Active Continual Learning for Planning and Navigation

Ahmed Qureshi, Yinglong Miao, Michael C. Yip

Correspondence: a1quresh@ucsd.edu

UC San Diego

Motion Planning

Find a path that satisfies all constraints between the given start and goal configurations.

- Collision Avoidance
- Dynamics
- Kinematics (e.g., end-effector)

Karaman et. al, 2011
Sequence in Robot Thinking

- Perceive
- Search
- Start Moving

Biological Planning?
Encoder Network (Enet):

- Input: obstacles point cloud $x_{\text{obs}} \in \mathbb{R}^d$
- Output: Embedding $Z \in \mathbb{R}^m$
- 3D CNN (Preprocess point-cloud to voxel)
- Feed forward neural network

Planning Network (Pnet):

- Input: $Z, c_t, c_T$
- Output: $\hat{c}_{t+1} \leftarrow \text{PNet}(c_t, c_T, Z)$
- Stochastic feed-forward neural network

Training Methods:

- Batch offline learning
- Active continual learning

Planning Algorithm:

- Bidirectional iterative planning method.
- Informed sampler integrated with SMPs
MPNet: Training Methods

**Batch Learning**

- MPNet Training Module
- Oracle Planner
- Problems
- Environment

**Active Continual Learning**

- MPNet Planning Module
- Training Module
- Oracle Planner
- Demonstration
- Path Solution
- Found Solution
- No Path

Mathematical Formulations:

1. \[ \frac{1}{N_p} \sum_{j,t} \left\| \hat{c}_{j,t+1} - c_{j,t+1} \right\|^2 \]

2. \[ \min_{\theta} l(f^t_\theta(s), y) \text{ s.t. } \mathbb{E}_{(s,y) \sim M}[l(f^t_\theta(s), y)] \leq \mathbb{E}_{(s,y) \sim M}[l(f^{t-1}_\theta(s), y)] \]

3. \[ \langle g, g_M \rangle = \langle \nabla_\theta l(f_\theta(s), y), \mathbb{E}_{(s,y) \sim M} \nabla_\theta l(f_\theta(s), y) \rangle \]

4. If \( \langle g, g_M \rangle < 0 \):
   \[ \min_{g'} \| g - g' \| \text{ s.t. } \langle g', g_M \rangle \geq 0 \]

Lopez-Paz et al., 2017
Active Continual Learning

\[ \langle g, g_M \rangle = \langle \nabla_\theta l(f_\theta(s), y), \mathbb{E}_{(s,y)\sim M} \nabla_\theta l(f_\theta(s), y) \rangle \]
\[ \text{if } \langle g, g_M \rangle < 0: \]
\[ \min_{g'} \| g - g' \| \text{ s.t. } \langle g', g_M \rangle \geq 0 \]

Learning from streaming data:
1. Given a feasible trajectory \( \sigma \)
2. \( M \leftarrow \text{Update episodic memory (} \sigma \text{)} \)
3. \( g_M \leftarrow \mathbb{E}_{(s,y)\sim M} \nabla_\theta l(f_\theta(s), y) \)
4. \( g \leftarrow \mathbb{E}_{(s,y)\sim \sigma} \nabla_\theta l(f_\theta(s), y) \)
5. project \( g \) to \( g' \) using above eq
6. Update MPNet parameters \( \theta \) with \( g' \)
MPNet: Online Path Generation
MPNet: Data Efficient Active Continual Learning
Thank you