Constrained ML via Data Projection

To be prepared is half of the victory

Let's Face It

9 times out of 10, we are dealing with supervised learning

Meaning a basic training problem is in the form:

$$\underset{\theta}{\operatorname{argmin}}\{L(\hat{y}, y) \mid \hat{y} = f(x;\theta)\}$$

- Where the ground truth vector $\boldsymbol{y} = \{y_i\}_{i=1}^{m}$ appears in the loss expression

Let's Face It

9 times out of 10, we are dealing with supervised learning

Meaning a basic training problem is in the form:

$$\operatorname*{argmin}_{\theta} \{ L(\hat{y}, y) \mid \hat{y} = f(x; \theta) \}$$

- Where the ground truth vector $y = \{y_i\}_{i=1}^{m}$ appears in the loss expression

The corresponding constrained version is:

$$\operatorname*{argmin}_{\theta} \{ L(\hat{y}, y) \mid \hat{y} = f(x; \theta), g(\hat{y}) \leq 0 \}$$

- The presence of the constraints usually contrasts with the loss function
- ...Resulting in a change in the optimal parameter vector

But what if it doesn't?

Redundant Constraints

Let's assume the constraint are aligned with the loss

In this case, we can expect:

$$\operatorname*{argmin}_{\theta} L(\hat{y}, y) \simeq \operatorname*{argmin}_{\theta} \{ L(\hat{y}, y) \mid g(\hat{y}) \leq 0 \} \qquad \text{with: } \hat{y} = f(x; \theta)$$

- Since the constraints are somewhat redundant
- ...We can expect the loss function to correlate with constraint violation

As a result, the two optimal parameter vector should be similar

In practice, solving the unconstrained problem approximately solves the constrained one

Enforcing Constraints via Preprocessing

Based on this idea, we can try to enforce the constraints via pre-processing

First, we compute an adjusted target vector:

$$z^* = \operatorname*{argmin}_z \{ L(z, y) \mid g(z) \leq 0 \}$$

...Then we solve an unconstrained supervised learning problem:

$$\theta^* = \operatorname*{argmin}_\theta \{ L(\hat{y}, z^*) \mid \hat{y} = f(x; \theta) \}$$

In terms of accuracy, we can expect θ^* to be a good parameter vector

- In particular, if L is a metric or quasimetric
- ...Then we have $L(f(x; \theta^*), y) \leq L(f(x; \theta^*), z^*) + L(z^*, y)$

Enforcing Constraints via Preprocessing

Based on this idea, we can try to enforce the constraints via pre-processing

First, we compute an adjusted target vector:

$$z^* = \operatorname*{argmin}_z \{ L(z, y) \mid g(z) \leq 0 \}$$

...Then we solve an unconstrained supervised learning problem:

$$\theta^* = \operatorname*{argmin}_\theta \{ L(\hat{y}, z^*) \mid \hat{y} = f(x; \theta) \}$$

In terms of constraint satisfaction:

- z^* will satisfy the constrains by construction
- $f(x; \theta^*)$ will be approximately feasible iff L is correlated with g

Preprocessing and Projection

It's useful to inspect our preprocessing step in detail

$$z^* = \operatorname*{argmin}_z \{ L(z, y) \mid g(z) \leq 0 \}$$

- This can be understood as a form of projection
- ...And compared with the proximal operator for the indicator function $\boldsymbol{I_g}$

$$\mathbf{prox}_g(y) = \operatorname*{argmin}_z \{ I_g(z) + \|z - y\|_2^2 \}$$

The main difference is the use of $\boldsymbol{L}$ on one side and $\| \cdot \|_2^2$ on the other

- In case of an MSE loss, there is no difference at all!
- For other loss functions, we still have a strong analogy

For this reason, we'll call this approach "data projection"

Analysis of the Approach

Unlike in training-time Lagrangians, here there's an inherent approximation

- So, it's difficult to provide guarantees
- What we can expect is approximate constraint satisfaction

Analysis of the Approach

Unlike in training-time Lagrangians, here there's an inherent approximation

- So, it's difficult to provide guarantees
- What we can expect is approximate constraint satisfaction

The approach makes use of a Monte-Carlo approximation

- So be on the lookout for overfitting issues
- ...And always check constraint satisfaction on the training set

Analysis of the Approach

Unlike in training-time Lagrangians, here there's an inherent approximation

- So, it's difficult to provide guarantees
- What we can expect is approximate constraint satisfaction

The approach makes use of a Monte-Carlo approximation

- So be on the lookout for overfitting issues
- ...And always check constraint satisfaction on the training set

The computational effort should be considered

- We need to solve a large scale constrained optimization problem
- ...But one that is defined entirely in target space
- ...And therefore typically with a nice structure

Analysis of the Approach

The approach is well suited to deal with relational constraints

- This is the case since we can access (in theory) all data at the same time
- The best target are distribution constraints
- ...Since for them approximate satisfaction is typically the best we can do

Analysis of the Approach

The approach is well suited to deal with relational constraints

- This is the case since we can access (in theory) all data at the same time
- The best target are distribution constraints
- ...Since for them approximate satisfaction is typically the best we can do

Data projection has wide compatibility

- Since we modify directly the dataset
- ...We can then use any supervised learning technique

Projection for our Fairness Case Study

We have a regression problem and our loss function is the MSE

$$\operatorname*{argmin}_{\theta} \|z - y\|_2^2$$

- z is the projected target vector, y the original one

Projection for our Fairness Case Study

We have a regression problem and our loss function is the MSE

$$\mathop{\text{argmin}}_{\theta} \|z - y\|_2^2$$

- z is the projected target vector, y the original one

We have a constratint on the DIDI index, with a single protected attribute

...Which can be understood as the sum of multiple deviation

$$\sum_{v \in D} \left| \frac{1}{m} \sum_{i=1}^{m} z_i - \frac{1}{|X_v|} \sum_{i=1}^{m} X_{v,i} z_i \right| \leq \varepsilon$$

- Every protected group is associated to a value v of the sensitive attribute
- Every entry $X_{v,i}$ is 1 iff the i-th example is part of group v

Projection for our Fairness Case Study

The objective can be stated directly, but the constraint takes more effort

We start by viewing the DIDI as a sum of deviations:

$$\sum_{v \in D} d_v \leq \varepsilon$$

- d_v represents the L1 norm between the average target for group v
- ...And the overal target average

Projection for our Fairness Case Study

The objective can be stated directly, but the constraint takes more effort

We start by viewing the DIDI as a sum of deviations:

$$\sum_{v \in D} d_v \leq \varepsilon$$

- d_v represents the L1 norm between the average target for group v
- ...And the overal target average

So it makes sense to model such averages through additional variables

$$\bar{z} = \frac{1}{m} 1^T z \quad \text{and} \quad \bar{z}_v = \frac{1}{\|X_v\|_1} X_v z \quad \forall v \in D$$

Both averages are defined by linear expressions

Projection for our Fairness Case Study

Based on the averages, we can model the L1 norms

$$d_v = |\bar{z}_v - \bar{z}| \quad \forall v \in D$$

There are non-linear expression, but can be linearized as:

$$d_v \geq \bar{z}_v - \bar{z} \quad \forall v \in D$$
$$d_v \geq -\bar{z}_v + \bar{z} \quad \forall v \in D$$

- The inequalities do not defined $\boldsymbol{d_v}$ univocally
- ...Rather, the only provide a lower bound

Projection for our Fairness Case Study

Based on the averages, we can model the L1 norms

$$d_v = |\bar{z}_v - \bar{z}| \quad \forall v \in D$$

There are non-linear expression, but can be linearized as:

$$d_v \geq \bar{z}_v - \bar{z} \quad \forall v \in D$$
$$d_v \geq -\bar{z}_v + \bar{z} \quad \forall v \in D$$

- The inequalities do not defined d_v univocally
- ...Rather, the only provide a lower bound

But that is still enough!

- Due to our constrain, keeping d_v is always preferable
- ...Meaning that optimization itself will keep the constraints tight

Projection for our Fairness Case Study

Overall, the projection problem can be formulated as:

$$\operatorname*{argmin}_{\hat{y}} \|y - z\|_2$$

$$\begin{aligned}
\text{subject to: } \bar{y} &= \frac{1}{m}\mathbf{1}^T z \\
\bar{y}_v &= \frac{1}{\|X_v\|_1} X_v z && \forall v \in D \\
d_v &\geq \bar{y} - \bar{y}_v && \forall v \in D \\
d_v &\geq -(\bar{y} - \bar{y}_v) && \forall v \in D \\
\sum_{v \in D} d_v &\leq \varepsilon
\end{aligned}$$

...Which is a quadratic program