Automating Feature Subspace Exploration via Multi-Agent Reinforcement Learning

Kunpeng Liu, Yanjie Fu,
Pengfei Wang, Le Wu, Rui Bo, Xiaolin Li
Outline

- Background and Motivation
  - Problem Statement
  - Methodology
  - Evaluation
  - Conclusion
Feature Selection

All Features → Selection Techniques → Selected Features
Feature Selection as An Exploration Process

Feature selection: An iterative exploration process to find an optimal / near optimal subset of features

Selecting the Optimal Subset

Set of all Features → Generate a Subset → Learning Algorithm → Performance
Reinforcement Learning as A Tool of Exploration

- Reinforcement learning: **exploration + exploitation**

![Diagram](image)

- Applications
  - Traffic light control via RL
  - Taxi fleet management via RL
Inspiration: Can reinforcement learning help to solve/improve feature selection?

Automated Feature Subspace Exploration

Reinforcement Learning (Exploration Tool) + Feature Selection (Exploration Problem) = ?

Selecting the Optimal Subset

Generate a Subset

Learning Algorithm

Environment

Agent

State

Reward

Action

Interpreter
How can we reformulate the feature selection problem into a reinforcement learning task?
Feature Selection as A Multi-Agent Reinforcement Learning Task (1)

Agent: Each feature is controlled by a corresponding feature agent.
Feature Selection as A Multi-Agent Reinforcement Learning Task (2)

Action: Select or deselect the corresponding feature.
Feature Selection as A Multi-Agent Reinforcement Learning Task (3)

Environment: Feature subset space: \{\{f_4\}, \{f_1,f_2\}, \{f_1,f_3,f_5\}...\}.
Feature Selection as A Multi-Agent Reinforcement Learning Task (4)

State: Representation of selected feature subset. E.g., $S\{f_1,f_3,f_5\}$. 

State:
- Environment
- Feature Agent 1
- Feature Agent N
- Interpretator
- Environment (Feature Subset Space)

Action:
- Action 1 (Select/Deselect Feature 1)
- Action N (Select/Deselect Feature N)

Reward Assignment
- Reward (Accuracy + Redundancy + Relevance)
- State (Representations of Selected Feature Subset)
Feature Selection as A Multi-Agent Reinforcement Learning Task (5)

Overall Reward: Weighted sum of prediction accuracy, redundancy and relevance of selected feature subset.
Feature Selection as A Multi-Agent Reinforcement Learning Task (6)

Reward Scheme: Assign overall reward to each agent.
How can we design the assignment strategy?
Participating & Non-participating Agents

- For the current $k_{th}$ iteration
- Participating feature agents:
  - Select action ($k_{th}$ iteration) & Select action($(k-1)_{th}$ iteration)
  - Select action ($k_{th}$ iteration) & Deselect action($(k-1)_{th}$ iteration)
  - Deselect action ($k_{th}$ iteration) & Select action($(k-1)_{th}$ iteration)
- Non-participating agents:
  - Deselect action ($k_{th}$ iteration) & Deselect action($(k-1)_{th}$ iteration)
Reward Assignment Strategy

- Participating agents
  - Equally share the overall reward.
- non-participating agents:
  - 0 reward.
How can we better quantify the state representation?
Three State Representation Methods

- Meta descriptive statistics.
- Auto-encoder based representation.
- Dynamic-graph based Graph Convolutional Network (GCN).
Step 1: Draw statistics column-wisely.
Step 3: Expand the statistics matrix.
Auto-Encoder Based Representation

Selected Feature Matrix

Latent Matrix

Static Encoded Matrix

State Vector

$\begin{bmatrix} f_1 & f_3 & f_6 & f_8 & f_9 \end{bmatrix}$

Sample 1
Sample 2
Sample 3
Sample 4

$\begin{bmatrix} \text{latent } 1 \\ \text{latent } 2 \\ \text{latent } 3 \\ \text{latent } 4 \\ \text{latent } k \end{bmatrix}$

$\begin{bmatrix} \text{latent } 1 \\ \text{latent } 2 \\ \text{latent } 3 \\ \text{latent } 4 \\ \text{latent } k \end{bmatrix}$

$\begin{bmatrix} \text{latent } a \\ \text{latent } b \\ \text{latent } c \\ \text{latent } d \end{bmatrix}$

$\begin{bmatrix} \text{latent } a_1 \\ \text{latent } a_2 \\ \text{latent } a_3 \\ \text{latent } a_4 \end{bmatrix}$

Step 1: Encode column-wisely.
Step 2: Encode row-wisely.
Step 3: Expand the encoded matrix.
Dynamic-Graph Based GCN

Step 1: Draw a fully-connected graph.
Step 2: Update each node’s representation.
Step 3: Aggregate all nodes’ representations.
How can we improve the training efficiency of DQN in MARL?
GMM Based Sampling for Acceleration

- Improve quality of training data in Experience Replay.

Conventional sampling strategy.

GMM based sampling strategy.
Summary

Control

FA 1
Select/Deselect

FA 2
Select/Deselect

...:
Select/Deselect

FA N

Selected Feature Subspace

Descriptive
Auto encoder
GCN

Representation of Selected Feature Subspace (State)

Overall Reward

Reward Assignment

State Representation

Memory 1
Sample
Mini-Batch 1
Train
DQN 1

...:

Memory N
Sample
Mini-Batch N
Train
DQN N

Policy

Agents Actions Environment

Reward Scheme

Store

DQN
Minibatch
Train

Sample

Sample

Sample

Mini-Batch
Experimental Setup

- **Experimental Data**
  - The experiments are carried on a publicly available dataset with 15120 samples and 54 features.
  - https://www.kaggle.com/c/forest-cover-type-prediction/data.

- **Predictive Task**
  - The task is to classify the forest cover types into 7 classes.

- **Experimental Questions**
  - Can our study improve feature selection performance?
  - How do different reward quantification methods impact the performance of our method?
  - How do different state representation methods impact the performance of our method?
  - Can GMM sampling strategy improve exploration efficiency?
Performances over Different Classifiers and Feature Selectors

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Predictors</th>
<th>RF</th>
<th>LASSO</th>
<th>DT</th>
<th>SVM</th>
<th>XGBoost</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-Best</td>
<td></td>
<td>0.7943</td>
<td>0.8246</td>
<td>0.8125</td>
<td>0.8324</td>
<td>0.8076</td>
</tr>
<tr>
<td>mRMR</td>
<td></td>
<td>0.8042</td>
<td>0.8124</td>
<td>0.8096</td>
<td>0.8175</td>
<td>0.8239</td>
</tr>
<tr>
<td>LASSO</td>
<td></td>
<td>0.8426</td>
<td>0.8513</td>
<td>0.8241</td>
<td>0.8131</td>
<td>0.8434</td>
</tr>
<tr>
<td>RFE</td>
<td></td>
<td>0.8213</td>
<td>0.8236</td>
<td>0.8453</td>
<td>0.8257</td>
<td>0.8348</td>
</tr>
<tr>
<td>GFS</td>
<td></td>
<td>0.8423</td>
<td>0.8318</td>
<td>0.8350</td>
<td>0.8346</td>
<td>0.8302</td>
</tr>
<tr>
<td>SARLFS</td>
<td></td>
<td>0.8321</td>
<td>0.8295</td>
<td>0.8401</td>
<td>0.8427</td>
<td>0.8450</td>
</tr>
<tr>
<td>MARLFS</td>
<td></td>
<td>0.8690</td>
<td>0.8424</td>
<td>0.8583</td>
<td>0.8542</td>
<td>0.8731</td>
</tr>
</tbody>
</table>

Baselines: K-Best Selection, mRMR, LASSO, Recursive Feature Elimination (RFE), Genetic Feature Selection (GFS) and Single-Agent Reinforcement Learning Feature Selection (SARLFS).

Evaluation Metrics: overall accuracy

Our method: **MARLFS**. For the accuracies, the **higher**, the **better**.
Performances over Different Reward Functions

Variants: ACC (accuracy), RV (relevance), RD (redundancy), ACC+RV+RD. Evaluation Metrics: overall accuracy, precision, recall and F-measure.

For accuracies and bars, the **higher**, the **better**.
Performances over Different State Representation Methods

Variants:
- **MDS** (meta descriptive statistics)
- **AE** (autoencoder based representation)
- **GCN** (GCN based representation)
- **MDS+AE** (combination of MDS and AE)
- **MDS+AE+GCN** (combination of MDS, AE and GCN)

Evaluation Metrics: overall accuracy, precision, recall and F-measure.

For accuracies and bars, the **higher**, the **better**.
Performance over Different GMM Sampling Strategies

Variants: proportion of high-quality varies from 0.1 to 1.0
Evaluation Metrics: overall accuracy, precision, recall and F-measure.

For accuracies and bars, the **higher**, the **better**.
Conclusions

- Understandings
  - Feature selection is a space exploration process.
  - Feature selection can be improved by multi-agent reinforcement learning framework.

- Techniques
  - We propose three state representation methods.
  - We propose GMM-based sampling strategy.
  - We design reward quantification and assignment.
Thank you!