Neural Networks for Predicting Algorithm Runtime Distributions

**Problem**

Algorithms often rely on random choices and decisions, hence their runtime can be described by a runtime distribution (RTD). In this work, we study how to predict parametric RTDs for unseen instances:

- A randomized algorithm \( A \)
- A set of instances \( \Pi \)
- For each instance \( \pi \in \Pi \):
  - \( m \) instance features \( f(\pi) \)
  - \( m \) runtime observations \( t(\pi) \) obtained by executing \( A \) on \( \pi \) with \( k \) different seeds.

The goal is to learn a model that can predict \( A \)'s RTD well for unseen instances \( \pi_{n+1} \) with given features \( f(\pi_{n+1}) \).

**In a Nutshell**

We compare different ways of predicting RTDs.

1. We propose DistNet, which can be trained using the loss function of interest and jointly predicts parameters of RTDs.
2. DistNet can be trained with less data than previous methods.

**Existing Approaches**

- Algorithm \( A \)
  - Run \( A \) \( k \) times on each \( \pi \in \Pi_{train} \)
  - Compute instance features \( f(\pi) \)
  - Use \( f(\pi) \) to predict \( \Theta \) for \( \pi_{n+1} \)

**Our Approach**

For each instance \( \pi \), fit the parametric distribution's parameters \( \Theta(\pi) = (\theta_1, ..., \theta_p) \) on observed runtimes to get training data \( f(\pi), t(\pi) \) \( \in \Pi_{train} \).

For each instance \( \pi \), the loss function of interest:

\[
\mathcal{L}(\pi) = -\sum_{\pi \in \Pi_{train}} \sum_{t=1}^{k} \log L_D(\Theta_\pi f(\pi), t(\pi))
\]

**Pipeline for predicting RTDs**

1. **Algorithm \( A \)**
   - Run \( A \) \( k \) times on each \( \pi \in \Pi_{train} \)
   - Compute instance features \( f(\pi) \)
   - Use \( f(\pi) \) to predict \( \Theta \) parameters

2. **Data**
   - \( f(\pi), t(\pi) \) for each \( \pi \in \Pi_{train} \)

3. **Estimate RTD family \( D \)**

4. **Fit RTD model \( \Theta \) to \( f(\pi) \rightarrow \Theta \)**

5. **New instance \( \pi_{n+1} \)**
   - Compute features \( f(\pi_{n+1}) \)
   - Use \( f(\pi_{n+1}) \) to predict \( \Theta \) parameters

**Future Work**

- Use a mixture of models to handle less homogeneous instance sets
- Consider an algorithm's configuration as an additional input
- Study non-parametric models

**Results**

- **Advantages and Limitations**
  - **DistNet** jointly learns distribution parameters and directly optimizes the loss function of interest.
  - **DistNet** can learn from only a few samples per instance.
  - We assume homogeneous instance sets.
  - We need to know beforehand which distribution family to use.

- **Open Questions & Future Work**
  - Use a mixture of models to handle less homogeneous instance sets.
  - Consider an algorithm's configuration as an additional input.
  - Study non-parametric models.

**Table 1** Parameters and configurations:

- **Scenario**
  - Clasp-Grover
  - Saps-CV-VAR
  - YaI-SAT-QCP
  - Syper-QCP
  - YaI-SAT-QCP
  - Syper-SVRG
  - LPG-Zerocat

- **Advantages and Limitations**
  - **DistNet** jointly learns distribution parameters and directly optimizes the loss function of interest.
  - **DistNet** can learn from only a few samples per instance.
  - We assume homogeneous instance sets.
  - We need to know beforehand which distribution family to use.

**Open Questions & Future Work**

- Use a mixture of models to handle less homogeneous instance sets.
- Consider an algorithm's configuration as an additional input.
- Study non-parametric models.

**Table 2** Parameters and configurations:

- **Scenario**
  - Clasp-Grover
  - Saps-CV-VAR
  - YaI-SAT-QCP
  - Syper-QCP
  - YaI-SAT-QCP
  - Syper-SVRG
  - LPG-Zerocat

- **Advantages and Limitations**
  - **DistNet** jointly learns distribution parameters and directly optimizes the loss function of interest.
  - **DistNet** can learn from only a few samples per instance.
  - We assume homogeneous instance sets.
  - We need to know beforehand which distribution family to use.

**Open Questions & Future Work**

- Use a mixture of models to handle less homogeneous instance sets.
- Consider an algorithm's configuration as an additional input.
- Study non-parametric models.

**Table 2** Parameters and configurations:

- **Scenario**
  - Clasp-Grover
  - Saps-CV-VAR
  - YaI-SAT-QCP
  - Syper-QCP
  - YaI-SAT-QCP
  - Syper-SVRG
  - LPG-Zerocat

- **Advantages and Limitations**
  - **DistNet** jointly learns distribution parameters and directly optimizes the loss function of interest.
  - **DistNet** can learn from only a few samples per instance.
  - We assume homogeneous instance sets.
  - We need to know beforehand which distribution family to use.

**Open Questions & Future Work**

- Use a mixture of models to handle less homogeneous instance sets.
- Consider an algorithm's configuration as an additional input.
- Study non-parametric models.

**Table 2** Parameters and configurations:

- **Scenario**
  - Clasp-Grover
  - Saps-CV-VAR
  - YaI-SAT-QCP
  - Syper-QCP
  - YaI-SAT-QCP
  - Syper-SVRG
  - LPG-Zerocat

- **Advantages and Limitations**
  - **DistNet** jointly learns distribution parameters and directly optimizes the loss function of interest.
  - **DistNet** can learn from only a few samples per instance.
  - We assume homogeneous instance sets.
  - We need to know beforehand which distribution family to use.

**Open Questions & Future Work**

- Use a mixture of models to handle less homogeneous instance sets.
- Consider an algorithm's configuration as an additional input.
- Study non-parametric models.

**Table 2** Parameters and configurations:

- **Scenario**
  - Clasp-Grover
  - Saps-CV-VAR
  - YaI-SAT-QCP
  - Syper-QCP
  - YaI-SAT-QCP
  - Syper-SVRG
  - LPG-Zerocat

- **Advantages and Limitations**
  - **DistNet** jointly learns distribution parameters and directly optimizes the loss function of interest.
  - **DistNet** can learn from only a few samples per instance.
  - We assume homogeneous instance sets.
  - We need to know beforehand which distribution family to use.

**Open Questions & Future Work**

- Use a mixture of models to handle less homogeneous instance sets.
- Consider an algorithm's configuration as an additional input.
- Study non-parametric models.