The Neighbor Course Distribution Method with Gaussian Mixture Models for AIS-based Vessel Trajectory Prediction

Bjørnar R. Dalsnes*†, Simen Hexeberg ‡, Andreas L. Flåten*, Bjørn-Olav H. Eriksen* and Edmund F. Brekke*
* Department of Engineering Cybernetics
* Norwegian University of Science and Technology (NTNU)
‡ BearingPoint Norway AS
† bjornrda@stud.ntnu.no

***Abstract*—When operating an autonomous surface vessel (ASV) in a marine environment it is vital that the vessel is equipped with a collision avoidance (COLAV) system. This system must be able to predict the trajectories of other vessels in order to avoid them. The increasingly available automatic identification system (AIS) data can be used for this task. In this paper, we present a data-driven approach to predict vessel positions 5-15 minutes into the future using AIS data. The predictions are given as Gaussian Mixture Models (GMMs), thus the predictions give a measure of uncertainty and can handle multimodality. A nearest neighbor algorithm is applied on two different data structures. Tests to determine the accuracy and covariance consistency of both structures are performed on real data.**

I. Introduction

In order for an autonomous surface vessel (ASV) to operate safely it is essential that it is equipped with a robust collision avoidance (COLAV) system. A major component of this system is the prediction of the future positions of other vessels. A simple solution to this problem is to assume that nearby vessels will continue with constant velocity. However, such predictions might not be sufficient for longer prediction horizons. In recent years automatic identification system (AIS) data has become more available and has been used to improve the trajectory predictions of vessels.

In previous work, vessel trajectories have been predicted by describing a vessel's velocity using a Ornstein-Uhlenbeck (OU) stochastic process. This was shown to give better results than to assume a near constant velocity [1].

Another approach for vessel movement prediction is to cluster trajectories based on historical AIS data and then assign an object's initial state to one of these clusters [2], [3], [4]. This approach can typically be divided into four sequential steps:

1) Cluster trajectories based on historical data.
2) Classify a new object to one of the clusters found in step 1.
3) Generate a representative trajectory for each cluster.
4) Predict the movement of the object along the representative trajectory found in step 3.

The trajectory clustering (TRACLUS) algorithm presented in [5] is widely used for trajectory clustering. This algorithm uses density based clustering to detect trajectories. The algorithm partitions trajectories into smaller line segments and then clusters these line segments. The algorithm is further improved in [6] where it is made less sensitive to its decision parameters.

As opposed to TRACLUS, the traffic route extraction for anomaly detection (TREAD) algorithm [2] was specifically designed for AIS data predictions. This algorithm defines waypoints that later are clustered instead of clustering trajectories directly. The waypoints are defined when a vessel enters or exits a pre-defined bounding box with minimal movement, as well as when it stays within the box for a certain amount of time. The algorithm is principally developed to detect low-likelihood anomalies that deviate from the main trajectories.

Clustering based methods like TRACLUS and TREAD, as well as the OU method from [1], were developed for much larger prediction horizons than what is of interest in a COLAV system. In addition, the OU method requires the estimation of process parameters, which may not be straightforward. Therefore, a radically different and more data-driven approach was proposed in [7]. The key concept in this approach was a single point neighbor search (SPNS). Given an AIS message, the SPNS algorithm considers historical messages within a given radius, called close neighbors, to predict a course and speed of the vessel. Historical messages with courses that deviate by a certain amount from the vessel's course are discarded in order to avoid influence from opposite moving vessels in the predictions. What remains are messages that have a similar position and course as that of the vessel. The median course and speed of these close neighbors are calculated and used as a predicted course and speed of the vessel. The predicted course and speed is then used to calculate the future position by a given step length parameter. The same process is then applied on the newly predicted state and this is then repeated until a trajectory of desired length is produced. The output of the SPNS algorithm is an array of waypoints with equal distances between the positions of any two subsequent messages.

The inability to estimate prediction uncertainty and the inability to handle branching of sea lanes are two of the SPNS algorithm's main shortcomings. The neighbor course distribution method (NCDM) was developed in the MSc thesis [8] to account for this. Whereas the output from the SPNS can be seen as a list of states which forms a single trajectory, the output from NCDM is a tree of states which forms several trajectories. Each individual trajectory is calculated in a similar manner as in the SPNS. The same set of close neighbors are used to predict the vessel's course and speed at each predicted position. However, while the SPNS predict the course and speed as the median course and speed of the closest neighbors, the NCDM samples course from the distribution of the neighbors' courses. The NCDM is thus able to predict trajectories in several branched sea lanes and it possess the ability to indicate an uncertainty measure of the predictions.

This paper builds upon the work done in [7] and [8]. We extend the NCDM by introducing a Gaussian Mixture Model (GMM) to represent the position of a vessel. The NCDM is implemented using both the original data structure that was used in [8] and a new data structure proposed in this paper, and the two implementations are tested on real AIS data.

The outline of this paper is as follows: Section II introduces the NCDM using both data structures. In Section III the methods are tested on a comprehensive set of AIS data from Trondheimsfjorden, Norway. Lastly, conclusions and suggestions for further work are given in Section IV and Section V.

II. Neighbor Course Distribution Method

This section describes the NCDM introduced in [8]. First the prediction tree is explained. Then the two different data structures are introduced: the AIS message structure from [8] and a new structure which utilizes the recent past trajectory of the vessel.

A. Prediction tree

The NCDM takes an initial state $\mathbf{X}_1$ of a ship as input. This initial state is the root node of the prediction tree, where each node represents a predicted state $\hat{\mathbf{X}}_{k,j}$. The tree has a depth index k and a width index j. $N_{k,j}$ gives the number of child nodes for node (k, j), while J_k is the number of nodes at level k. In this paper we choose the number of children for the root node to be $N_{1,1} = J^{max}$, while the number of children at all other levels are set to one. In other words the tree has J^{max} branches which all originate in its root node. A predicted position $\hat{\mathbf{p}}_{k,j}$ is calculated from a random sample at every node in the tree. The method for obtaining the predicted position varies depending on the data structure used. The prediction

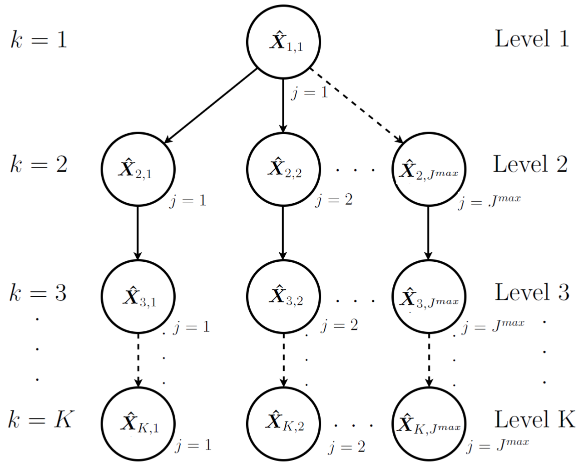


Fig. 1: Prediction tree structure. $N_{1,1} = J^{max}$, otherwise $N_{k,j} = 1$.

tree is illustrated in Figure 1.

Algorithm 1 Neighbor Course Distribution Method

```
Input parameters:
  • $\mathbf{X}_1$                          ▷ Initial state
  • $N_{k,j}$          ▷ Number of child nodes from node (k,j)
  • $K$                         ▷ Total number of tree levels
Set $\hat{\mathbf{X}}_{1,1} = \mathbf{X}_1$
for $k = 1$ to $K - 1$ do
    $q = 0$                     ▷ Indexing variable at level $k$
    for $j = 1$ to $J_k$ do
        Find close neighbors
        for $N_{k,j}$ iterations do
            q = q + 1
            Obtain random sample
            Calculate the next position
            Update $\hat{\mathbf{X}}_{k+1,q}$ based on the latest prediction
        end for
    end for
end for
```

B. Old data structure

The first structure, referred to as $\mathbf{X}^A$, is used in [7] and [8]. It consists of a list of AIS messages:

$$\mathbf{X}^A = \begin{bmatrix}\mathbf{X}_1 & \mathbf{X}_2 & \dots & \mathbf{X}_n\end{bmatrix}^T, \quad (1)$$

where each message is given as

$$\mathbf{X}_i = \begin{bmatrix}\text{MMSI}_i & t_i & \mathbf{p}_i & \chi_i & v_i\end{bmatrix}, \quad (2)$$

where MMSI_i, t_i, $\mathbf{p}_i$, χ_i and v_i are the unique vessel identification number, time stamp, position vector, course over ground (COG) and speed over ground (SOG), respectively. The position vector can be written as $\mathbf{p}_i = \begin{bmatrix}\lambda_i & \phi_i\end{bmatrix}$ where λ_i and ϕ_i are the longitude and latitude in the WGS84 coordinate system.

The set of close neighbors (CNs) at step (k, j) is defined as

$$\mathbf{C}_{k,j} = \{\mathbf{X}_i | \, d(\hat{\mathbf{p}}_{k,j}, \mathbf{p}_i) \leq r_c, \chi_i \in I, \mathbf{X}_i \in \mathbf{X}\}, \tag{3}$$

where $d(\hat{\mathbf{p}}_{k,j}, \mathbf{p}_i)$ is the Euclidean distance between the predicted position and a position in the dataset calculated with the Haversine formula [9]. Further, r_c is the search radius and I is the interval of accepted course angles given by

$$I = \left[\chi_{k,j} - \Delta\chi, \quad \chi_{k,j} + \Delta\chi\right], \tag{4}$$

where $\Delta\chi > 0$ is the maximum course angle deviation. In other words a state is considered a close neighbor to the predicted state if its position is within a radius r_c from the predicted position and if its course is within $\Delta\chi$ of the last predicted course.

A random sample is drawn from the CNs and the predicted course and speed are obtained from this sample. These values are used to update the state $\hat{\mathbf{X}}_{k+1,q}$. The position of the next state is given by:

$$\hat{\mathbf{p}}_{k+1,q} = \hat{\mathbf{p}}_{k,j} + \Delta l \left[\sin(\hat{\chi}_{k,j}) \quad \cos(\hat{\chi}_{k,j})\right], \tag{5}$$

where $\hat{\mathbf{p}}_{k,j}$ is the ship's predicted position at node (k, j), $\hat{\chi}_{k,j}$ is the predicted course at node (k, j) and Δl is the fixed step parameter. The time stamp is updated using the following equation:

$$\hat{t}_{k+1,q} = \hat{t}_{k,j} + \frac{\Delta l}{\hat{v}_{k,j}}. \tag{6}$$

This means that the new state is given as

$$\hat{\mathbf{X}}_{k+1,q} = \left[\mathrm{MMSI}_i \quad \hat{t}_{k+1,q} \quad \hat{\mathbf{p}}_{k+1,q} \quad \hat{\chi}_{k,j} \quad \hat{v}_{k,j}\right], \tag{7}$$

where $\hat{\chi}_{k,j}$ and $\hat{v}_{k,j}$ are the randomly drawn course and speed respectively.

C. New data structure

The second structure, referred to as $\mathbf{X}^B$, structures the data as a list of sub-trajectories. A sub-trajectory $\mathbf{S}$ consists of n points

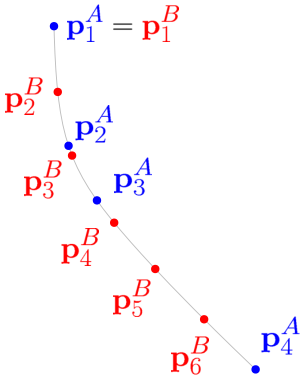
$$\mathbf{S} = \left[\mathbf{p}_1 \quad \mathbf{p}_2 \quad \cdots \quad \mathbf{p}_n\right], \tag{8}$$

where $\mathbf{p}$ is a point given by longitude and latitude. There is an equal amount of time elapsed between any two subsequent points. These points are found using cubic spline interpolation on the points given in the AIS messages. A trajectory is given by:

$$\mathbf{T} = \left[\mathbf{S}_1 \quad \mathbf{S}_2 \quad \cdots \quad \mathbf{S}_M\right]^T, \tag{9}$$

and the entire dataset is given by a list of trajectories:

$$\mathbf{X}^B = \left[\mathbf{T}_1 \quad \mathbf{T}_2 \quad \cdots \quad \mathbf{T}_N\right]^T. \tag{10}$$



$$\mathbf{X}^A = \begin{bmatrix} \mathrm{MMSI} & t_1 & \mathbf{p}_1^A & \chi_1 & v_1 \\ \mathrm{MMSI} & t_2 & \mathbf{p}_2^A & \chi_2 & v_2 \\ \mathrm{MMSI} & t_3 & \mathbf{p}_3^A & \chi_3 & v_3 \\ \mathrm{MMSI} & t_4 & \mathbf{p}_4^A & \chi_4 & v_4 \end{bmatrix}$$

$$\mathbf{X}^B = \begin{bmatrix} \mathbf{p}_1^B & \mathbf{p}_2^B & \mathbf{p}_3^B \\ \mathbf{p}_2^B & \mathbf{p}_3^B & \mathbf{p}_4^B \\ \mathbf{p}_3^B & \mathbf{p}_4^B & \mathbf{p}_5^B \\ \mathbf{p}_4^B & \mathbf{p}_5^B & \mathbf{p}_6^B \end{bmatrix}$$

Fig. 2: An example of a short trajectory represented with both data structures: The blue dots represents the ship's positions as given in the original AIS data. The red dots are the new data points obtained using interpolation. Here the sub-trajectories consist of $n = 3$ points.

A comparison between the old and the new structure can be seen in Figure 2. Reformatting the dataset from the original structure $\mathbf{X}^A$ to the new structure $\mathbf{X}^B$ involves three main steps:

1) Find trajectories from the original dataset: Points that belong to the same MMSI number and have less than 15 minutes between two subsequent points are considered as part of the same trajectory. The time limit makes sure that trajectories from vessels which leave and later enter the dataset window or which stay in port for a long time, are split into separate trajectories

2) Interpolate trajectories to get new data points: Cubic spline interpolation is used to extract new data points at a specified time interval. The new data points now form a new trajectory.

3) Create sub-trajectories of n points: The first sub-trajectory consists of the first n points of the new data points. This is the first row of the data structure. The next row is a new sub-trajectory shifted one point forward, thus creating a structure of partly overlapping sub-trajectories. This continues until the end of the trajectory. With this method a trajectory of k points will result in $k - n + 1$ sub-trajectories of n points.

Sub-trajectories can be seen as $2 \times n$ dimensional points, where n is the number of points of the sub-trajectories in $\mathbf{X}^B$. The distance between sub-trajectories is thus the Euclidean distance between these points. Close neighbors are found as in (3) where $d(\hat{\mathbf{p}}_{k,j}, \mathbf{p}_i)$ now is replaced by the distance between the two sub-trajectories. A key difference now is that the course tolerance parameter $\Delta\chi$ is no longer needed. Instead of choosing an arbitrary parameter for accepted courses, sub-trajectories with similar courses to the initial state are selected because these sub-trajectories

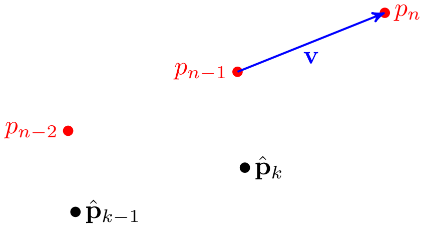


Fig. 3: The sub-trajectory shown in red is a close neighbor of $\mathbf{X}_1$ if the Euclidean distance between $\begin{bmatrix}\hat{\mathbf{p}}_{k-1} & \hat{\mathbf{p}}_k\end{bmatrix}$ and $\begin{bmatrix}\mathbf{p}_{n-2} & \mathbf{p}_{n-1}\end{bmatrix}$ is less than r_c. The location of $\hat{\mathbf{p}}_{k+1}$ is determined by adding $\mathbf{v}$ to $\hat{\mathbf{p}}_k$.

are closer to the initial state. Two nearby sub-trajectories that point in opposite directions will still have a large distance between each other while two sub-trajectories that point in roughly the same direction will have a small distance between each other. Sub-trajectories with similar courses will thus be considered close neighbors. The step length parameter Δl is also no longer needed as predictions are made with fixed time steps as defined by the data structure.

The input in this case is a sub-trajectory of $n-1$ points where the last point will be the first of the predicted trajectory. This state is compared to the $n-1$ first columns of $\mathbf{X}^B$ to find close neighbors. A random sample (a sub-trajectory from $\mathbf{X}^B$) is drawn from the CNs. From this sub-trajectory a vector $\mathbf{v} = \mathbf{p}_n - \mathbf{p}_{n-1}$ is obtained. As shown in Figure 3, this vector is added to $\hat{\mathbf{p}}_{k,j}$ to obtain $\hat{\mathbf{p}}_{k+1,q}$. The state, which is a sub-trajectory, is updated by removing the first point from the sub-trajectory and adding the newly calculated one at its end.

An advantage with the new data structure is that vessels with similar speeds are more likely to be considered close neighbors. This is because sub-trajectories of points are compared instead of just single points. Intuitively, sub-trajectories of similar length will be considered closer than sub-trajectories of different length.

D. *Gaussian mixture representation*

For both structures the predicted future position of a vessel is given by a number of J^{max} points taken from the desired level of the prediction tree. A GMM is then fitted to these points to give a probabilistic model of the future position. This is done using the Expectation Maximization (EM) algorithm which will fit the maximum likelihood GMM for the given points. The number of components used are increased until the means of two components are less than a pre-specified distance apart, then one less component is used. It is also possible to consider a model selection criterion such as AIC [10] or BIC [11] instead of looking at the distance between the means, but in the context of vessel trajectories we are mostly interested in a multimodal distribution to enable predictions in branching sea lanes.

III. Tests and Results

The NCDM is tested on AIS data gathered during 2015 in Trondheimsfjorden, Norway.

A. *Test setup*

The dataset is initially divided into a training set, from where the and a test set. The first 90% of the data points are used as training data, while the remaining 10% are used for tests. Both structures are tested to predict vessel positions 5, 10 and 15 minutes into the future from an initial state. We use the same prediction tree as described in Section II-A with $J^{max} = 200$, i.e., the tree has 200 branches which all originate in its root node. All tests were done on $N = 400$ initial states randomly sampled from the test set without replacement. The close neighbors of this sample are obtained from the training set. A test of a given method for a given time horizon on N initial states will from here be referred to as a 'test' while a test of a single initial state will be referred to as an 'individual test'.

The initial states and their corresponding trajectories must fulfill one requirement: The time at the end of the trajectory minus the time of the initial state must be larger or equal to the test time. This ensures that there is a true position in the test trajectory that the prediction can be compared against.

An individual test will be discarded if more than 25% of the predicted positions at the desired level of the prediction tree are identical. This is an indication that there is not enough data in the area of the initial state for the algorithm to make any reasonable predictions, and it is also difficult to fit a GMM to the points in this case.

Table I shows the decision parameters used for the test of the NCDM using data structure $\mathbf{X}^A$. The trajectories produced using this structure have a fixed step length between subsequent states and each state has a predicted time stamp. Most likely they will not have time stamps equal to the test time (5, 10 or 15 minutes). It is therefore necessary to interpolate the trajectories to obtain a position at the desired time.

TABLE I: Decision parameters for NCDM with $\mathbf{X}^A$

Decision parameter	Value	Description
r_c	100m	Search radius for CNs
Δl	100m	Prediction step length
$\Delta\chi$	35°	Maximum course deviation for CNs

Data structure $\mathbf{X}^B$ was tested using the decision parameters in Table II. The search radius r_c is set to the same as for the first data structure. Parameters n and t are chosen when generating $\mathbf{X}^B$ as described in Section II-C. The choices of n and t are somewhat arbitrary. The minimum value for n is three and was chosen for simplicity. Higher values of n were tried with no visible improvement, but this has not been extensively tested. The average SOG for the messages in the dataset is roughly 5.4m/s. If similar step length as with $\mathbf{X}^A$ was to be chosen the time step would be $t = (100/5.4)s \approx 18.5s$. This was tried, but with such small sub-trajectories it was found that sub-trajectories pointing in opposite directions often were considered close neighbors. Therefore, t was increased to 60 seconds.

TABLE II: Decision parameters for NCDM with $\mathbf{X}^B$

Decision parameter	Value	Description
r_c	100m	Search radius for CNs
n	3	Number of points in each sub-trajectory
t	60s	Time between each point

B. *Performance measures*

Two performance measures are used to determine the quality of a set of predictions: the root mean square error (RMSE) and a generalized concept of filter consistency, also sometimes referred to as credibility.

The RMSE is used to measure the accuracy of a prediction. The lower the RMSE, the better is the prediction. We define the RMSE as

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{J^{max}} \|\mathbf{p} - \hat{\mathbf{p}}_i\|^2}{J^{max}}}, \tag{11}$$

where $\mathbf{p}$ is the true position, $\hat{\mathbf{p}}_i$ is the predicted position at iteration i and J^{max} is the number of individual predictions as defined in Section II-A. This measure gives an idea of how close the mean of the predictions is to the true value. The RMSE is generally not very well suited for multimodal distributions. The mean of a multimodal distribution might be in an empty area between components. This is, however, a rare scenario.

The second performance measure is used to determine if the GMM produced is consistent. Filter consistency is colloquially explained as follows in [12]: The estimation error should have magnitude commensurate with the corresponding covariance that is yielded by the estimator. In order to deal with multimodality, consistency of the prediction methods is measured using the same method as in [13]. First, the Probability Density Function (PDF) value as given by the GMM for the true position is noted for each individual test. This value, called f, is then compared to the maximum value of the same PDF, called f_{max}. The ratio $f/f_{max} \in [0, 1]$ thus serves as a quality measure where a single prediction is better the closer the ratio is to one. The distribution of f/f_{max} for N predictions will then give an idea of the consistency. The values f and f_{max} are illustrated in Figure 4.

C. *Results*

The median values of the RMSE of all the methods for each test time is given in Table III. The same information is plotted in Figure 5. As can be seen, the accuracy of the NCDM is roughly the same using both data structures. It can seem as if $\mathbf{X}^B$ has a slight advantage for short prediction horizons while $\mathbf{X}^A$ is more accurate for longer horizons. However, the sample size might be too small to draw any definitive conclusions.

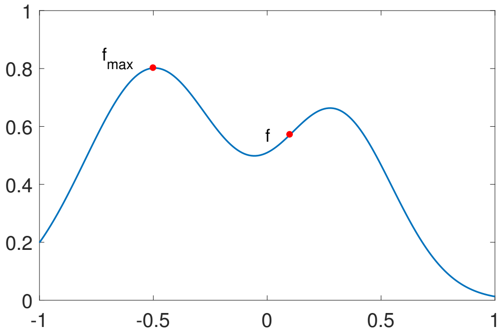


Fig. 4: Performance measure illustrated using one-dimensional data. The true value is $x = 0.1$. In this case $f/f_{max} \approx 0.57/0.8 \approx 0.71$.

TABLE III: Median RMSE for each test time

	NCDM with $\mathbf{X}^A$	NCDM with $\mathbf{X}^B$
5 min	282m	269m
10 min	509m	537m
15 min	661m	758m

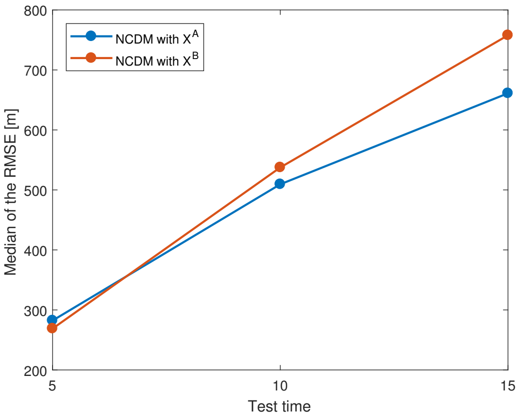
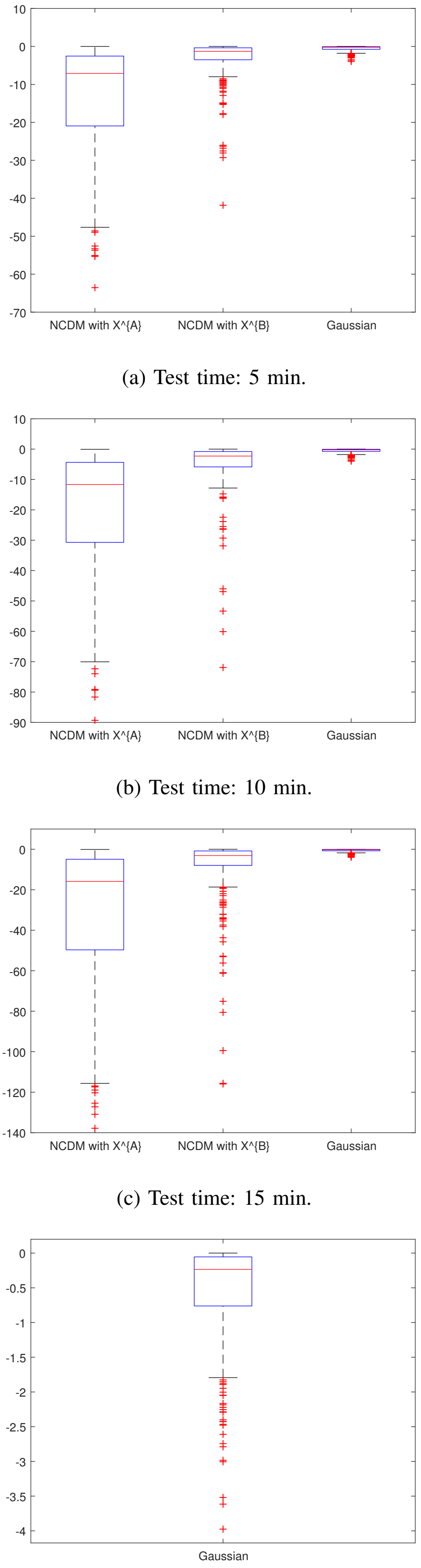


Fig. 5: Median RMSE over time

The natural logarithms of the f/f_{max} ratios are displayed in Figure 6 in order to investigate the consistency. The far-right box plots are values sampled from a Gaussian distribution (also shown in Figure 6d). This plot should therefore exhibit ideal consistency properties and will serve as a comparison for the box plots to the left. A consistent prediction method would have a box plot similar to the one from the Gaussian distribution.



(a) Test time: 5 min.

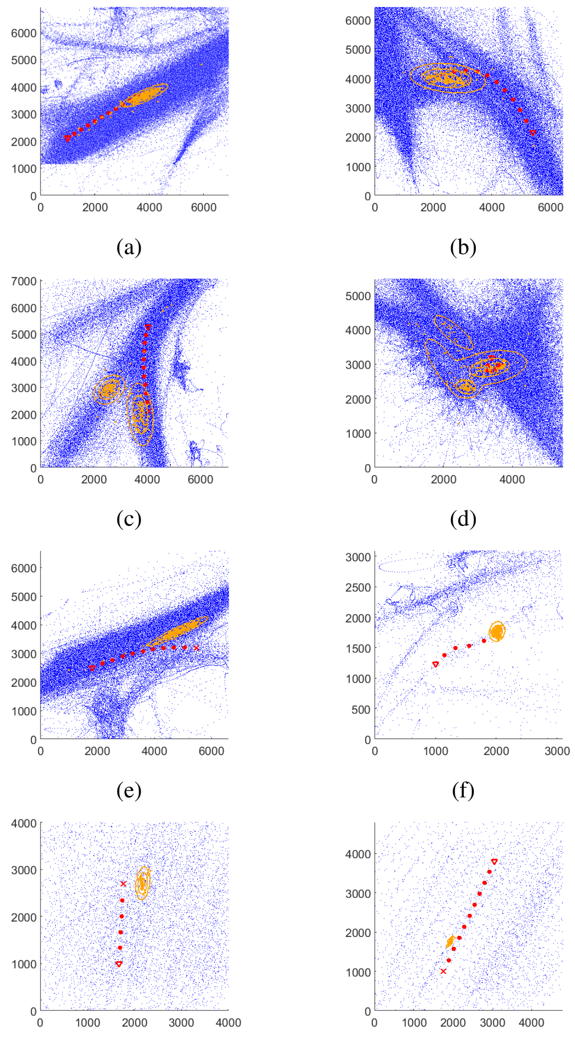
(b) Test time: 10 min.

(c) Test time: 15 min.

(d) Close-up of Gaussian

Fig. 6: Box plot of the PDF-ratios

The consistency behavior deteriorates as the test time increases across all methods (notice the change of y-axis). It can be seen that NCDM using $\mathbf{X}^B$ over $\mathbf{X}^A$ results in a significant improvement in the consistency. However, compared to the Gaussian it is evident that the consistency properties of $\mathbf{X}^B$ still remains far from ideal.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

(h)

Fig. 7: The blue dots are reported ship positions from the AIS dataset. The red dots indicate the true trajectory of a ship with the triangle representing the initial position and the cross the true position at the time the position is predicted. The orange dots are predicted positions and the orange lines correspond to the 1, 2 and 3 standard deviation equi-probability contours of the GMM fitted to them. The axis are in meters.

Figure 7 shows plots where the true trajectories of vessels are compared to their predicted positions produced by the NCDM. They are not intended to show a representative selection of predictions or to give a comparison of the two structures, but are chosen to highlight the strengths and weaknesses of the NCDM in general.

Figure 7a and Figure 7b show two typical outcomes where relatively accurate predictions are made. Figure 7c and Figure 7d show cases where the algorithm has produced multimodal predictions, Figure 7c in particular gives a good example of how the algorithm is able to handle branching. However, Figure 7e shows an example where the algorithm fails to produce a component for the less traveled lane. Figure 7f, Figure 7g and Figure 7h show predictions made in areas with sparse data density. The first produces a good prediction, while the last two highlight a major weakness of the algorithm. In areas with sparse data density it often makes overconfident and inaccurate predictions. The plots also show that there is usually more uncertainty in the speed prediction than in the prediction of the course (best illustrated in Figure 7a). The same conclusion was also reached in [8].

IV. Conclusion

The NCDM is able to give a probabilistic position prediction of vessels. The predicted position distribution can be multimodal and the algorithm is thus able to handle branching. A new data structure was developed to use data from the recent trajectories of vessels in the predictions. The accuracy of the predictions produced using the NCDM with the new structure are similar to the ones using the old structure. However, the uncertainty evaluations of the predictions are significantly more reliable with the new structure. There is still a clear potential for improvements in consistency, although it should be noted that the algorithm is intended to be used in a highly proactive manner, i.e., only for suggesting how other ships possibly may move 5-15 minutes into the future. This may relax the requirements to consistency somewhat and future research is needed to determine what is acceptable.

V. Suggestions for future work

The performance of the NCDM in areas with low data density has to be improved. A way of compensating for the lack of data may be to include the possibility that the vessel moves straight ahead with constant speed in such areas. Furthermore, the current method for determining the number of components for the EM algorithm is not ideal and may be improved by using automatic model selection [14]. The next step is to then assess the suitability of the NCDM as part of a COLAV system [15]. Another approach for trajectory prediction using Guassian Process Regression can also be investigated.

Acknowledgment

This work was supported by the Research Council of Norway through projects 223254 (Centre for Autonomous Marine Operations and Systems at NTNU) and 244116/O70 (Sensor Fusion and Collision Avoidance for Autonomous Marine Vehicles). The authors would like to express great gratitude to DNV GL, especially Øystein Engelhardtsen and Hans Anton Tvete, for providing real-world AIS data. The work of the last author is funded by DNV GL.

References

[1] Leonardo M Millefiori et al. “Modeling vessel kinematics using a stochastic mean-reverting process for long-term prediction”. In: *IEEE Transactions on Aerospace and Electronic Systems* 52.5 (2016), pp. 2313–2330.

[2] Giuliana Pallotta, Michele Vespe, and Karna Bryan. “Traffic Knowledge Discovery from AIS Data”. In: *16th International Conference on Information Fusion (Fusion)*. IEEE. Istanbul, Turkey, 2013.

[3] Fabio Mazzarella, Michele Vespe, and Carlos Santamaria. “SAR Ship Detection and Self-Reporting Data Fusion Based on Traffic Knowledge”. In: *IEEE Geoscience and Remote Sensing Letters* vol. 12, no. 8 (Aug. 2015), pp. 1685–1689.

[4] Fabio Mazzarella, Virginia Fernandez Arguedas, and Michele Vespe. “Knowledge-based vessel position prediction using historical AIS data”. In: *2015 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, Oct. 2015.

[5] Jae-Gil Lee, Jiawei Han, and Kyu-Young Whang. “Trajectory clustering”. In: *Proceedings of the 2007 ACM SIGMOD international conference on Management of data - SIGMOD ’07*. Association for Computing Machinery (ACM), 2007.

[6] Chen Jiashun. “A new trajectory clustering algorithm based on TRACLUS”. In: *Proceedings of 2nd International Conference on Computer Science and Network Technology*. IEEE.

[7] Simen Hexeberg et al. “AIS-based vessel trajectory prediction”. In: *20th International Conference on Information Fusion (Fusion)*. IEEE. Xi’an, China, 2017.

[8] Simen Hexeberg. “AIS-based Vessel Trajectory Prediction for ASV Collision Avoidance”. MA thesis. Norwegian University of Science and Technology, 2017.

[9] Naomi S Altman. “An introduction to kernel and nearest-neighbor nonparametric regression”. In: *The American Statistician* 46.3 (1992), pp. 175–185.

[10] Hirotugu Akaike. “A new look at the statistical model identification”. In: *IEEE transactions on automatic control* 19.6 (1974), pp. 716–723.

[11] Gideon Schwarz et al. “Estimating the dimension of a model”. In: *The annals of statistics* 6.2 (1978), pp. 461–464.

[12] Yaakov Bar-Shalom and Xiao-Rong Li. *Multitarget-multisensor tracking: principles and techniques*. Vol. 19. YBs London, UK: 1995.

[13] Edmund Brekke and Mandar Chitre. “Bayesian multi-hypothesis scan matching”. In: *OCEANS*. IEEE. Bergen, Norway, 2013.

[14] Lei Shi, Shikui Tu, and Lei Xu. “Learning Gaussian mixture with automatic model selection: A comparative study on three Bayesian related approaches”. In: *Frontiers of Electrical and Electronic Engineering in China* 6.2 (2011), pp. 215–244.

[15] Bjørn-Olav Holtung Eriksen and Morten Breivik. "MPC-based Mid-level Collision Avoidance for ASVs using Nonlinear Programming". In: *Proc. of IEEE CCTA - to appear*. Kohala Coast, Hawai'i, 2017.