Feature-Incremental ML

Yubo Cai, Vassilis Digalakis Jr, Michael Li

October 2023

Feature-Incremental Machine Learning

Assume that we have a set of features $S_X = \{x_1. \cdots, x_p\}$ and an outcome Y. We are interested in developing a **single** computationally efficient machine learning (ML) model $f(\cdot)$ to predict the outcome where the features arrive sequentially (and possibly unknowable order), and we want the model to perform well across the entire setting.

We will assume without loss of generality that the machine learning model operates on sets of features (i.e. permutation invariant), which can be easily accommodated for all models. Then all orders of the feature set is the symmetric Group S_p. Consider a permutation $s \in S_p$ so that the order of the sequence is $s(1), \cdots, s(p)$. We are interested in developing a machine learning model that performs "well" across $f(\{x_{s(1)}\}), f(\{x_{s(1)}, x_{s(2)}\}), \cdots, f(\{x_{s(1)}, x_{s(2)}, \cdots, x_{s(p)}\})$. Specifically, performing well at least includes three criteria:

- **Accuracy** The function f should have low MSE at the end of the sequence:
$$\min_{f \in F} \mathbb{E}_s[(y - f(\{x_{s(1)}, x_{s(2)}, \cdots, x_{s(p)}\}))^2]$$
- **Flexibility** The function should have low MSE as early as possible, i.e. for every $l \ll p$:
$$\min_{f \in F} \mathbb{E}_s[(y - f(\{x_{s(1)}, x_{s(2)}, \cdots, x_{s(l)}\}))^2]$$
- (Strong) **Consistency** for any given $(\boldsymbol{x}, y)$, and any sequence s we should have for all $l \in \{1, \cdots, p-1\}$:
$$|y - f(\{x_{s(1)}, x_{s(2)}, \cdots x_l\})| \geq |y - f(\{x_{s(1)}, x_{s(2)}, \cdots x_{l+1}\})|$$
Note that strong consistency requires for this constraint to be satisfied by every data point. In contrast, weak consistency holds in expectation.

More importantly, we are interested in machine learning models that perform well early on. There are two settings in which we are interested in:

- **Known Sequence**: We know s ahead of time, and therefore we are only interested in optimizing the performance of f with respect to s.
- **Unknown Sequence**: We do not know s ahead of time, and instead we assume that $s \sim F_{S_p}$ where F_{S_p} is some (potentially unknown) distribution over S_p.

The unknown sequence case is more interesting and realistic but there might be interesting insights for the known sequence case too.

Unknown Sequence

I can see two different types of approaches here.

Statistical Approach

For a function class $\mathcal{F}$ we aim to optimize the average case performance over all possible permutations:

$$\min_{f \in F} \mathbb{E}_{s \in F_{S_p}, l \in \{1, \cdots, p\}} [(y - f(\{x_{s(1)}, x_{s(2)}, \cdots, x_{s(l)}\}))^2]$$

The most "direct" and "naive" way to approach this is to utilize a neural network to serve as f and consider masking layers to randomly mask features to take care of the distribution over s, l. Using masking to deal with missing features is not a new idea in neural networks, but there are no implementations that does this effectively to solve the sequential problem. Furthermore the masking solution does not solve the consistency issue, and it is not immediately clear on how one can guarantee this in this formulation. A Lagrangian formulation of the constraint would add an exponential number of terms due to s being unknown.

Robust Optimization Approach

$$\min_{f \in F} \max_{s \in F_{S_p}} \sum_{l=1}^{p} w_l (y - f(\{x_{s(1)}, x_{s(2)}, \cdots, x_{s(l)}\}))^2$$

subject to the consistency constraints and w_l are weights for p. Given the discrete formulation, it seems to be difficult to come up with effective robust counterparts, but there are recent advances in this space that can potentially help us.