Learning to Multi-Target Tracking in Dense Clutter Environment with JPDA-Recurrent Neural Networks

Hui Zhang¹, Hua-jun Liu¹ and Cai-ling Wang²

¹ School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China;
² College of Automation, Nanjing University of Posts and Telecommunications, Nanjing 210023, China

Corresponding author: liuhj@njust.edu.cn

Abstract. Multi-target tracking in dense clutter environment is an open problem and still involves many challenges. Different from the classical tracking methods involving complex models and accurate prior knowledge, deep learning methods have been researched for this problem in recent years. In this paper, a novel JPDA-recurrent neural networks (RNNs) based tracking approach is proposed. Three stages which can address data association, state prediction and track management are included in our approach. For the prediction stage, a model for modeling temporal sequence based on long-short term memory (LSTM) is employed to learn the targets motion parameters from radar noisy measurements. Moreover, an RNN-based track existence probability model is proposed to assess the track quality and to automatically initialize, maintain and terminate the tracks. This approach is demonstrated over a simulation scenario, with promising results.

1. Introduction

The continuous and accurate identity maintenance and state estimation of all targets in a clutter environment is a critical requirement for successful multi-target tracking. One question is how to model the variable motions of target present in a sensor surveillance noisy data that include dense clutter, so that to estimate the position of targets. Another key task is to discriminant whether the measurements originate from targets, background or clutter, which called data association. Moreover, due to the clutter and the number of targets are not known and vary with time, a track management strategy is needed to initialize, maintain and termination track automatically. The online multi-target tracking can be viewed as a time series prediction, and data association in the complex environment.

The conventional multi-target tracking methods are based on Bayes filters [1], nearest neighbor data association (NNDA) and probability data association (PDA) [2]. However, these methods assume that mathematical models of target motion and clutter distributions are known in advance, which can deteriorate the tracking performance. In recent years, deep learning has revolutionized how to track multiple targets. In particular, the recurrent neural network (RNN) has shown outstanding performance. Milan et al. [3] presented a unified recurrent neural networks structure to the whole visual multi-target tracking tasks, shows promising results. However, RNN can only process information at a certain time intervals, is hard to predict target motion if the time interval is too long. Therefore, this paper extends the long short-term memory (LSTM) which can memorize and filter information at any time interval to the target motion prediction. In data association process, Milan et
al. [3] proposed an LSTM-based data association architecture which only utilizes the Euclidean distance between the targets and measurements just like the NNDA using only the nearest neighbor measurement, is weak in anti-interference ability and may lead to more association errors when used for radar tracking. PDA can barely track when multiple targets are in dense clutter environment. In contrast, joint probability data association (JPDA) is employed in this work for that it can better adapt to the multiple targets tracking in such environment.

The contributions of our work are as follows. First, a novel tracking architecture based on deep learning and JPDA (see Figure 1) is proposed to allow tracking multiple targets in dense clutter environment. Second, it is shown that the proposed models for the time-series prediction problem can be learned completely from data without any prior knowledge. In addition, to the best of our knowledge, this is one of first papers addressing the radar multi-target tracking in dense clutter environment with RNNs, the result on multi-target simulation scenario shows remarkable potential.

![Figure 1. Tracking architecture of our approach.](image)

2. Related work

One of the concerned strategies is how to associate each measurement with the corresponding target. NNDA and PDA [2] are used to address this problem traditionally. Previous works, such as the multiple hypothesis tracker (MHT) [4] and JPDA [5] are widely reused in novel works [6, 7] recently. Both still have enormous potential in solving data association problems. Therefore the JPDA is used in this paper to address the task of data association.

When comes to the challenging problems of tracking multiple targets, a lot of works have studied on these problems. It is worth noting that with the rise of deep learning, deep learning approaches have been successful in similar problems. For instance, the convolutional neural networks (CNNs) can be used for hand-written digit recognition [8], image classification [9], and so on. Although CNNs has achieved the most advanced results in many fields, but are subject to the output format and not well suited for a sequential data. In our work, we consider a more effective deep learning approach to address the radar multi-target tracking problem.

Early studies have demonstrated the potential of RNNs for state estimation and tracking problem [10]. RNNs can store and exploit the past information through feedback loops, which enable it to learn temporal kinematic behavior of time series [11]. Recently, RNNs and its variant LSTMs with a structure of gate show the outstanding performance for time series tasks, such as connected handwriting recognition [12]. Many attempts were carried on in order to solve the tracking problem. Alahi et al. [13] presented an LSTM-based model to predict human trajectories, but the human movement follows common sense rules and social customs, their method it is not suitable for the unconstrained environment. A framework for deep tracking was presented in [14], which can predict the state avoiding any hand-crafting operations. Then, an entirely RNNs structure is proposed to accomplish all the multi-target tracking tasks [3], which has made a significant step for online multi-

```latex
\textit{Figure 1. Tracking architecture of our approach.}

2. Related work

One of the concerned strategies is how to associate each measurement with the corresponding target. NNDA and PDA [2] are used to address this problem traditionally. Previous works, such as the multiple hypothesis tracker (MHT) [4] and JPDA [5] are widely reused in novel works [6, 7] recently. Both still have enormous potential in solving data association problems. Therefore the JPDA is used in this paper to address the task of data association.

When comes to the challenging problems of tracking multiple targets, a lot of works have studied on these problems. It is worth noting that with the rise of deep learning, deep learning approaches have been successful in similar problems. For instance, the convolutional neural networks (CNNs) can be used for hand-written digit recognition [8], image classification [9], and so on. Although CNNs has achieved the most advanced results in many fields, but are subject to the output format and not well suited for a sequential data. In our work, we consider a more effective deep learning approach to address the radar multi-target tracking problem.

Early studies have demonstrated the potential of RNNs for state estimation and tracking problem [10]. RNNs can store and exploit the past information through feedback loops, which enable it to learn temporal kinematic behavior of time series [11]. Recently, RNNs and its variant LSTMs with a structure of gate show the outstanding performance for time series tasks, such as connected handwriting recognition [12]. Many attempts were carried on in order to solve the tracking problem. Alahi et al. [13] presented an LSTM-based model to predict human trajectories, but the human movement follows common sense rules and social customs, their method it is not suitable for the unconstrained environment. A framework for deep tracking was presented in [14], which can predict the state avoiding any hand-crafting operations. Then, an entirely RNNs structure is proposed to accomplish all the multi-target tracking tasks [3], which has made a significant step for online multi-
```
3. Overall architecture

In this paper, we present a JPDA-recurrent neural network architecture (Figure 1) to tracking multiple targets from radar data in the dense clutter environment. The tracking problem is decomposed into three processing stages: Under the premise that the predicted positions of the targets from the previous time and the current measurements are given. First, the assignment probability matrix is computed by JPDA. Then, the position and the existence probability of each target are computed separately. These positions and existence probabilities are then used to perform track management. Note that before the track management stage, all measurements are assumed to be targets. In the following sections, each of these stages is detailed except the data association stage as a detailed derivation can be found in [6].

3.1. Notation

At each time-instant $t$, a vector $x_t \in \mathbb{R}^{N \times D}$ represents the states of all targets, where the $i^{th}$ target among the $N$ targets in the scene is represented by its $xy$-coordinates $(x_{ti}^t, y_{ti}^t)$, such that $D = 2$. Similarly, $M$ measurements at time-instant $t$ are represented by $z_t \in \mathbb{R}^{M \times D}$.

$\beta \in [0,1]^{N \times (M+1)}$ is the assignment probability matrix (calculated by JPDA) which indicates the probability that measurements originate from the targets in track, i.e. $\beta_{t \cdot j}$ (the $j^{th}$ measurement comes from the $i^{th}$ target), where $\forall_t: \sum_j \beta_{t \cdot j} = 1$. In addition, $\beta_{t \cdot (M+1)}$ corresponds to the probability that there is no measurement comes from the $i^{th}$ target, or to the probability that the current measurement is a false alarm (or clutter) or a missed measurement.

4. Time series prediction

This section describes the second processing stage of the tracking architecture.

4.1. Position Prediction with LSTMs

Position prediction is to predict future position based on past trends, which can be viewed as a time series prediction task. To that end, an LSTM-based model is designed, which is illustrated in Figure 2. The model can be learned totally from data and without any prior knowledge.

![Figure 2. An LSTM-based model for position prediction.](image)

At a time $t$, the hidden state $h_t$ and the cell state $c_t$ are assumed to learn the information necessary from the previous state of trajectory for predicting the state of next time $x_{t+1}$. In addition, the $x_{t+1}$ depends on $x_t$ and $z_{t+1}$ as well. $\hat{x}_{t+1}$ is calculated by weighting the measurements and the assignment probability $\beta$, the $\hat{x}_{t+1}$ is simply given by:

$$\hat{x}_{t+1} = \sum_{j=1}^{M} z_{t+1}^j \beta_{t+1}^{i \cdot j}$$

(1)

The model outputs are as follows:
Where $\hat{x} = [W_{x\hat{x}}x_t; W_{\hat{x}\hat{z}}\hat{z}_{t+1}]$ means connecting $x_t$ and $\hat{z}_{t+1}$ together, the operator $\odot$ denotes dot product. Sigmoid $\sigma(\cdot)$ and hyperbolic tangent $\tanh(\cdot)$ are activation functions, and the learnable parameters are expressed by $W$. In order to calculate the fitness of the model, we train the model by minimizing the mean squared error (MSE) loss of all targets:

$$L(x_t, \bar{x}_t) = \frac{1}{N_0} \sum ||x_t - \bar{x}_t||^2$$

Where $x$ is the predicted value, and $\bar{x}$ is the ground truth.

4.2. Existence probability prediction with RNNs.

A measure at a time may originate from one of the existing targets, be a false alarm or came from a new target. It is described above that the assignment probability $\beta$ indicates whether a target is existent. Therefore, a variable $\epsilon \in [0,1]^N$ is used to denote the existence probability of a target. An RNN-based model is proposed which can learn the correlation between the $\beta$ and $\epsilon$ to output the existence probability. This model is illustrated in Figure 3. The definitions of the output are as follows:

$$h_{t+1} = \tanh([W_{\beta}\beta_{t+1} + W_{e\epsilon}\epsilon_t] + W_{hh}h_t)$$

$$\epsilon_{t+1} = W_{\epsilon h}h_{t+1}$$

Where the initial value of $\epsilon_1$ is 0.5. During testing, once $\epsilon$ is bellowing an existence threshold $\gamma$ (0.5 in our experiments), this measurement will be considered to be a clutter.

The parameters of the RNN-based model are learned by MSE loss as well:

$$L(\epsilon_t, \bar{\epsilon}_t) = \frac{1}{N} \sum ||\epsilon_t - \bar{\epsilon}_t||^2$$

Where $\epsilon$ is the predicted value, and $\bar{\epsilon}$ denotes the true value.

5. Track management

In the actual tracking problem, the time of the target appears and disappears within the surveillance area is unpredictable, which complicates tracking management and target tracking. There are several operations for tracking management, tracking initialization, maintenance and termination.

This paper proposes an algorithm that considers the existence probability of the previous time and decides which tracking management operation to apply. An additional paired variables ($s_t \in (0,1), \epsilon_t \in (0,1)$) are used to determine when to initialize, maintain or terminate a track:

$$(s_t, \epsilon_t) = \begin{cases} (0,0) & \text{no track exists} \\ (1,0) & \text{initialization} \\ (0,1) & \text{termination} \\ (1,1) & \text{maintenance} \end{cases}$$

For a possible trajectory, the tracking management is activated at $\theta^{th}$ time, where $\theta$ is a time threshold (3 in our experiments). When the first $\theta$ values from current time $t$ of $\epsilon_t$ are under (bellow) the existence threshold, $s_t = 1$ ($\epsilon_t = 1$).
6. Experiments
Data simulations are conducted to demonstrate the efficacy of the proposed approach in all the tasks of tracking multiple targets. We calculate the Optimal Sub-pattern assignment (OSPA, [15]) distance, which is a mathematically rigorous metric to captures all difference between the real tracks and the tracking results, including the cardinal error, the localization error, and the labeling error. We choose $p = 10$ and $c = 300$. The proposed architecture is implemented in Lua and Torch7, and all the experiments are tested in an NVIDIA Titan Xp GPU and an E2-2630 CPU. Both models are trained with one layer and a fixed hidden state dimension of 50. Both models are trained with 100 sequences of length 500–1000 time steps, where the training data are divided into 10 consecutive samples and normalized with the zero-mean method. Second, RMSprop [16] is used to minimize the loss, the rate of learning is 0.005, and the iteration number is 50,000. The training of these models takes approximately 3 hours.

In our experimental scenario, we set an area of 15km×15km, consider a time-varying number of targets with velocity from 100m/s to 300m/s. There are 5 targets having various birth times and lifespans, and a lot of clutter with uniform distribution. Targets are either in uniform motion or in variety acceleration motion with acceleration from $5m/s^2$ to $10m/s^2$.

In Table 1, we show the average and standard deviation for the OSPA distance and the related terms (cardinal error and localization error). The proposed method (RNN-JPDA) is compared with the conventional NNDA and JPDA when the Kalman Filter (KF) used for estimating the state of the targets. The RNN-JPDA is superior to other methods in terms of overall OSPA. Despite the higher OSPA cardinality error than others, the Loc and OSPA distance are better respectively.

| Method       | OSPA Card | OSPA Loc | OSPA          |
|--------------|-----------|----------|---------------|
| KF-NNDA      | 8.14±43.69| 81.27±42.66 | 87.99±54.13  |
| KF-JPDA      | 8.29±44.58| 79.77±32.39 | 84.78±52.65  |
| RNN-JPDA (ours) | 13.89±57.66 | 74.51±30.92 | 77.46±47.59  |

Figure 4 illustrates the tracking results of the proposed method. We can see that our method can track different targets in complex clutter environment, and even the targets cross each other.

Moreover, in Figure 5, we show the predicted number of targets for different methods at every time step. The general trend is that it has delayed initiation and termination, which is inevitable in online tracking. Correspondingly, the OSPA (see Figure 6) has a higher value at the time of birth and death of the targets. Another reason for the higher OSPA value is that LSTMs has not learned the motion in the early steps of the target.

7. Conclusion
This paper presented an architecture that using JPDA-Recurrent Neural Networks, which can address the state prediction, data association, and track management for multi-target tracking in a clutter environment. By utilizing the assignment probability calculated by JPDA, two models based on LSTMs and LSTMs are used to complete the tasks of time series prediction respectively. By sharing
the information between the two models, management can be performed automatically. Finally, the approach is demonstrated in the simulation scenario. We think it should be considered deeply in the future work by incorporating velocity, acceleration, and data association method to reach better performance in a realistic situation.

**Figure 5.** Comparison of the predicted number of targets for different methods.

**Figure 6.** Comparison of OSPA for different methods.

**References**

[1] Pillai S K, Seshagiri Rao V, Balakrishnan S S, and Sankar S 1984 A New Method of Model Compensation in Kalman Filter Application to Launch Vehicles In Proc. of the 14th Int. Symp. on Space Technology and Science (Tokyo)

[2] Colegrove S B, Davis A W, and Ayliffe J K 1986 Track initiation and nearest neighbors incorporated into probabilistic data association Journal of Electrical and Electronics Engineers, (Australia) 6 pp 191–198

[3] Milan A, Rezatofighi S H, Dick A, Reid I, and Schindler K 2017 Online multi-target tracking using recurrent neural networks 31st AAAI Conf. on Artificial Intelligence 5 pp 4225-32

[4] Reid D 1979 An algorithm for tracking multiple targets IEEE Transaction on Automatics Control 24 pp 1202-11

[5] Fortmann T E, Bar-Shalom Y, Scheffe M 1980 Multi-target tracking using joint probabilistic data association Decision and Control Including the Symp. on Adaptive Processes pp 807-812

[6] Kim C, Li F, Ciptadi A, and Rehg J M 2015 Multiple Hypothesis Tracking Revisited IEEE Int. Conf. on Computer Vision pp 4696-4704

[7] Rezatofighi S H, Milan A, Zhang Z, Shi Q 2015 Joint Probabilistic Data Association Revisited IEEE Int. Conf. on Computer Vision IEEE Computer Society pp 3047-55

[8] Haykin S, Kosko B 2009 Gradient-Based Learning Applied to Document Recognition Wiley-IEEE Press pp 306-351

[9] Krizhevsky A, Sutskever I, Hinton G E. 2012 ImageNet classification with deep convolutional neural networks Int. Conf. on Neural Info. Proc. Systems Curran Associates Inc. 60 pp 1097-11

[10] Chenna S K, Jain Y K, Kapoor H et al 2004 State Estimation and Tracking Problems: A Comparison Between Kalman Filter and Recurrent Neural Networks Int. Conf. on Neural Info. Proc. (Springer, Berlin, Heidelberg) pp 275-281

[11] Milos M 2012 Comparative analysis of Recurrent and Finite Impulse Response Neural Networks in Time Series Prediction Indian Journal of Computer Science and Eng. pp 180-191

[12] Graves A, Liwicki M, Fernández S, Bertolami R, Bunke H, and Schmidhuber J 2009 A novel connectionist system for unconstrained handwriting recognition IEEE Trans Pattern Anal Machine Intelligence 31 pp 855-868

[13] Alahi A, Goel K, Ramanathan V, Robicquet A, Li F F, and Savarese S 2016 Social LSTM: Human Trajectory Prediction in Crowded Spaces IEEE Conf. on Computer Vision and Pattern Recognition pp 961-971

[14] Ondruska P, Posner I 2016 Deep Tracking: Seeing Beyond Seeing Using Recurrent Neural Networks Proc. of the 13th AAAI Conf. on Artificial Intelligence pp 3361-67
[15] Ristic B, Vo B N, Clark D, and Vo B T 2011 A Metric for Performance Evaluation of Multi-Target Tracking Algorithms *IEEE Transactions on Signal Proc.* 59 pp 3452-34
[16] Tieleman T, and Hinton G 2012 Lecture 6.5-Rmsprop: Divide the gradient by a running average of its recent magnitude *Coursera: Neural networks for machine learning*