LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Introduction to Machine Learning

Evaluation: Introduction and Remarks

Department of Statistics – LMU Munich

INTRODUCTION

In predictive modeling, performance estimation can have different goals:

- **Performance estimation of a model:** Estimate *generalization* error of a model on new (unseen) data, drawn from the same data generating process that training data came from.
- **Performance estimation of an algorithm:** Estimate *generalization* error of a learning algorithm, trained on a data set of a certain size, on new (unseen) data, all drawn from the same data generating process.
- **Model selection:** Select the best model from a set of potential candidate models (e.g., different model classes, different hyperparameter settings, different features)
- **Learning curves:** How does the generalization error scale when an algorithm is trained on training sets of different sizes?

Obviously, all goals are quite related, i.e., reliable estimation of (predictive) performance is the foundation for all of them.

PERFORMANCE ESTIMATION

- Goal: Estimate performance on new data
- For now, we start by assuming to have a fixed model, already fitted on some data.
- We also assume to have some reasonable data for testing available.
- ML performance evaluation provides clear and simple protocols for reliable model validation. These protocols are often much simpler than classical statistical model diagnosis and rely only on few assumptions
- Most important assumption: Data we use is realistic and i.i.d.
- ML evaluation is still hard enough and offers LOTS of options to cheat yourself and especially your clients, mistakes can happen on many levels.

PERFORMANCE ESTIMATION

Clearly, we are looking for a statistical estimator - for the generalization error! Different levels of randomness are involved:

- Even if we are evaluating on a fixed test data set, this is only a sample and not full reality. The sample can be too small, then our estimator will be of high variance; the sample could not be from the distribution of interest, then our estimator will be biased.
- The same holds true for our model - it was only fitted on a sample. This creates another source of randomness, the training data sample. This is true, even if the fitting algorithm is deterministic.
- In ML, many learning algorithms are stochastic: Think random forest, or stochastic gradient descent. This is a third source of randomness.

METRICS: INNER VS. OUTER LOSS

- To judge the performance of a model on a given data set, we might want to produce a quantitative measure of the performance on that set.
- Usually we define a function that measures the quality of a prediction per observation.
- We then aggregate over the complete set - often by some form of averaging.

Don't we already know this? Sounds like loss functions and risk estimation? It nearly is the same. Nearly!

METRICS: INNER LOSS

- We already covered this, its associated empirical risk is optimized during model fitting
- The keyword above is **optimization**, some functions are much tamer to handle than others, smoothness and differentiability are often required for efficient optimization
- For this reason, we often choose something that's easier to handle numerically
- Another pretty practical reason might be: Our toolkit of choice only implements the optimization of a certain inner loss; changing the outer loss to something custom is simple, changing the inner often is not

METRICS: OUTER LOSS

- Performance metric to assess the model
- Should be carefully considered and selected
- There are no objectively better or worse metrics - YOU have to select depending on your application, domain and what you want
- Except for “should be reasonable and reflect what I want” there are no huge requirements for an outer metric, it is a function which takes a vector of prediction and a vector of labels and evaluates them
- Think about what will happen after a (wrong) prediction of your model, that should help to design an outer loss
- Yes, in model selection (later), we optimize it, but we use special techniques there anyway that can deal with arbitrary metrics

METRICS: INNER VS. OUTER LOSS

Usually, it is desired that inner and outer loss match, however, this is not always possible, as the outer loss is often numerically hard(er) to be handled during optimization and we might opt to approximate it.
Examples:

- In logistic regression we minimize the binomial loss
- In kNN there is no explicit loss minimization
- But when evaluating the models we might be more interested in (cost-weighted) classification error
- Or some of the more advanced metrics from ROC analysis like AUC

Nowadays, one can also optimize many of the harder losses directly, but this is less standard, less often implemented and we will not cover this here.