Combinatorial Optimization Augmentede Machine Learning

Learning by Imitation

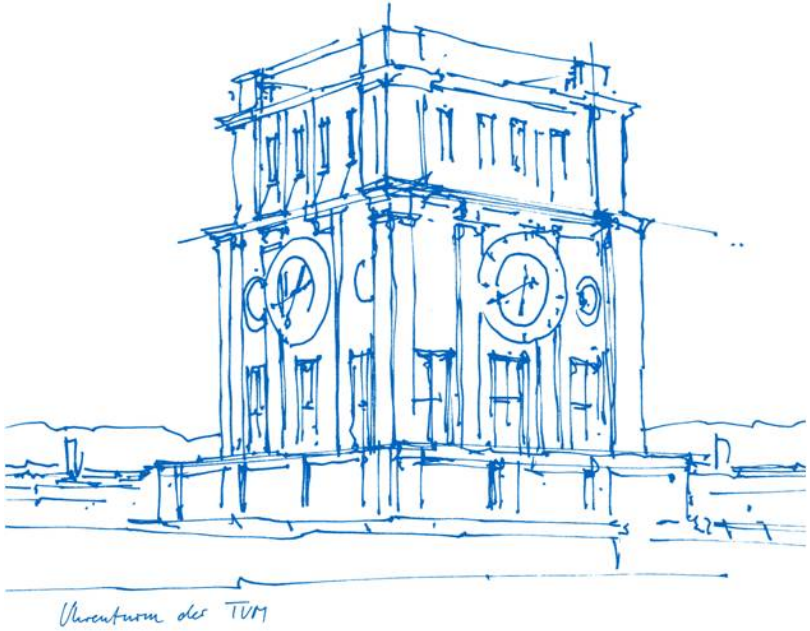
Prof. Dr. Maximilian Schiffer

Professorship for Business Analytics and Intelligent Systems

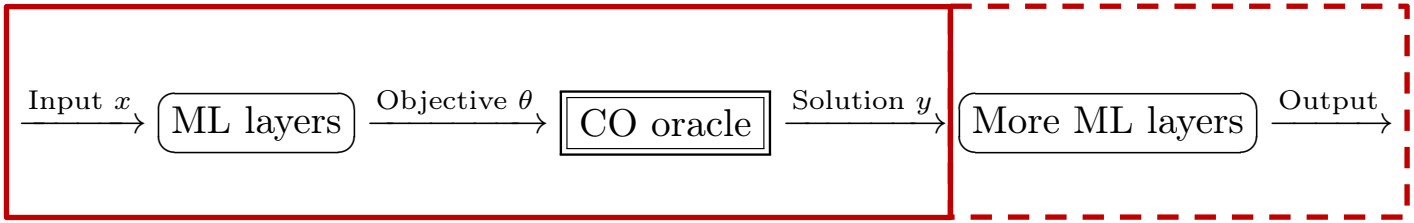
School of Management

Munich Data Science Institute

Technical University of Munich



A general CO-augmented ML Pipeline



ML layers

- Consider your favorite statistical model, e.g., a generalized linear model, your favorite neural network

CO oracle

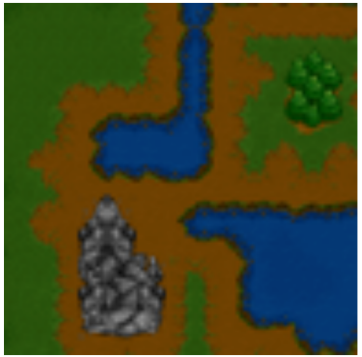
- No limits: a shortest path algorithm, Gurobi / Cplex, column generation, your favorite metaheuristic

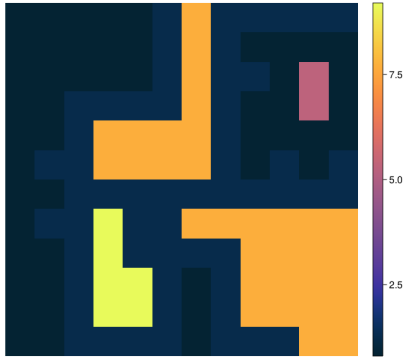
Such a pipeline allows us to leverage the predictive / anticipative power of ML as well as the combinatorial power of CO algorithms

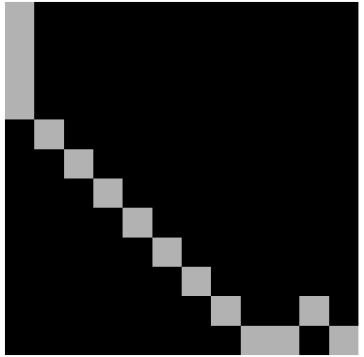
A specific example (that will haunt us today and tomorrow…)

Computing paths in unknown areas

- We aim to find shortest paths on a map without access to information about the underlying terrain







Map image → Conv. Neural Network → Cell costs → Dijkstra's algorithm → Shortest path

Training a CO-augmented ML-Pipeline

$$\xrightarrow{\text{Input } x} \boxed{\text{ML layers}} \xrightarrow{\text{Objective } \theta} \boxed{\boxed{\text{CO oracle}}} \xrightarrow{\text{Solution } y}$$

Learning by experience

- We employ this morning‘s RL paradigm and use a reward signal to iteratively improve on the ML layer

End-to-end learning

- Derive a loss function that allows to compute a gradient over our ML-CO pipeline

Which one is your favorite ?

End-to-end Learning – Our Agenda for the next hours

$$\xrightarrow{\text{Input } x} \boxed{\text{ML layer } \varphi_w} \xrightarrow{\text{Objective } \theta = \varphi_w(x)} \boxed{\boxed{\text{CO oracle } f}} \xrightarrow{\text{Solution } y = f(\theta)} \boxed{\text{Loss function } \mathcal{L}}$$

Main Problems

- What is a good loss function $\min_w \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\Big(\overbrace{f\big(\underbrace{\varphi_w(x^{(i)})}_{\theta^{(i)}}\big)}^{y^{(i)}}, \dots \Big)$?
- Even when we define a useful loss, f is non-differentiable, which deprives us from applying automatic differentiation to learn varphi_w

We will understand how to design such a pipeline by…

- … taking a very general perspective on f…
- … transitioning to a distributional understanding of it…
- … defining a proper loss function…
- … and the respective learning paradigm.

A very generic view on CO problems

Problem 1: CO-layer

$$f : \theta \longmapsto \operatorname*{argmax}_{v \in \mathcal{V}} \theta^\top v$$

- $\theta \in \mathbb{R}^d$ Objective direction
- $\mathcal{V} \subset \mathbb{R}^d$ (finite) set of feasible solutions (note that precisely: $\mathcal{V}(x) \subset \mathbb{R}^{d(x)}$)

Assuming our objective to be linear in θ, we could also study:

$$\operatorname*{argmax}_{v \in \mathcal{V}} \theta^\top g(v) + h(v)$$

We keep $g(v) = v$ and $h(v) = 0$ for ease of exposition

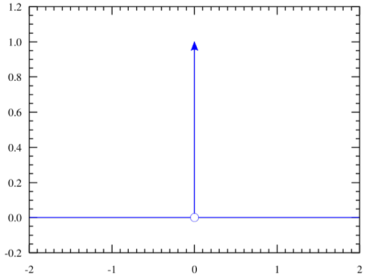
Problem 1 is very generic, it can be any LP or ILP (when the feasible set is bounded)

A probabilistic perspective

F as a distribution

- Instead of thinking of $f(\theta)$ returning a single element of the feasible set, treat f as a distribution $p(\cdot|\theta)$
- A natural choice would be a dirac mass

$$p(v|\theta) = \delta_{f(\theta)}(v)$$



Still not differentiable

Our new goal:

- We aim to derive an approximation of p such that

$$\theta \longmapsto \widehat{p}(\cdot|\theta)$$

A probabilistic perspective

Approximating p

- We aim to derive an approximation of p such that

$$\theta \longmapsto \widehat{p}(\cdot|\theta)$$

  is smooth with respect to θ

- If we succeed in finding such a $\widehat{p}$ than the following (expectation) mapping will also be smooth

$$\widehat{f} : \theta \longmapsto \mathbb{E}_{\widehat{p}(\cdot|\theta)}[V] = \sum_{v \in \mathcal{V}} v\widehat{p}(v|\theta)$$

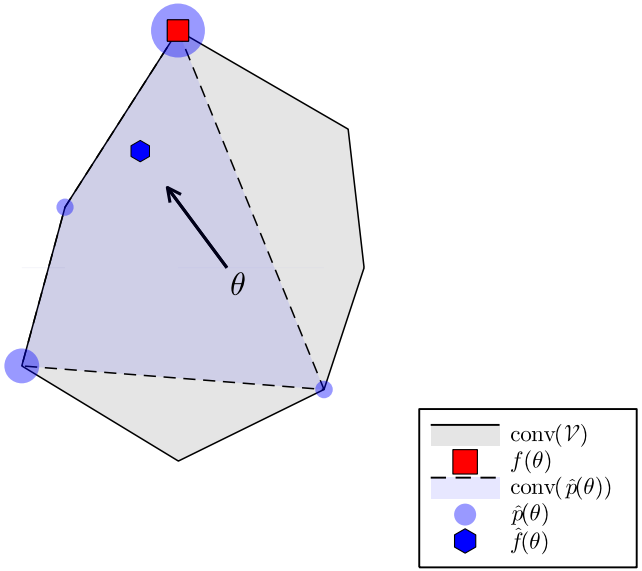
If we find such a $\widehat{p}$, we obtain a probabilistic CO-layer that is smooth

A probabilistic perspective

Understanding a probabilistic CO-layer

Probabilistic CO-layers work in two steps:

1. Construct a probability distribution $\widehat{p}(\cdot|\theta) \in \Delta^{\mathcal{V}}$
2. Return its expectation $\widehat{f}(\theta) = \mathbb{E}_{\widehat{p}(\cdot|\theta)}[V] \in \mathrm{conv}(\mathcal{V})$



In the following we impose some conditions:

1. All computations can be done by querying f
2. The expectation must be tractable
3. The mapping $\theta \longmapsto \widehat{p}(\cdot|\theta)$ needs to be differentiable

To achieve this, we need to spread out the (naive) dirac mass to multiple vertices

Regularization

Problem 2: regularized CO-layer

$$\widehat{f}_\Omega : \theta \longmapsto \underset{\mu \in \mathrm{dom}(\Omega)}{\mathrm{argmax}} \; \theta^\top \mu - \Omega(\mu)$$

- $\Omega : \mathbb{R}^d \to \mathbb{R}$: a smooth and convex function
- We enforce $\Omega(\mu) = +\infty$ whenever $\mu \notin \mathrm{conv}(\mathcal{V})$
- $\mathrm{dom}(\Omega) \subseteq \mathrm{conv}(\mathcal{V})$
- Recall that mu is the expectation of some distribution over V

We can rewrite $\widehat{f}_\Omega(\theta)$ as a convex combiantion of elements of V

Creating a distribution via additive perturbations

Additive Perturbations

$$\widehat{f}_\varepsilon^+(\theta) = \mathbb{E}\left[\operatorname*{argmax}_{v\in\mathcal{V}}(\theta+\varepsilon Z)^\top v\right] = \mathbb{E}\left[f(\theta+\varepsilon Z)\right]$$

- We can use stochastic perturbation to create a distribution over V in an additive fashion with
  - $\varepsilon > 0$ being the perturbation amplitude, and
  - $Z \sim \mathcal{N}(0, I)$ being a standard gaussian vector
- We can explicitly describe the resulting distribution:

$$\widehat{f}_\varepsilon^+(\theta) = \sum_{v\in\mathcal{V}} v\widehat{p}_\varepsilon^+(v|\theta) \qquad \text{with} \qquad \widehat{p}_\varepsilon^+(v|\theta) = \mathbb{P}\left(f(\theta+\varepsilon Z) = v\right)$$

Technical excursus – why did we do alll this math?!

Proposition 1

$$\nabla_\theta \widehat{p}_\varepsilon^+(v|\theta) = \frac{1}{\varepsilon}\mathbb{E}\left[\mathbb{1}\{f(\theta+\varepsilon Z)=v\}Z\right]$$

$$J_\theta \widehat{f}_\varepsilon^+(\theta) = \frac{1}{\varepsilon}\mathbb{E}\left[f(\theta+\varepsilon Z)Z^\top\right]$$

Proposition 2

Via convex conjugation, we get with $F_\varepsilon^+(\theta) = \mathbb{E}\left[\max_{v\in\mathcal{V}}(\theta+\varepsilon Z)^\top v\right]$ and $\Omega_\varepsilon^+ = (F_\varepsilon^+)^*$ that

$$\widehat{f}_\varepsilon^+(\theta) = \operatorname*{argmax}_{\mu\in\mathrm{conv}(\mathcal{V})} \theta^\top\mu - \Omega_\varepsilon^+(\mu) = \widehat{f}_{\Omega_\varepsilon^+}(\theta).$$

Creating a distribution via multiplicative perturbations

Additive Perturbations

$$\widehat{f}_{\varepsilon}^{+}(\theta) = \mathbb{E}\left[\operatorname*{argmax}_{v \in \mathcal{V}} (\theta + \varepsilon Z)^{\top} v\right] = \mathbb{E}\left[f(\theta + \varepsilon Z)\right]$$

Can you think of cases where an additive perturbation fails?

Problem of Additive Perturbations

- Some components of $\theta + \varepsilon Z$ may switch their sign
- Can be problematic if our oracle‘s behavior is sign dependent (e.g., dijkstra)
- Ways to preserve the sign of theta: changing the distribution of Z vs. Changing the perturbation scheme

Creating a distribution via multiplicative perturbations

Multiplicative Perturbations

$$\widehat{f}_{\varepsilon}^{\odot}(\theta) = \mathbb{E}\left[\underset{v \in \mathcal{V}}{\operatorname{argmax}} \left(\theta \odot e^{\varepsilon Z - \varepsilon^2 \mathbf{1}/2}\right)^{\top} v\right] = \mathbb{E}\left[f\left(\theta \odot e^{\varepsilon Z - \varepsilon^2 \mathbf{1}/2}\right)\right]$$

- With $\odot$ Being the Hadamard product
- And $-\varepsilon^2 \mathbf{1}/2$ being a correction term
  - Necessary since $\mathbb{E}[e^{\varepsilon Z}] = e^{\varepsilon^2 \mathbf{1}/2} \neq 1$ $\quad (\mathbb{E}[\theta \odot e^{\varepsilon Z - \varepsilon^2 \mathbf{1}/2}] = \theta)$

We can formally show the equivalance of using multiplictive perturbations to regularization, using a similar argument as in the additive case

Imitation learning: an end-to-end loss function

E-2-E loss function

- Assume that for every sample $x^{(i)}$, we have a target $\bar{t}^{(i)}$
- Then we seek for a loss function of the form

$$\min_{w} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\Big(f\big(\varphi_w(x^{(i)})\big), \bar{t}^{(i)}\Big)$$

- taking the optimization layer into account

Imitation Learning perspective

$$\mathcal{L}^{\text{aux}}_{\ell,\Omega}(\theta, \bar{t}, y) = \underbrace{\ell(y, \bar{t})}_{\text{base loss}} + \underbrace{\theta^\top (y - \bar{y})}_{\substack{\text{gap between} \\ y \text{ and } \bar{y} \text{ for the} \\ \text{CO problem } (2)}} - \underbrace{(\Omega(y) - \Omega(\bar{y}))}_{\text{regularization term}}$$

Imitation learning: an end-to-end loss function

In other words…

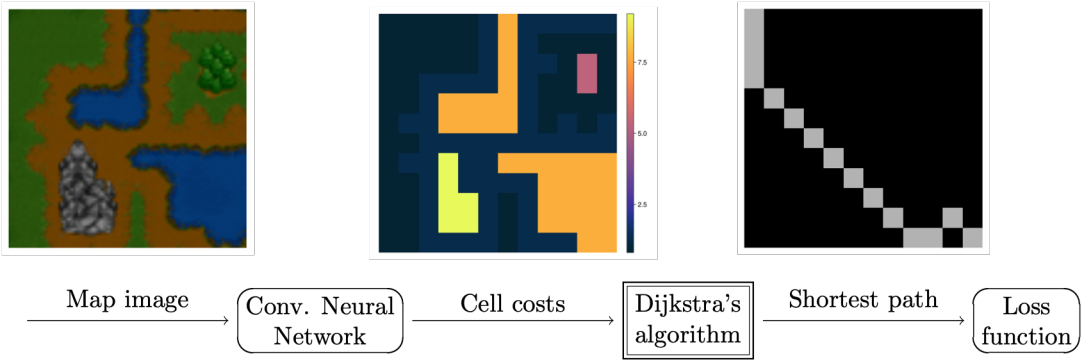
$$\mathcal{L}^{\text{aux}}_{\ell,\Omega}(\theta, \bar{t}, y) = \underbrace{\ell(y, \bar{t})}_{\text{base loss}} + \underbrace{\left(\theta^\top y - \Omega(y)\right) - \left(\theta^\top \bar{y} - \Omega(\bar{y})\right)}_{\substack{\text{gap between } y \text{ and } \bar{y} \\ \text{for the regularized CO problem } (9)}}$$

Structured Learning via Fenchel-Young Losses

$$\mathcal{L}^{\text{FY}}_{\Omega}(\theta, \bar{y}) = \max_{y \in \text{conv}(\mathcal{V})} \left(\theta^\top y - \Omega(y)\right) - \left(\theta^\top \bar{y} - \Omega(\bar{y})\right)$$

- replace the regularization by your favorite perturbation mechanism (additive or multiplicative)

Hands on…

Let us test things out by…

- Loading the following notebook and data: https://github.com/BatyLeo/TP_Warcraft
- Listening to Gerhard for some Julia basics, and specifics related to the notebook (-:
- Start working on the notebook, we will make breaks along the way to catch up and discuss