nearly complete and available at all times, which questions the usefulness of trajectory clustering for anomaly detection in coastal surveillance. Furthermore, each individual observation on a trajectory is used for cluster creation. Anomaly detection systems are often intended for online use, which put demands on the algorithms to have a low computational complexity and a small memory footprint. This suggests that we do not need to store every observation; it is merely enough to construct clusters from the minimum number of points required to extract the information.

4 Spline-based clustering

The problem with matching difficulties can easily be solved with minor algorithmic modifications, such as for example only matching against the latter parts of a trajectory after a child cluster has been reached. Similarly, the cluster creation problems can be solved by for example associating trajectories to clusters instead of the other way around and then continuously verifying the consistency when a cluster is being created/updated. Another possible solution could be to apply batch learning. The third problem with gaps in the data stream, is however fundamental, and needs to be addressed. It is reasonable to ask whether prefix matching is a good approach towards describing the behavior at sea. In addition, the amount of data stored to represent individual clusters is too extensive as each individual observation is stored.

To lower the complexity of the cluster representation, we propose the use of splines for representing the main trajectory of a cluster. The data redundancy inherent in the observations in coastal surveillance supports this proposal, since vessels often tend to travel in straight lines for long periods of time. Splines and similar curves are described by a number of control points that shape a curve of varying degree. Any point on a curve can be accessed through its control points. A straight line would only require two control points, one for each endpoint, whilst more complex curves require more information. In a worst case scenario one control point would be needed for each observation. This would dynamically be decided when a spline is created and updated. Therefore, splines offer the amount of information inherent in all observations, whilst at the same time possibly allowing for a reduction in computational complexity and memory usage. In the creation phase, a new observation could be included as follows. First, an additional control point is inserted at the end of the spline. Secondly, the difference between this spline, and a spline created if removing the previous control point, is calculated. If the splines only differ slightly, then the previous control point can safely be removed while still keeping the shape of the curve. In an update phase additional control points would be inserted on the spline if necessary. The variance on a spline would be maintained in an additional spline, which is aligned with the positional spline. The second spline would however only be used to shape the variance around the main spline. For more information regarding splines and similar curves see for example [PT95].

The main problem with trajectory clustering has been determined to be that it is based on prefix matching. For some applications this choice is clearly feasible, but not for coastal surveillance where there might be huge gaps in coverage. We would like to match any observed trajectory with any part of a cluster, i.e. not fixing everything around a static prefix. This could however possibly result in exponential computational complexity when matching, and is clearly not suitable in an application intended for online use.

A well established approach for lowering the complexity of algorithms is to design them in a hierarchical fashion. This could be achieved in the proposed spline representation as well. Consider a spline consisting of five control points. Remove the second and fourth control points and achieve another spline. Measure the difference between the new spline consisting of points 1, 3, and 5, with the spline consisting of control points 1, 2, and 3, and another spline consisting of control points 3, 4, and 5. Perform the same complexity reduction once more, to achieve a straight line defined by control points 1 and 5, and calculate the difference between the line and the splines beneath it. Now, to find where on the spline a trajectory match, calculate the distance between the first point and the line created from control points 1 and 5. If this distance is lower than the calculated difference between the line and its underlying spline, then a match occurs somewhere on the spline. Continue by investigating if the trajectory fits any child splines of the matched spline. In the end, a single point on the spline is found, if a match occurs. Hierarchical structures like this usually offer a logarithmic complexity, compared to exponential, and this approach is therefore considered suitable for providing more flexible matching. The usefulness and applicability of the suggested approach is however left for future work, together with a more detailed description.

5 Conclusion

In this paper we have investigated the applicability of trajectory clustering [PFS05, SPF06] for anomaly detection in a coastal surveillance scenario. Some weaknesses have been identified, which questions the usefulness of trajectory clustering in this type of setting. This paper has presented spline-based clustering as a potential extension to alleviate some of these problems. In spline-based clustering, clusters are represented by splines, which effectively compress the main trajectory of clusters, as well as possibly reducing the computational complexity coupled with matching. Finally, we have also addressed some minor algorithmic problems: (1) whole trajectory matching, (2) creation trajectory being outrun, and (3) splits occurring during creation.

Acknowledgements

This work was supported by the Information Fusion Research Program (University of Skövde, Sweden) in partnership with Saab Microwave Systems and the Swedish Knowledge Foundation under grant 2003/0104.

References

[DN07] Dahlbom, A.; Niklasson, L.: Trajectory Clustering for Coastal Surveillance, The 10th International Conference on Information Fusion, Quebec, Canada, 2007.

[En00] Endsley, M. R.: Theoretical Underpinnings of Situation Awareness: A Critical Review. In (Endsley, M. R.; Garland, D. J.): Situation Awareness Analysis and Measurement. Mahwa, NJ: Lawrence Erlbaum Associates, 2000.

[PFS05] Piciarelli C.; Foresti G.L.; Snidaro L.: Trajectory Clustering and its Applications for Video Surveillance, IEEE Conference on Advanced Video and Signal Based Surveillance, 2005.

[PT95] Piegl L.; Tiller W.: The NURBS Book, 2nd edition, Springer, 1995.

[Ro01] Roy J.: From Data Fusion to Situation Analysis, Proceedings of the 4th International Conference on Information Fusion, Montreal, Canada, 2001.

[SPF06] Snidaro L.; Piciarelli C.; Foresti G.L.: Fusion of trajectory clusters for situation assessment, The 9th Int. Conference on Information Fusion, Florence, Italy, 2006.

[WP04] Warston H.; Persson H.: Ground surveillance and fusion of ground target sensor data in a Network Based Defense, Proceedings of the Seventh International Conference on Information Fusion, Stockholm, Sweden, June 28 - July 1, 2004.