Improved IMM algorithm based on XGBoost

Da Li*, Pei Zhang and Ruifang Li

School of Information Engineering, Wuhan University of Technology, Wuhan, 430070, Hubei, China

frankli26@whut.edu.cn

Abstract. Based on Markov hypothesis, IMM uses multiple motion models to match the moving states of the target, and assumes the transfer probability of each model according to the prior knowledge, which has strong adaptability to tracking maneuvering target. However, it is not direct enough to obtain prior knowledge from source data statistics and then make decision according to maximum likelihood, and the information of source data is not fully utilized. Therefore, we use XGBoost in machine learning algorithm to replace this process. We propose XGBoost-IMM model algorithm. XGBoost can fully learn the information of the source data and make decision on the target motion model, and then IMM can perform multi-filter filtering based on the decision. Experimental results show that our algorithm has good performance.

1. Introduction

Maneuvering target tracking needs to use a suitable target tracking model. Since only one model is difficult to match target trajectory, IMM algorithm[1-3] uses multiple motion models to match. According to the prior knowledge, it assumes the transfer probability between the models, which has strong adaptability to tracking maneuvering targets, but it is not direct enough and does not make full use of the information of the source data.

Machine learning algorithm can fully learn the information in the source data and generalize it into features. The target motion model matching problem can also be regarded as a multi-label classification task in machine learning field. Boosting tree is an efficient and widely used machine learning algorithm, and XGBoost[4] algorithm is the most widely used entity of boosting tree strategy. XGBoost is widely used in chemistry such as Wu Z et al.[5], biology as Aibar S et al.[6], and computer science such as Chen M et al.[7], and has made remarkable achievements. We now use XGBoost to improve the traditional IMM and propose XGBoost-IMM algorithm. The simulation results show that our algorithm can identify the target model more accurately and get smaller estimation error.

The structure of the paper is as follows: Section 2 introduces the related work. The XGBoost-IMM model is introduced in Section 3. Section 4 is the implementation and performance test of our algorithm, and we have summarized our work in section 5.

2. Related Work

With the improvement of computer computing ability, machine learning algorithm has been further developed. It can extract more effective features and process more complex data. It is an irresistible trend to replace traditional statistical methods with learning methods. Deep learning is also a machine learning algorithm, but it is different from the orthodox machine learning algorithm because of its poor interpretability.
K. Thormann et al.[8] used the random forest algorithm in machine learning to track the target, and achieved good results in the low noise scene. However, since the measured values are directly used as input features, when the tracks are randomly initialized, the estimation accuracy of their algorithm is not enough. Bowen Zhai et al.[9] improved on the basis of this, and utilized the centralized strategy to extract the relevant kinematic information from the random trajectory input. However, because it is also an end-to-end system, its stability is not high. M. Sun et al.[10] used Elman neural network to assist the traditional IMM Kalman filter in tracking maneuvering target, which is better than IMM in tracking trajectory and root mean square error, but its interpretability is not high. Based on the dependence of input trajectories, Lichuan Deng et al.[11] proposed LSTM-IMM algorithm. However, like K. Thormann's algorithm, it is not suitable for random input scenarios. Zhu h et al.[12] proposed NACT-IMM algorithm for CT model, which has excellent results for specific CT input.

3. XGBoost-IMM algorithm

The IMM algorithm first sets the initial model probability \( u_0 \) and constant model transition probability \( P \) through historical data and experience, then calculates the mixed estimation value through the posterior probability, and then filters the mixed estimation value, judges and updates the model probability according to the likelihood function. Finally, the above steps are repeated to get the final result. It can be seen that IMM algorithm is not direct enough and does not make full use of the information of source data. Therefore, we use XGBoost proposed by Tianqi Chen to improve the traditional IMM and propose XGBoost-IMM algorithm. The structure of our algorithm is shown in Figure 1.

![Figure 1. The structure of XGBoost-IMM.](image)

**Step 1:** Preprocessing. A sliding window of size \( w \) should be maintained. When the system input at time \( k \) is \( x_k \), the first vector in the window is moved out, and the rest of the vectors are moved forward, and \( x_k \) will be the last vector in the window. \( \{x_k^{w+1}, x_k^{w+2}, \ldots, x_k^w\} \) will be gotten. Use this window as the input of XGBoost System at time \( k \).

**Step 2:** Soft judgment. For the set of set tracking models, XGBoost System makes soft decision and outputs the probability of the corresponding model. Assuming that the number of tracking models is \( n \), we will get \( \{p_1^k, p_2^k, \ldots, p_n^k\} \).

**Step 3:** Filtering. Use \( n \) interactive filters running in parallel in the IMM algorithm to filter, and then use the probability obtained in step 2 to weight to obtain the weighted filtering result:
Step 4: Repeat the steps from step one to step three until the complete filtering result is obtained.

4. Experiment and simulation results

4.1. Training XGBoost

XGBoost belongs to the boosted tree model, and the model of the boosted tree algorithm is as follows:

\[
x_k^o = p_k^1 \cdot x_k^0 + p_k^2 \cdot x_k^0 + \cdots + p_k^n \cdot x_k^n
\]

(1)

Where \( x_k \) represents the sample, \( x_k^0 \) represents the output value of the sample \( i \), \( K \) represents the number of trees (base learners), \( f_k \) represents the \( k\)th tree, and \( F \) represents the tree space.

The loss function is as follows:

\[
L(\phi) = \sum_i l(\hat{y}_i, y_i)
\]

(2)

Where \( y_i \) is the label of the \( i\)th sample, and \( l \) is the defined loss function (differentiable convex function) of a single sample.

We set the input \( x_i \) as a fixed-length sampling point coordinate vector, the label \( y_i \) is the label code of different tracking models, the value of \( K \) is 200, and the loss function is an exponential loss function:

\[
l(y_i, \hat{y}_i) = e^{-y_i \hat{y}_i}
\]

(4)

Figure 2. The average probability distribution of the motion models of the training samples.

There are 4 kinds of motion models in the initial setting tracking model set, which are CV, CA, CT and Singer. Figure 2 is the average probability distribution of four motion models of all training samples. We clearly see XGBoost doing well in the training set. In order to prevent over fitting, we set a smaller shrinkage value and a smaller maximum number of leaf nodes.
Figure 3. The average probability distribution of the motion models of the testing samples. Figure 3 is the result of XGBoost on the test set, which is basically the same as the result of the training set. It can be seen that the three models of CV, CA, and CT have a relatively high degree of discrimination. CV and Singer or CA and Singer all have a certain degree of similarity. Because Singer is a higher-order model, it can be reduced to CV or CA.

Figure 4. The average probability distribution of three motion models of the testing samples. Figure 4 shows when the Singer model is removed, a more accurate probability distribution can be obtained, but the RMSE of the whole XGBoost-IMM algorithm will increase due to the reduction of the number of models. Figure 5 shows When a higher-order jerk model is added, the probability distribution is more stable. Although the final RMSE may be reduced, the calculation cost is increased.
4.2. XGBoost-IMM algorithm simulation

Assuming that the target detection sensor selects passive radar, the target observation model is a point target measurement model, and the radial distance $r$ and azimuth angle $\theta$ of the target can be obtained. Set the noise to Gaussian white noise, and the related parameters are set as follows: $\sigma_r = 50m$, $\sigma_\theta = 0.1^\circ$. The sampling interval is $T_s = 1s$, the sampling points are $N_s = 800$. Let's make the target move in a straight line at a constant speed in the first 200 seconds, make a uniform circular motion from 200 seconds to 400 seconds, then continue to move in a straight line at a constant speed from 400 seconds to 600 seconds and accelerate the target with the acceleration $a$ in the last 200 seconds. The RMSE is used as the evaluation standard of the simulation to evaluate the performance of the filtering algorithm.

![Figure 6. Root mean square error of position estimation.](image)

Figure 6 shows the RMSE of XGBoost-IMM algorithm. It can be seen that XGBoost-IMM algorithm is relatively stable and RMSE is small at most sampling points. In the process of target motion model switching, just as the target changes from uniform linear motion to uniform circular motion in the 200th second, XGBoost-IMM algorithm makes the RMSE of position estimation stable within the window length.

5. Conclusion

Aiming at the problem that the IMM algorithm is not direct enough and the source data is not fully utilized, we propose an improved XGBoost-IMM algorithm. The algorithm makes full use of source data through machine learning algorithms, and can directly make tracking model matching decisions. Experimental results show that the algorithm has better recognition rate and filtering effect. In the future, we will try to merge the filtering process with the model matching process to achieve a stable end-to-end system.

References

[1] Li D, Zou X, Lou P, et al. Iterative Unbiased Conversion Measurement Kalman Filter with Interactive Multi-model Algorithm for Target Tracking[C]// Asian Simulation Conference. Springer, Singapore, 2018.
[2] SUSTIKA R, SURYANA J. Nonlinear-Filtering with Interacting Multiple-Model Algorithm for Coastal Radar Target Tracking System [J]. Applied Mechanics & Materials, 2015, 13(1): 211-20.

[3] Yoo K, Chun J, Shin J. Target tracking and classification for missile using interacting multiple model (IMM)[C]//2018 International Conference on Radar (RADAR). IEEE, 2018: 1-6

[4] Chen T , Guestrin C . XGBoost: A Scalable Tree Boosting System[J]. 2016.

[5] Wu Z , Ramsundar B , Feinberg E N , et al. MoleculeNet: A Benchmark for Molecular Machine Learning[J]. Chemical Science, 2017, 9(2).

[6] Aibar S , Carmen Bravo González-Blas, Moerman T , et al. SCENIC: single-cell regulatory network inference and clustering[J]. Nature Methods, 2017.

[7] Chen M , Tian Y , Chang K W , et al. Co-training Embeddings of Knowledge Graphs and Entity Descriptions for Cross-lingual Entity Alignment[J]. 2018.

[8] K. Thormann, F. Sigges and M. Baum, "Learning an object tracker with a random forest and simulated measurements", Int. Conf. on Information Fusion, pp. 1-4, July 2017.

[9] Zhai B , Yi W , Li M , et al. Data-driven XGBoost-based filter for target tracking[J]. The Journal of Engineering, 2019, 2019(20):6683-6687.

[10] M. Sun, Z. Ma and Y. Li, "Maneuvering target tracking using IMM Kalman filter aided by Elman neural network", Int. Conf. on Intelligent Human-Machine Systems and Cybernetics, pp. 144-148, August 2015.

[11] L. Deng, D. Li, and R. Li, "Improved IMM Algorithm based on RNNs," Journal of Physics: Conference Series, vol. 1518, p. 012055, 2020/04 2020.

[12] Zhu H, Xiong W, Cui Y, et al. Research on IMM Algorithm of New Adaptive Turn Model[J]. Computer Engineering and Applications, 2019,55(17): 252-258.