AAAI 2023 Optimization with Constraint Learning Lab
Part IV: DOFramework

Orit Davidovich[1]

[1]IBM Research Lab, Haifa

AAAI 2023

Overview

1. Decision Optimization
2. DOFramework
3. Profiling
4. Design
5. Deployment

Mathematical Decision Optimization (DO)

Mathematical Decision-Optimization (DO) Model $M(\mathbf{w})$ (*):

$$
\begin{array}{rl}
\mathbf{x}^*(\mathbf{w}) \in \arg\min_{\mathbf{x}\in\mathbb{R}^n, \mathbf{y}\in\mathbb{R}^m} & f(\mathbf{x}, \mathbf{y}, \mathbf{w}) \\
\text{s.t.} & \mathbf{g}(\mathbf{x}, \mathbf{y}, \mathbf{w}) \le \mathbf{0} \\
 & \mathbf{y} = \mathbf{h}(\mathbf{x}, \mathbf{w}) \\
 & \mathbf{x} \in \Omega(\mathbf{w})
\end{array}
$$

$\mathbf{w} \in \mathbb{R}^k$ – fixed uncontrollable.
$\Omega(\mathbf{w})$ – polytope (possibly unbounded).

Mathematical Decision Optimization (DO)

Learned DO Model $\widehat{M}(\mathbf{w})$:

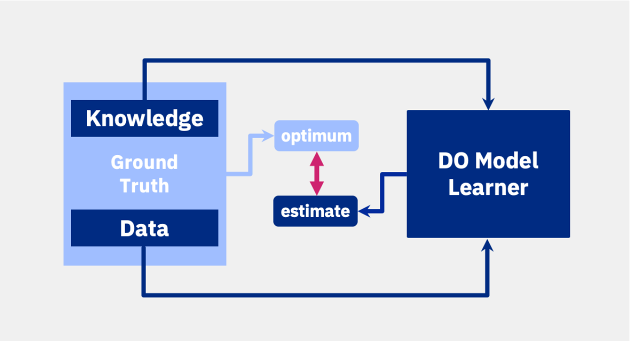
$$
\begin{aligned}
\widehat{\mathbf{x}}(\mathbf{w}) \in \arg\min_{\mathbf{x}\in\mathbb{R}^n,\mathbf{y}\in\mathbb{R}^m} \quad & \widehat{f}(\mathbf{x},\mathbf{y},\mathbf{w}) && \Longleftarrow \text{learn} \\
\text{s.t.} \quad & \mathbf{g}(\mathbf{x},\mathbf{y},\mathbf{w}) \leq \mathbf{0} \\
& \mathbf{y} = \widehat{\mathbf{h}}(\mathbf{x},\mathbf{w}) && \Longleftarrow \text{learn} \\
& \mathbf{x} \in \Omega(\mathbf{w})
\end{aligned}
$$

$\mathbf{w} \in \mathbb{R}^k$ – fixed uncontrollable.
$\Omega(\mathbf{w})$ – polytope (possibly unbounded).

A **DO model learner** is an algorithm, routine, or pipeline that incorporates learning and knowledge into a DO model that integrates both. Its output is a learned DO model.

DOFramework

DOFramework: a testing framework for DO model learners.



Knowledge Ω	Bounded convex polytope $\Omega \subseteq \mathrm{dom}\,(f) \subseteq \mathbb{R}^d$, $d = n + m + k \geq 2$.
Ground Truth f	Continuous PWL f with (combinatorially) known $\mathbf{x}^* \in \arg\min_{\mathbf{x}\in\Omega} f(\mathbf{x})$.
Data D	Gaussian mix model in $\mathrm{dom}\,(f)$.

Profiling

DO Problem Instance: $(f, \Omega, D, \mathbf{x}^*)$

Estimate solution $\hat{\mathbf{x}}$ score:

$$\text{score}(\hat{\mathbf{x}}) = \frac{f(\hat{\mathbf{x}}) - f(\mathbf{x}^*)}{f_{\text{max}} - f_{\text{min}}}$$

Profiling

DO Problem Instance: $(f, \Omega, D, \mathbf{x}^*)$

Estimate solution $\hat{\mathbf{x}}$ score:

$$\text{score}(\hat{\mathbf{x}}) = \frac{f(\hat{\mathbf{x}}) - f(\mathbf{x}^*)}{f_{\text{max}} - f_{\text{min}}} \quad \Longrightarrow \text{score density}$$

Profiling

DO Problem Instance: $(f, \Omega, D, \mathbf{x}^*)$

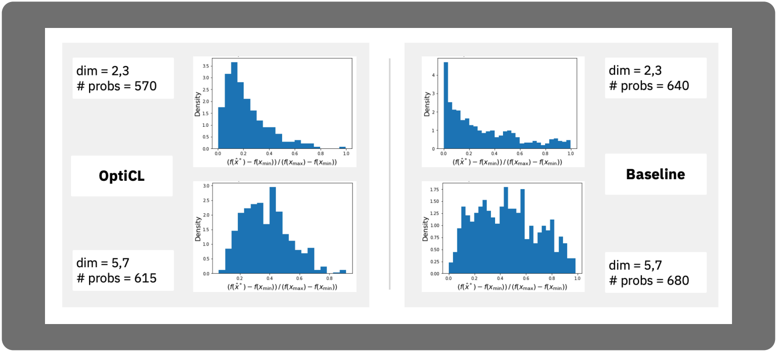
Estimate solution $\hat{\mathbf{x}}$ score:

$$\text{score}(\hat{\mathbf{x}}) = \frac{f(\hat{\mathbf{x}}) - f(\mathbf{x}^*)}{f_{\text{max}} - f_{\text{min}}} \quad \Longrightarrow \text{score density}$$

Solution Quality Probability:

$$Pr[f(\hat{\mathbf{x}}) - f(\mathbf{x}^*) < \epsilon(f_{\text{max}} - f_{\text{min}})]$$

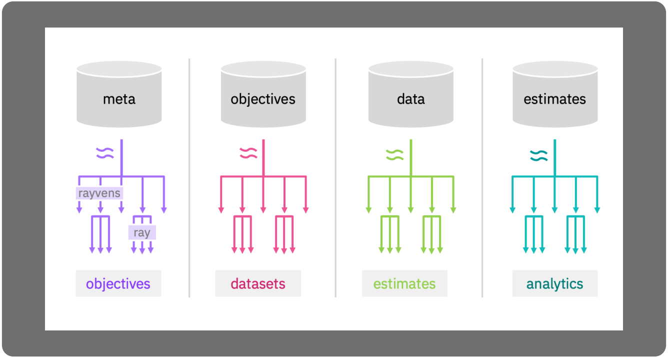
Profiling



Demo: Go to `ocl_lab/DOFramework/profile.ipynb`

– Mitchell, OSullivan, Dunning. *PuLP: A Linear Programming Toolkit for Python* (2011).

Design



– Moritz et. al.. *Ray: A Distributed Framework for Emerging AI Applications* (OSDI 2018). `https://github.com/ray-project/ray`.

– Bercea, Tardieu. *Rayvens: Event sources and sinks on Ray*, `https://github.com/project-codeflare/rayvens`

Deployment

Requirements:

- Storage: Local / Cloud Object Storage (~/project_folder/configs.yaml)
- Compute: Local / K8S Cluster (AWS / IBM Cloud)

Installation:

```
$ pip install doframework
$ cd ~/project_folder
$ doframework-setup.sh --configs configs.yaml
```

Running:

```
$ python user_module.py

$ ray submit doframework.yaml user_module.py
```

user_module.py

```python
import doframework as dof

@dof.resolve
def ocl(data: np.array, constraints: np.array, **kwargs):
    ...
    return optimal_arg, optimal_val, regression_model

if __name__ == '__main__':
    ...
    dof.run(ocl, 'configs.yaml', objectives=5, datasets=3)
```

Demo: Go to `ocl_lab/DOFramework/example.ipynb`

input.json

```
{
    "f": {
        "vertices": {
            "num": 7,
            "range": [[5.0,20.0],[0.0,10.0]],
        },
        "values": {
            "range": [0.0,5.0]
        },
    },
    "omega" : {
        "ratio": 0.8
    },
    "data" : {
        "N": 750,
        "policy_num": 2,
        "scale": 0.4,
        "noise": 0.01
    },
    "input_file_name": "input.json"
}
```