Online Multi-Target Tracking Using Recurrent Neural Networks

CSE 291D

02/28/2018
Haotian Zhang
Outline

1. Introduction
2. Background
3. Approach
4. Training Data
5. Experiment
6. Conclusion
Introduction -- Motivation

1. Exploit power of deep learning for multi-target tracking
2. Data-driven approach, first step towards end-to-end learning (Q: Why End-to-End)
3. Efficient inference (up to 300 Hz on a CPU)
Introduction -- Challenges

Challenges:

1. an a-priori unknown and time-varying number of targets,
2. a continuous state estimation of all present targets, and
3. a discrete combinatorial problem of data association.
Background -- What is RNNs?
Background -- What is RNNs? (Q: Why not CNNs)
Background -- RNNs and CNNs

RNNs -- recognize patterns across time: Hot -> Dog, Big -> dog

1. Arbitrary input/output lengths.
2. Ideal for text and speech analysis.

CNNs -- recognize patterns across space: Hot Dog, Big Dog

1. Fixed size input and generate fixed-size outputs.
2. Ideal for images and videos processing.
Background -- What is LSTM

The repeating module in a standard RNN contains a single layer.

Can’t handle the combinatorial task of data association.
Background -- What is LSTM

The repeating module in an LSTM contains four interacting layers.

Can handle the combinatorial task of data association.

Up: Pure RNNs

Down: RNNs with LSTM
Approach -- High Level

A schematic illustration of our architecture. We use RNNs for temporal prediction and update as well as track management. The combinatorial problem of data association is solved via LSTMs for each frame.

Based on **Bayesian filtering**

\[
p(x_t | z_{1:t}) \propto p(z_t | x_t) \int p(x_t | x_{t-1}) p(x_{t-1} | z_{1:t-1}) dx_{t-1}
\]
Approach -- Low Level Building Blocks

1. One can isolate and debug individual components effectively.
2. Framework becomes modular, making it easy to replace each module or to add new ones.
3. Enable one to (pre)train every block separately, which not only significantly speeds up the learning process but turns out to be necessary in practice to enable convergence.

Left: An RNN-based architecture for state prediction, state update, and target existence probability estimation.

Right: An LSTM-based model for data association.
Approach -- Target Motion

1. Prediction: Learn a complex dynamic model for predicting target motion in the absence of measurements.

2. Update: Learn to correct the state distribution, given target-to-measurement assignments.

3. Birth / death: Learn to identify track initiation and termination based on the state, the measurements and the data association.
Approach -- Target Motion Loss

\[ L(x^*, x, \mathcal{E}, \tilde{x}, \tilde{\mathcal{E}}) = \frac{\lambda}{ND} \sum_{\text{prediction}} \| x^* - \tilde{x} \|^2 + \frac{\kappa}{ND} \| x - \tilde{x} \|^2 + \nu L_\mathcal{E} + \xi \mathcal{E}^* \]

Note that omit the time index is omit here for better readability. In practice the loss for one training sample is averaged over all frames in the sequence.
Approach -- Initiation and Termination

The effect of the pairwise smoothness prior on the existence probability.

Binary cross entropy (BCE) loss

\[ \mathcal{L}_\mathcal{E}(\mathcal{E}, \widetilde{\mathcal{E}}) = \widetilde{\mathcal{E}} \log \mathcal{E} + (1 - \widetilde{\mathcal{E}}) \log(1 - \mathcal{E}) \]
Approach -- Data Association with LSTMs

Exploit the LSTM’s temporal step-by-step functionality to predict the assignment for each target one target at a time.

\[
\mathcal{L}(A^i, \tilde{a}) = - \log(A_{i\tilde{a}});
\]

common negative log-likelihood loss to measure the misassignment cost
Training Data

Very limited amount of labelled data for pedestrian tracking is publicly available today.

Synthetic generation by sampling from a simple generative trajectory model learned from real data.

Approximately 100K 20-frame long sequences. The data is divided into mini-batches of 10 samples per batch and normalised to the range $[-0.5, 0.5]$, w.r.t. the image dimensions.
Experiment - Synthetic

Five targets with random birth and death times are generated in a rather cluttered environment. The initiation / termination indicators are illustrated in the bottom row.

Top: Ground truth (x-coordinate vs. time).
Middle: Our reconstructed trajectories.
Bottom: The existence probability $E$ for each target.

Note the delayed initiation and termination.
Experiment - Real MOTChallenge 2015 benchmark

22 video sequences (11/11 for training and testing, respectively)

<table>
<thead>
<tr>
<th>Method</th>
<th>Rcll↑</th>
<th>Prcn↑</th>
<th>MT↑</th>
<th>ML↓</th>
<th>FP↓</th>
<th>FN↓</th>
<th>IDs↓</th>
<th>FM↓</th>
<th>MOTA↑</th>
<th>MOTP↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kalman-HA</td>
<td>28.5</td>
<td>79.0</td>
<td>32</td>
<td>334</td>
<td>3,031</td>
<td>28,520</td>
<td>685</td>
<td>837</td>
<td>19.2</td>
<td>69.9</td>
</tr>
<tr>
<td>Kalman-HA2*</td>
<td>28.3</td>
<td>83.4</td>
<td>39</td>
<td>354</td>
<td>2,245</td>
<td>28,626</td>
<td>105</td>
<td>342</td>
<td>22.4</td>
<td>69.4</td>
</tr>
<tr>
<td>JPDA(_m)*</td>
<td>30.6</td>
<td>81.7</td>
<td>38</td>
<td>348</td>
<td>2,728</td>
<td>27,707</td>
<td>109</td>
<td>380</td>
<td>23.5</td>
<td>69.0</td>
</tr>
<tr>
<td>RNN.HA</td>
<td>37.8</td>
<td>75.2</td>
<td>50</td>
<td>267</td>
<td>4,984</td>
<td>24,832</td>
<td>518</td>
<td>963</td>
<td>24.0</td>
<td>68.7</td>
</tr>
<tr>
<td>RNN.LSTM</td>
<td>37.1</td>
<td>73.5</td>
<td>50</td>
<td>260</td>
<td>5,327</td>
<td>25,094</td>
<td>572</td>
<td>983</td>
<td>22.3</td>
<td>69.0</td>
</tr>
</tbody>
</table>

Table 1: Tracking results on the MOTChallenge training dataset. *Denotes offline post-processing.

<table>
<thead>
<tr>
<th>Method</th>
<th>MOTA↑</th>
<th>MOTP↑</th>
<th>MT%↑</th>
<th>ML%↓</th>
<th>FP↓</th>
<th>FN↓</th>
<th>IDs↓</th>
<th>FM↓</th>
<th>FPS↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDP (Xiang et al. 2015)</td>
<td>30.3%</td>
<td>71.3%</td>
<td>13.0</td>
<td>38.4</td>
<td>9,717</td>
<td>32,422</td>
<td>680</td>
<td>1,500</td>
<td>1.1</td>
</tr>
<tr>
<td>SCEA (Hong Yoon et al. 2016)</td>
<td>29.1%</td>
<td>71.7%</td>
<td>8.9</td>
<td>47.3</td>
<td>6,060</td>
<td>36,912</td>
<td>604</td>
<td>1,182</td>
<td>6.8</td>
</tr>
<tr>
<td>JPDA(_m)* (Rezatofighi et al. 2015)</td>
<td>23.8%</td>
<td>68.2%</td>
<td>5.0</td>
<td>58.1</td>
<td>6,373</td>
<td>40,084</td>
<td>365</td>
<td>869</td>
<td>32.6</td>
</tr>
<tr>
<td>TC.ODAL (Bae and Yoon 2014)</td>
<td>15.1%</td>
<td>70.5%</td>
<td>3.2</td>
<td>55.8</td>
<td>12,970</td>
<td>38,538</td>
<td>637</td>
<td>1,716</td>
<td>1.7</td>
</tr>
<tr>
<td>RNN.LSTM (ours)</td>
<td>19.0%</td>
<td>71.0%</td>
<td>5.5</td>
<td>45.6</td>
<td>11,578</td>
<td>36,706</td>
<td>1,490</td>
<td>2,081</td>
<td>165.2</td>
</tr>
</tbody>
</table>

Table 2: Tracking results on the MOTChallenge test dataset. *Denotes an offline (or delayed) method.
Conclusion

Pro: Two orders of magnitude faster than the top accuracy in pedestrian online tracking.

Con: Not top accuracy in pedestrian online tracking.

Contributions:

1. Addressed the challenging problem of data association and trajectory estimation within a neural network.

2. The first approach that employs RNNs to address online multi-target tracking.

3. RNN-based approach can be utilised to learn complex motion models in realistic environments.

4. An LSTM network is able to learn one-to-one assignment, which is a non-trivial task for such an architecture.
Thanks
Reference:

https://arxiv.org/abs/1604.03635

https://www.youtube.com/watch?v=UNmqTiOnRfg

https://www.youtube.com/watch?v=WCUNPb-5EYI

http://colah.github.io/posts/2015-08-Understanding-LSTMs/