Decision transformer for multitask offline RL

Decision transformer - ML model to make a decision based on the agent's state.
Offline RL - training is done using a dataset of recorded interactions, instead of live interactions with the environment.
Multitask - the agent should learn to solve similar but new tasks based on examples.

01 Introduction

In the field of Artificial Intelligence (AI), techniques like Reinforcement Learning (RL) and Decision Transformer are utilized by machines to learn from experiences and solve problems. This approach is particularly valuable in situations where traditional training methods are costly or challenging. For instance, in robotics, multi-task learning enables robots to use experiences from various tasks, such as picking up objects, to solve new problems like placing them down. The use of offline datasets, enables the machines to learn without exploiting real environment which might be costly.

02 Objective

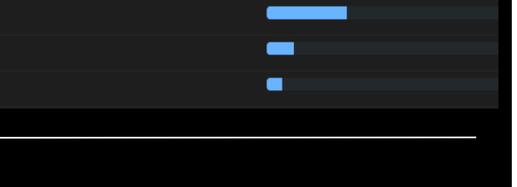
This research explores the effectiveness of Decision Transformers in multi-task environments through theoretical discussions and practical examples. It also tries to answer the question, of how introducing sub-optimal training data, affects the performance or generalisation ability of the model

03 Methodology

The process consists of the following main steps:

1.. Research and data gathering.
2. Analysis of performance using optimal data.
3. Analysis of performance using sub-optimal data.
4. Present the findings

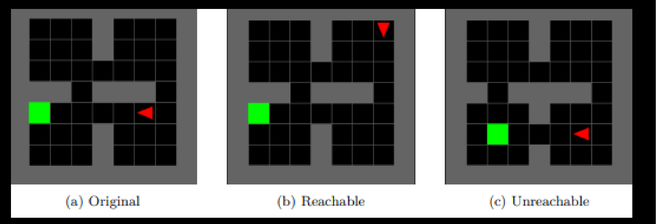
For the multitask environment, a version of minigrid is used. For the decision transformer implementation, the D3RLPY library was chosen.



03 Background

Transformers are designed to efficiently model sequential data. These models are composed of stacked self-attention layers with residual connections. The main benefit of using a Decision Transformer is that it can reason about the state within a certain context, and not only based on the most recent state.

Reachability - To summarise the idea: A reachable state 's' is a state for which there exists a policy whose probability of encountering 's' during training is non-zero.



Path stitching - introducing suboptimal data to the dataset may positively influence its performance in certain scenarios

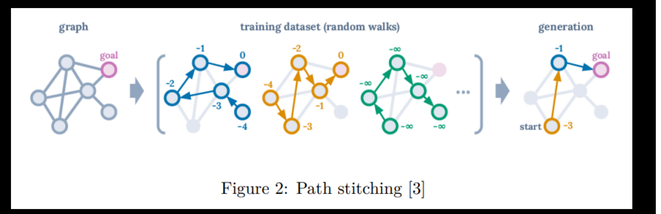


Figure 2: Path stitching [3]

04 Hyperparameter tuning

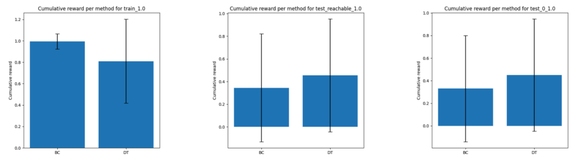
To correctly evaluate the performance of the model, hyperparameter tuning was performed. It is worth noticing, that the Offline models tend to be unstable. Thus the random seed had similar influence on the model to other hyper parameters.

05 Benchmarks

To ensure a reliable comparison of the model performance, the following benchmark models will be used:
- Optimal trajectory - compare the number of steps taken by the model to those required by the shortest path algorithm.
- Behaviour Cloning - for convenient comparison of the methods across different environments, Behaviour Cloning is also going to be used as a benchmark.

06 Results

Models trained on optimal data:



(a) Train data (b) Reachable test data (c) Unreachable test data

Figure 8: Results of Optimal Evaluations

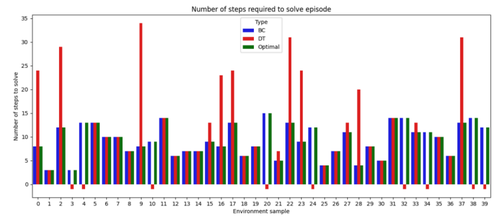
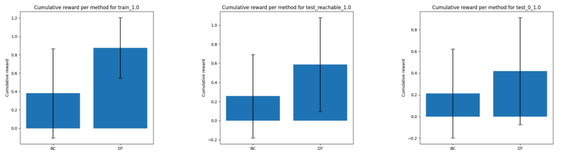


Figure 10: Runs for models trained on optimal train data, evaluated on train environment

Models trained on sub-optimal data:



(a) Train data (b) Reachable test data (c) Unreachable test data

Figure 9: Results of Sub-optimal Evaluations

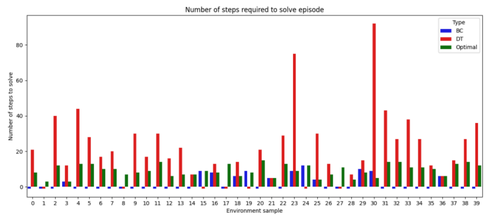
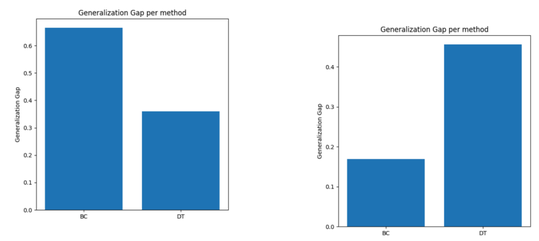


Figure 11: Runs for models trained on sub-optimal train data, evaluated on train environment

Generalization Gap:



(a) Generalization gap when trained on optimal data

(b) Generalization gap when trained on sub-optimal data

Figure 12: Generalization Gap

07 Conclusions

Optimal dataset performance:
- Behavioral Cloning (BC) outperforms Decision Transformer (DT) on optimal dataset training and evaluation.
- Both models significantly drop in performance on unseen data (Reachable and Unreachable): DT by about 40%, BC by about 60%.
- Figure 8 demonstrates high variance in average results, reflecting typical randomness in these scenarios.
- BC shows reliable performance when trained and evaluated with optimal data.
- DT surprisingly outperforms BC on unseen data, potentially due to its ability to capture longer contexts, beneficial for handling unseen problems.

Sub-optimal dataset performance:
- Training on the sub-optimal dataset improved the Decision Transformer's (DT) performance on train and reachable data.
- DT's performance slightly decreased on unreachable data.
- The improvement is due to additional examples in the mixed dataset, providing more "reachable" environment examples.
- Behavioral Cloning (BC) performance decreased across all testing environments.
- BC's decrease is expected as sub-optimal data introduced conflicting actions for the same state.
- BC struggles with sub-optimal data because it doesn't capture longer contexts.

Steps per environment run:
- When trained on optimal data:
  - Behavioral Cloning (BC) usually performs perfectly.
  - Decision Transformer (DT) often fails to complete tasks or completes them suboptimally.
- When trained on sub-optimal data:
  - BC rarely solves any instances of the training environment.
  - DT manages to solve the majority of instances, despite its behavior being suboptimal most of the time.

Answer research question - The performance of the Decision Transformer in Offline RL is poor. It achieves up to 80% accuracy on seen examples, indicating it can learn and reproduce from training data. However, on unseen environments, its performance drops, solving less than 60% of cases consistently.

Answer sub-question:
Introducing sub-optimal data improved the performance of the Decision Transformer in training and reachable environments, and had little effect on unseen environments.

Generalisation Gap:
- Consider not only the gap but also the overall model performance.
- The drop in gap for Behavioral Cloning (BC) is primarily linked to its performance decline in the training environment when trained on sub-optimal data.
- For Decision Transformer (DT), the apparent increase in generalization gap is mostly due to changes in performance in one testing environment, driven by improved performance on the training dataset evaluation.

Author:
Piotr Bieszczad
Thesis committee:
Matthijs Spaan,
Max Weltevrede,
Elena Congeduti

TU Delft