# ILAEDA: An Imitation Learning Based Approach for Automatic Exploratory Data Analysis

Abhijit Manatkar[1] Devarsh Patel[2]
Hima Patel[3] Naresh Manwani[4]

[1]Machine Learning Lab, KCIS, IIIT Hyderabad

[2]IISER, Pune

[3]IBM Research, India

[4]Machine Learning Lab, KCIS, IIIT Hyderabad

December 11, 2024

# Introduction

Exploratory Data Analysis (EDA) involves analyzing and visualising data in order to uncover patterns,uncover the outliers and find relations between variables. Based on the number of attributes we are analyzing we can divide it into three types:

1. **Univariate** : Histogram , Box Plots , Bar chart and Summary Statistics.
2. **Bivariate** : Scatter Plot , Correlation Coefficient.
3. **Multivariate** : Pair plots and PCA.

# AutoEDA

Doing EDA manually takes a lot of effort and time. So recent advancements have sought to automate this process through AutoEDA. Some of such methods are:

- Ydata-Profiling
- AutoViz
- Sweetivz
- Data Prep

Among the above Data Prep stands out since it's visualizations have insight notes that are interactive.

# RL based AutoEDA

- Automated EDA Advances:In recent years, methods to automate the exploratory data analysis (EDA) process have emerged.
- Deep Reinforcement Learning (RL) for AutoEDA:
  - AutoEDA is treated as a sequential decision-making problem.
  - RL algorithms are trained to predict sequences of EDA operations.
  - Rewards in these systems are based on rules or heuristics.
- ATENA:
  - ATENA is an RL-based AutoEDA system.
  - It automatically generates EDA sessions for a given tabular dataset.

# Interestingness Measures

Various interestingness measures have been studied in the literature. Some of them have been used as rewards in RL-based systems and they are also useful for analyzing EDA operations.

- A-INT It is defined for FILTER and GROUP operations as follows:
  **For Group operation:** This score uses conciseness measures which rewards compact group-by views covering many rows as these views are considered informative and easy to understand.
  **For Filter operation:** The Kullback-Leibler (KL) divergence on each column is used to measure how much the filtered data view differs from the unfiltered view. This score favors filter operations whose resultant display deviates significantly from the previous display.

- **Diversity score**: This metric favors a display that highlights parts of the dataset that are different from those seen in any of the previous displays in the session so far. It is calculated as the minimum Euclidean distance between the resultant display of an operation and all previous displays.
- **Readability score** : Highly compact displays with few rows are considered more readable and are given a higher reward.
- **Peculiarity score** : Peculiarity score favors display with a high difference from the initial display at the start of the analysis.
- **Coherence score** : This score is a highly detailed metric determined by a set of handcrafted rules that assign each display a penalty or reward based on whether the operation performed at the current step is coherent with previous operations.

# Problem Setup

The input is a tabular dataset $D$ with a set of attributes (columns) $A$ and the desired output is a sequence of analysis operations that will form an EDA session.

- Statespace
- Action space
- Reward function
- Transition function

Statespace $(S)$ : $s_t \in S$.

$$s_t = \text{Encode}(d_{t-2}) \oplus \text{Encode}(d_{t-1}) \oplus \text{Encode}(d_t),$$

Where $\oplus$ is the vector concatenation operator, $d_t$ is display of data set at time $t$.
Encode $d_t$ is a vector which says about the following:

- For each attribute in the display, we include its entropy, the number of distinct values, and the number of null values.
- For each attribute, we include a feature to determine whether that column is grouped, aggregated, or neither.
- Three global features: number of groups, the mean size of groups, and variance of size of groups.

**Actionspace**($A$): The action space contains four operations with possible parameters that can also be represented in vector form. The operations are:

- GROUP(grpcol, aggcol, aggfunc):
- FILTER(filtercol, filterfunc, filterterm)
- BACK()
- STOP()

**Reward function**($R$): Reward will be calculated after each operation using interestingness measures.
**Transition function:** $P : S \times A \to S$ is a deterministic transition function.

Let $\pi$ be the space of all stationary policies that return a distribution over actions in $A$ given a state $S$.
The goal is to learn an optimal policy $\pi^* \in \pi$ which maximizes the reward for a time horizon $T$

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right],$$

where $s_t \in S$ is the state at time-step $t$ given by $s_t = P(s_{t-1}, a_{t-1})$, $a_t \sim \pi(\cdot|s_t)$, and $\gamma \in [0, 1]$ is the discounting factor.

# Drawbacks of Reward Based Methods

The reward function in RL-based AutoEDA like (ATENA) is:

$$R(s_t, a_t) = r_{int}(s_t, a_t) + \lambda_1 r_{div}(s_t, a_t) + \lambda_2 r_{coh}(s_t, a_t)$$

$\lambda_1, \lambda_2$ are constants that are tuned to calibrate the reward.

- For the calculation of the coherence score, we will find the score based on some rules that are highly data-specific. So, This is one of the drawbacks of RL-based AutoEDA.

- We Analysed some of the Expert EDA sessions. After every operation, we found rewards using predefined interestingness measures. We observed that at each time step different combinations of interestingness measures are optimized. So, in the above reward function, we do not consider all measures and also they are given particular weights at all time steps.

| Action | Measure |
|---|---|
| GROUP highest_layer AGGREGATE COUNT packet_number | A-INT, Diversity |
| BACK | - |
| GROUP eth_src AGGREGATE COUNT packet_number | A-INT, Readability |
| GROUP ip_src AGGREGATE COUNT packet_number | Peculiarity |
| FILTER info_line CONTAINS Echo (ping) reply | Diversity, Coherence |
| BACK | - |
| BACK | - |
| BACK | - |
| FILTER highest_layer NEQ ICMP | Coherence |
| GROUP tcp_srcport AGGREGATE COUNT packet_number | Diversity |
| GROUP ip_src AGGREGATE COUNT packet_number | Coherence, Peculiarity |
| FILTER ip_src NEQ 192.168.1.122 | Coherence |

# Imitation Learning

As seen previously, there are many drawbacks of using Reward based methods for automating the EDA process.

- The goal is to find an optimal policy without knowing the rewards.
- Alternative Techniques:
  - Behavioral Cloning
  - Imitation Learning
  - Inverse Reinforcement Learning (IRL)
- All the above techniques require expert trajectories to learn an optimal policy.

# Cont'd

Generative Adversarial Imitation Learning (GAIL) treats imitation learning as a min-max adversarial problem. Describes two modules - a discriminator D and a policy $\pi$ with opposing goals:

- The goal of policy $\pi(a \mid s)$ for a given state s is to mimic the expert demonstration as closely as possible.
- The goal of discriminator is to differentiate the between (s,a) pairs from $\pi$ and expert demonstration.

# ILAEDA Methodology

We have three neural networks Policy $\pi_\theta$, Value $V_\phi$, and Dicriminator $D_w$ in ILEDA:

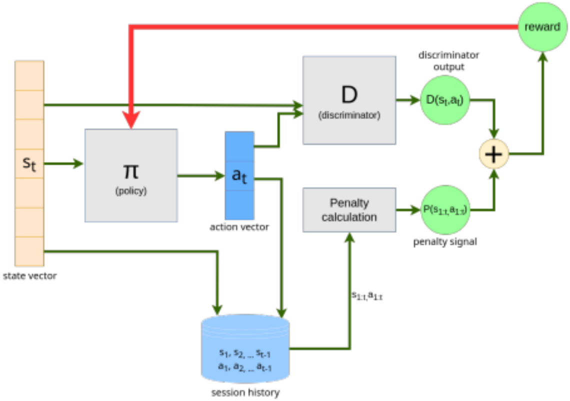


**Figure 1: ILAEDA System Architecture**

- **Initialization**
  Rather than initializing the random parameters in the policy network, we initialize using behavioral cloning since we have few expert demonstrations for imitation learning, pre-training with behavioral cloning is valuable in initializing the policy at a point in parameter space that is likely to converge to the optimum faster.
  During pre-training the objective is:

$$\text{minimize}_\theta \mathbb{E}_{(s,a)\in\mathcal{E}}\left[-\log \pi_\theta(a|s) + \|\theta\|^2\right]$$

- **Discriminator**
  After the policy network generates some action to the given state then it is passed to the discriminator network along with action generated from expert sessions. It tries to classify the generated as fake, and actions from expert sessions as real.
  The discriminator tries to minimize the following loss:

$$\mathcal{L}_D = \mathbb{E}_{\pi_\theta}\left[\log D_w(s,a)\right] + \mathbb{E}_{\pi_E}\left[\log\left(1 - D_w(s,a)\right)\right]$$

- **Reward Calculation with Penalty**

  Reward contains the discriminator term and penalty term:

$$\mathcal{L} = -\log\left(1 - D(s_t, a_t)\right) + P(s_{1:t}, a_{1:t})$$

  There will be some actions that may happen such as:

  - (a) The model takes a BACK action at the beginning of the analysis.
  - (b) The model consecutively repeats the same FILTER/GROUP action.
  - (c) The model alternates between a FILTER/GROUP action and BACK action.

  To discourage these actions, we have added the penalty term to the reward. The penalty term is:

$$P(s_{1:t}, a_{1:t}) = \begin{cases} -1.0 & \text{if } a_t = \text{BACK and } s_t = 1, \\ -1.0 & \text{if } a_t \neq \text{BACK and } a_t = a_{t-1}, \\ -1.0 \times k & \text{if } a_t = a_{t-2} = \cdots = a_{t-2k} = \text{BACK}, \\ & a_{t-2(k+1)} \neq a_{t-2k}, \\ & a_{t-1}, a_{t-3}, \ldots, a_{t-2k-1} \\ & \in \{\text{FILTER, GROUP}\}, k > 1, \\ 0.0 & \text{otherwise.} \end{cases}$$

- **Updating Policy and Parameters Using PPO**

  So, with this reward, we will update the policy parameters in the policy network using Proximal Policy Optimization(PPO).

$$\theta_{t+1} = \arg\max_{\theta} \mathbb{E}_{T \sim B} \left[ \min \frac{\pi_\theta(a|s)}{\pi_{\theta t}(a|s)} \cdot A^t(T), g(\epsilon, A^t(T)) \right]$$

  Where,

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A & \text{if } A \geq 0, \\ (1-\epsilon)A & \text{if } A < 0. \end{cases}$$

  We wish to update parameters $\theta_t$ to $\theta_{t+1}$ such that the expected reward is maximized while making sure that $\pi_{\theta_{t+1}}$ is not too far from $\pi_{\theta_t}$

  Also here, $A^t$ is Advantage function, $T$ are the transitions sampled from environment interaction with policy $\pi_{\theta_t}$

$$A^t(T) = A^t(s, a, r, s') = r + V_{\phi_t}(s') - V_{\phi_t}(s)$$

# Results

| Dataset | Method | Precision | TBLEU-1 | TBLEU-2 | TBLEU-3 | EDA-Sim |
|---|---|---|---|---|---|---|
| CS 1 | BC | 0.2117 | 0.1421 | 0.0826 | 0.0519 | 0.2496 |
| | ATENA | 0.1855 | 0.1855 | 0.1377 | 0.0625 | 0.2704 |
| | ILAEDA | **0.3750** | **0.3333** | **0.2041** | **0.0841** | **0.2950** |
| CS 2 | BC | 0.3287 | 0.1497 | 0.0726 | 0.0497 | 0.3491 |
| | ATENA | 0.2340 | 0.2325 | 0.1873 | **0.1182** | 0.2682 |
| | ILAEDA | **0.4000** | **0.2857** | **0.2182** | 0.1060 | **0.3900** |
| CS 3 | BC | **0.1614** | 0.1235 | 0.0456 | **0.0393** | 0.2728 |
| | ATENA | 0.1153 | 0.1122 | **0.0550** | 0.0320 | 0.2462 |
| | ILAEDA | 0.1429 | **0.1314** | 0.0449 | 0.0359 | **0.3497** |
| CS 4 | BC | 0.1950 | 0.1474 | 0.0771 | 0.0472 | **0.3378** |
| | ATENA | 0.1929 | 0.1929 | 0.1451 | 0.0708 | 0.3017 |
| | ILAEDA | **0.7500** | **0.3333** | **0.2041** | **0.0841** | 0.2051 |
| SD 6 | BC | **0.9167** | 0.1428 | 0.1092 | 0.0612 | 0.2929 |
| | ATENA | 0.1111 | 0.0515 | 0.0173 | 0.0132 | 0.1539 |
| | ILAEDA | 0.4286 | **0.4286** | **0.1816** | **0.0669** | **0.5647** |
| SD 7 | BC | 0.7071 | 0.2190 | 0.1520 | 0.0932 | 0.3505 |
| | ATENA | 0.1111 | 0.0429 | 0.0143 | 0.0114 | 0.1265 |
| | ILAEDA | **0.8333** | **0.5333** | **0.4781** | **0.3852** | **0.5536** |

Figure: Results