Using EM to Learn Motion Behaviors of Persons with Mobile Robots

Maren Bennewitz  Wolfram Burgard  Sebastian Thrun

Also involved: Dirk Schulz, Dirk Hähnel and many others
Motivation

- Robots that know where people are and what they do can do better!
- Examples...
Minerva
Perl: A Nursing Robot
Albert: An Interactive Service Robot
Three-Month Deployment of Albert at the HNF
Tracking People

- **Key questions**
  - How many people are there?
  - Where do they go?

- **Requirements**
  - Real time
  - No model of the environment
  - Robot in motion
Example Run
Tracking with a Moving Robot
Mapping in Populated Environments

Filtering beams corresponding to persons improves maps:
Increased Matching Accuracy by Filtering People
Learning 3d-Maps
Learning Motion Patterns

Knowledge of typical motion patterns helps robots to
- predict behavior of persons
- avoid possible conflicts
- improve their service
- ...
2D Map of a Domestic Environment, Learned by a Robot
Learning Trajectories of People in Their Homes

- Which trajectory does the person take?
- Where is the person going to?
Tracking People/Motion Segmentation

Input: Set $S$ of data sequences $s_1, ..., s_N$
What we are looking for:

- Set $\theta$ of position-sequences $\theta_1, \ldots, \theta_M$, one for each pattern.

- Correspondence table $x_{m,n}$ telling us, which data $s_n$ set belongs to which motion pattern $\theta_m$.

Problem:

How can we estimate $x_{m,n}$?
Density Representation

- One Gaussian with fixed variance for every time step of every motion pattern
Formal Specification

We want to maximize

\[ E_x[\log p(s, x | \theta)] = E[c_1 - c_2 \sum_{n=1}^{N} \sum_{m=1}^{M} x_{m,n} \log p(s_n | \theta_m)] \]

Linearity of \( E[... \text{ ]} \)

\[ = c_1 - c_2 \sum_{n=1}^{N} \sum_{m=1}^{M} E[x_{m,n}] \log p(s_n | \theta_m) \]

Gaussians

\[ = c_1 - c_2 \sum_{n=1}^{N} \sum_{m=1}^{M} \sum_{t=1}^{T} E[x_{m,n}] \cdot \|s_n^t - \mu_m^t\| \]

Extension of k-means clustering to trajectories!
Solution by Applying the EM-Algorithm

Maximize $E_x[\log p(s, x \mid \theta)]$ through an iterative sequence of models $\theta^1, \theta^2, \ldots$

E-Step:

$$E[x_{m,n}] \leftarrow \alpha \ p(s_n \mid \theta_m) = \alpha \prod_{t=1}^{T} e^{-\frac{\|x_n^t - \mu_m^t\|^2}{2\sigma^2}}$$
The M-Step

\[ \theta_m \leftarrow \arg \max_{\theta_m} \sum_{n=1}^{N} \sum_{m=1}^{M} E[x_{m,n}] \cdot \log p(s_n \mid \theta_m) \]

Since we have Gaussians with a fixed variance:

\[ \mu_m^t \leftarrow \frac{\sum_{n=1}^{N} E[x_{m,n}] \cdot x_n^t}{\sum_{n=1}^{N} E[x_{m,n}]} \]
Estimating the Number of Model Components

Whenever EM has converged to a (local) maximum:

1. Try to introduce a new motion pattern for the trajectory which has the lowest likelihood under the current model.

2. Try to eliminate the motion pattern which has the lowest utility.

Select model $\theta$ which has the highest evaluation

$$E_x[\log p(s, x | \theta)] - M\alpha$$

where $M = \#\text{model components}, \alpha = \text{penalty term}$
# Application of EM

<table>
<thead>
<tr>
<th>19</th>
<th>18</th>
<th>17</th>
<th>16</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>s0</td>
<td>s1</td>
<td>s2</td>
<td>s3</td>
<td>s4</td>
<td>s5</td>
<td>s6</td>
<td>s7</td>
<td>s8</td>
<td>s9</td>
</tr>
<tr>
<td>s10</td>
<td>s11</td>
<td>s12</td>
<td>s13</td>
<td>s14</td>
<td>s15</td>
<td>s16</td>
<td>s17</td>
<td>s18</td>
<td>s19</td>
</tr>
<tr>
<td>s20</td>
<td>s21</td>
<td>s22</td>
<td>s23</td>
<td>s24</td>
<td>s25</td>
<td>s26</td>
<td>s27</td>
<td>s28</td>
<td>s29</td>
</tr>
<tr>
<td>s30</td>
<td>s31</td>
<td>s32</td>
<td>s33</td>
<td>s34</td>
<td>s35</td>
<td>s36</td>
<td>s37</td>
<td>s38</td>
<td>s39</td>
</tr>
<tr>
<td>s40</td>
<td>s41</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Model Selection

![Graph showing the number of motion patterns and model evaluation over iterations. The number of motion patterns decreases initially, then stabilizes, while the model evaluation increases, showing a non-linear relationship.]
Clustering Results
Prediction Accuracy

![Graph showing prediction accuracy for office and home environments. The X-axis represents the length of the trajectory in percentage, and the Y-axis represents the correctly classified trajectories in percentage. The graph shows two lines: one for the office environment and one for the home environment. The office environment line is consistently higher than the home environment line, indicating better prediction accuracy in the office environment.]
Why it’s sometimes difficult ...

during learning:

... because there are serious overlaps!

during classification:
Conclusions and Future Work

- Technique to learn motion patterns of people in home and office environments.
- Learning more abstract patterns (lower complexity models, e.g. linear piecewise approximations)
- Adapting the robot’s behavior according to the predicted behavior
- Applications
Example: Markov Chains
Thanks ...

... and goodbye!