Contextual Bandits Data Visualization

with Jupyter Notebooks

RLOS 2020

Quick Introduction

- Name : Milind Agarwal
- Program : Bachelor's/Master's in Computer Science
- School : Johns Hopkins University
- Mentors : Marco Rossi, Paul Mineiro

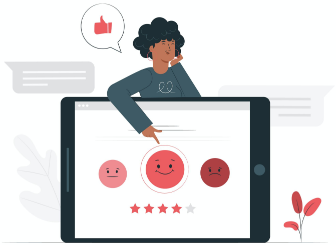
Why do we even need contextual bandit visualizations?

- Good exploratory data analysis + relevant data visualization:
    - Allows you to kickstart your analysis by easily understanding the patterns in your data
    - Helps you inspect/probe policies visually to understand their behaviour

- Visualizations are a powerful yet easily digestible way to understand your data, policies, and results well!

Vectors designed by Freepik

Who may want to use this library?



Customers
A presentable and well-documented visualization library can prove to be useful to customers to better understand their data and policies.

designed by freepik

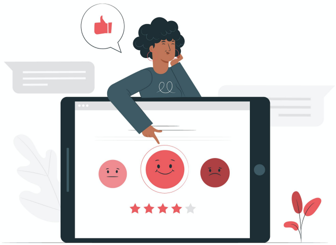
Independent Researchers
Graphics that are easy to understand and use can prove to be helpful for independent researchers exploring contextual bandits through pyVW

Microsoft

MSR Researchers and Investigators
A robust toolkit capable of working with varying Vowpal Wabbit formats can prove to be useful for MSR Researchers for the variety of experiments they run with CB

Vectors designed by Freepik

What considerations should we keep in mind?



designed by freepik

Microsoft

Customers
A presentable and well-documented visualization library can prove to be useful to customers to better understand their data and policies.

Independent Researchers
Graphics that are easy to customize and use can prove to be helpful for independent researchers exploring contextual bandits through pyVW

MSR Researchers and Investigators
A robust toolkit capable of working with varying Vowpal Wabbit formats can prove to be useful for MSR Researchers for the variety of experiments they run with CB

Vectors designed by Freepik

Initially Proposed Goals for RLOS Project

The initial call for proposals for this project contained the following goals:

- Feature importance
- Action distributions
- Action/reward distribution by feature(s) or model used
- Model comparison

Refined Goals and Workflow

1

Exploration

- Feature correlations
- Hierarchical relationships between features
- Prune your dataset to build faster learners

2

Policy Training

- Expected reward
- Comparing different trained policies
- Comparing policy to baseline

3

Feature Importance

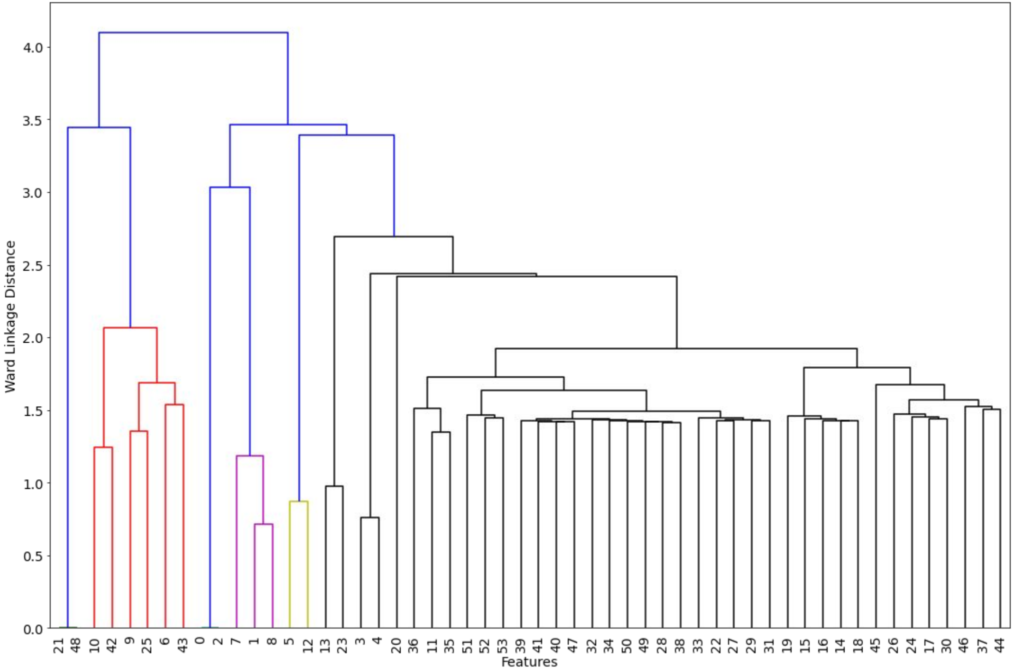
- Compare features importances
- Highlight most important features in a trained policy

4

Action Distributions

- Action set distributions in your policy

Exploration: Correlations and Hierarchy of Features



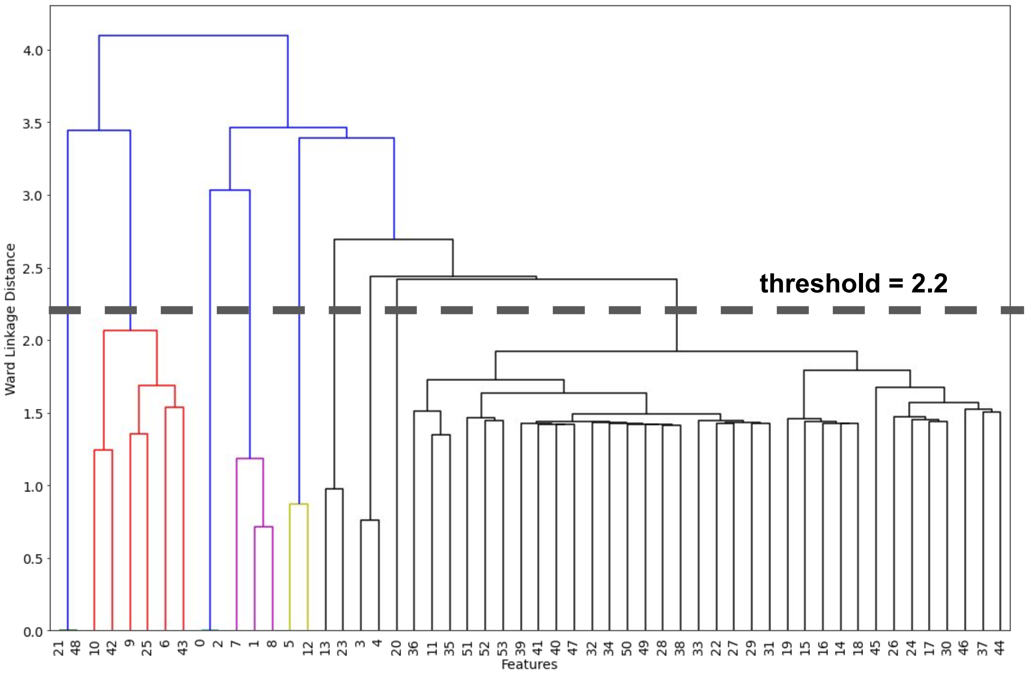
Key Points:

This is a dendrogram (tree-based diagram) illustrating relationships (y-axis) between features (x-axis)

You can use this plot to:

- Visualize hierarchical correlations between your features
- Identify clusters of closely related features
- Prune your dataset based on correlation

Exploration: Pruning Data for Correlations



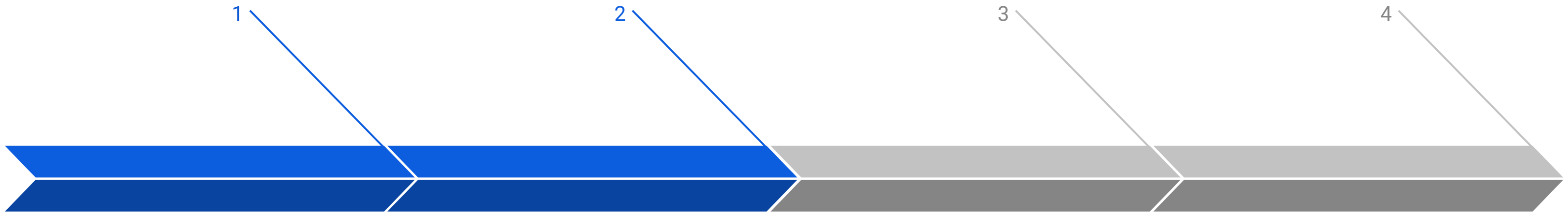
Key Points:

In this graphic, we set an illustrative threshold of 2.2 to prune the dataset's features

This is useful:

- To help policy learn faster by reducing dataset size
- To remove redundant features by clustering

Refined Goals and Workflow



Exploration

- Feature correlations
- Hierarchical relationships between features
- Prune your dataset to build faster learners

Policy Training

- Expected reward
- Comparing different trained policies
- Comparing policy to baseline

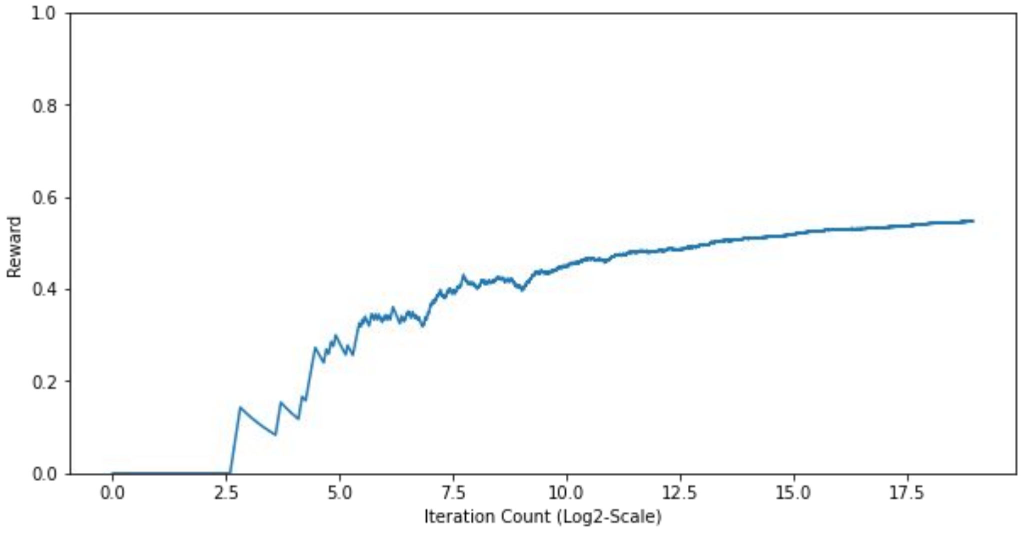
Feature Importance

- Compare features importances
- Highlight most important features in a trained policy

Action Distributions

- Action set distributions in your policy

Policy Training: Expected Reward

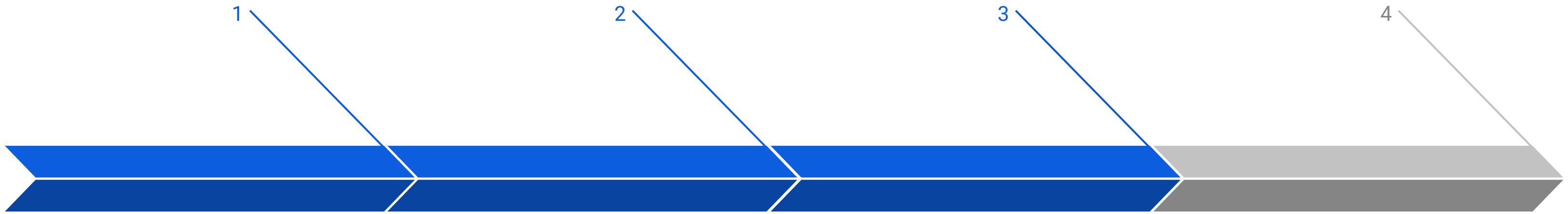


Key Points:

With this visualization, you can look at expected reward for your policy during training
This is useful:

- To gauge example size that allows your policy to learn well enough
- To highlight variation in the policy's learning as it sees more examples

Refined Goals and Workflow



Exploration

- Feature correlations
- Hierarchical relationships between features
- Prune your dataset to build faster learners

Policy Training

- Expected reward
- Comparing different trained policies
- Comparing policy to baseline

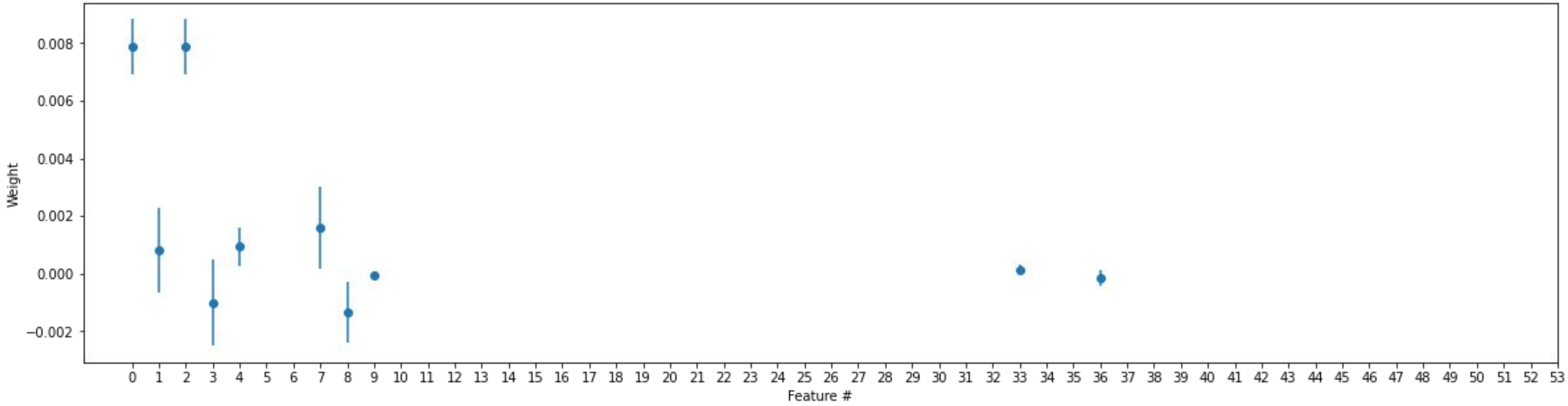
Feature Importance

- Compare features importances
- Highlight most important features in a trained policy

Action Distributions

- Action set distributions in your policy

Feature Importance: Compare Feature Importances



Key Points:
This feature importance chart shows importance scores along with deviation (y-axis) for all features in the dataset (x-axis). It uses the permutation-based algorithm. By using this algorithm:

- You introduce noise
- You corrupt the value of that feature for each example
- The feature can no longer provide valuable information to the policy

This allows us to measure importance of that feature.

Feature Importance: Permutation-Based Metrics

	0	1	2	3	4	5	6
0	2596.0	51.0	5192.0	258.0	0.0	-5.0	221.0
1	2590.0	56.0	5180.0	212.0	-6.0	-5.0	220.0
2	2804.0	139.0	5608.0	268.0	65.0	-5.0	234.0
3	2785.0	155.0	5570.0	242.0	118.0	-5.0	238.0
4	2595.0	45.0	5190.0	153.0	-1.0	-5.0	220.0

Key Points:

By shuffling the values of a feature:

- You introduce noise
- You corrupt the value of that feature for each example
- The feature can no longer provide valuable information to the policy

Feature Importance: Multicollinearity Caveat

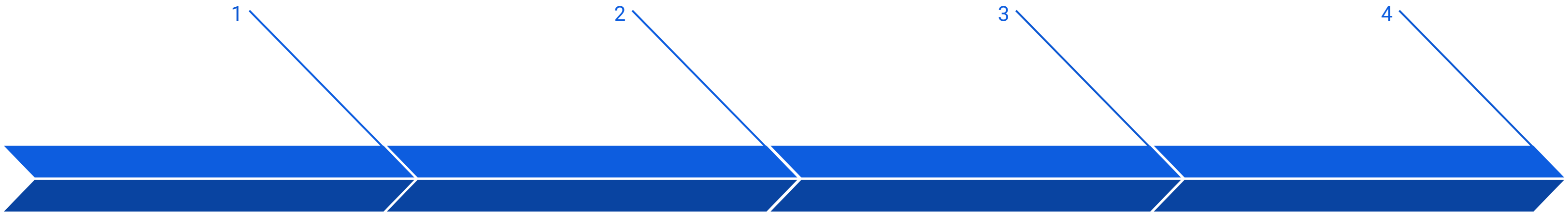
Key points:

Let’s say features 2 and 4 are correlated with each other.
Feature 2 is shuffled → policy uses feature 4 for information → feature 2 **not important!**
Feature 4 is shuffled → policy uses feature 2 for information → feature 4 **not important!**

	0	1	2	3	4	5	6
0	2596.0	51.0	5192.0	258.0	0.0	-5.0	221.0
1	2590.0	56.0	5180.0	212.0	-6.0	-5.0	220.0
2	2804.0	139.0	5608.0	268.0	65.0	-5.0	234.0
3	2785.0	155.0	5570.0	242.0	118.0	-5.0	238.0
4	2595.0	45.0	5190.0	153.0	-1.0	-5.0	220.0

	0	1	2	3	4	5	6
0	2596.0	51.0	5192.0	258.0	0.0	-5.0	221.0
1	2590.0	56.0	5180.0	212.0	-6.0	-5.0	220.0
2	2804.0	139.0	5608.0	268.0	65.0	-5.0	234.0
3	2785.0	155.0	5570.0	242.0	118.0	-5.0	238.0
4	2595.0	45.0	5190.0	153.0	-1.0	-5.0	220.0

Refined Goals and Workflow



Exploration

- Feature correlations
- Hierarchical relationships between features
- Prune your dataset to build faster learners

Policy Training

- Expected reward
- Comparing different trained policies
- Comparing policy to baseline

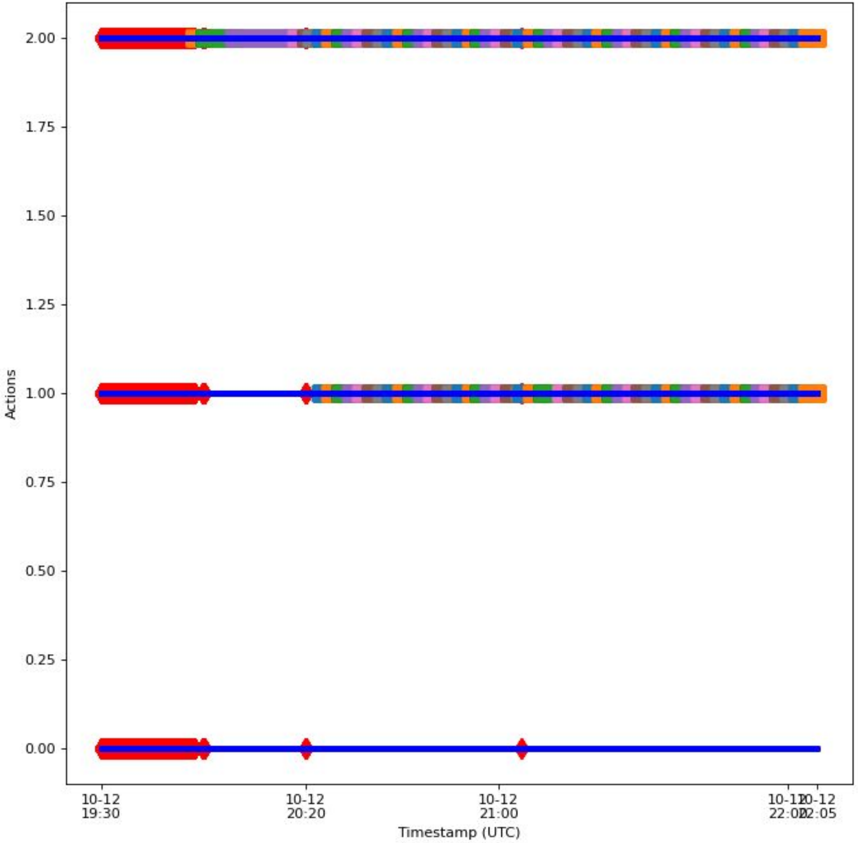
Feature Importance

- Compare features importances
- Highlight most important features in a trained policy

Action Distributions

- Action set distributions in your policy

Action Distribution: Action Visualizations



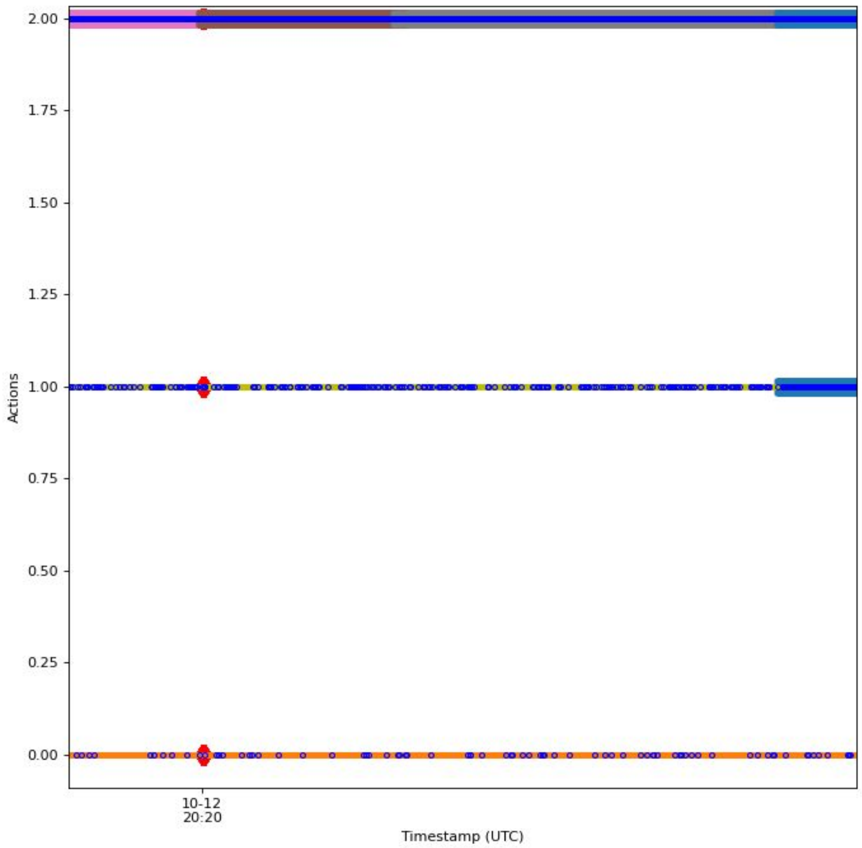
Key Points:

With this visualization, you are looking at action distributions (y-axis) and action-related metadata (colors) across all your training examples (x-axis)

Scheme:

- Red diamonds: Policy uninstantiated
- Blue: Action available in action set
- Colored squares: model available, colored based on reward

Action Distribution: Action Visualizations



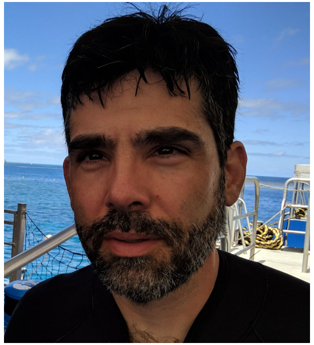
Key Points:

You can also zoom into this graphic and look at more fine-grained information.

Now, we can also see:

- Blue circles: Exact instances of reward
- Orange: colored action (baseline)

Thank you!



Marco Rossi **Paul Mineiro** **Jack Gerrits**

Contextual Bandits Data Visualization

with Jupyter Notebooks

RLOS 2020