SLT - Short Summary

> *By Joost Visser*

`LaTeX commands`

Chapter 1 - Some definitions

Chapter 2 - Binary classification and Regression

Chapter 3 - Competing Goals: approximation vs estimation

Chapter 4 - Estimation of Lipschitz smooth functions

Chapter 5 - Introduction to PAC learning

Chapter 6 - Concentration Bounds

Chapter 7 - General bounds for bounded losses

Chapter 8 - Countably Infinite Model Spaces

Complexity Regularization Bounds

Theorem 6 - Complexity Regularized Model Selection:

Let $\mathcal{F}$ be a *countable* collection of models and assign a real number $c(f)$ to each $f \in \mathcal{F}$ such that:

$$\sum_{f \in \mathcal{F}} e^{-c(f)} \leq 1$$

Let us define the minimum complexity regularized model (i.e. the ERM with regularization):

$$\hat{f}_n = \arg\min_{f \in \mathcal{F}} \left\{ \hat{R}_n(f) + \sqrt{\frac{c(f) + \frac{1}{2}\log n}{2n}} \right\}$$

Then we can bound our expected risk compared to the actual risk:

$$\mathbb{E}[R(\hat{f}_n)] \leq \inf_{f \in \mathcal{F}} \left\{ R(f) + \sqrt{\frac{c(f) + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \right\}$$

This theorem is quite useful, as it lets us bound the expected risk and thererfore gives us an idea of what a good estimator is for $\hat{f}_n$.

Histogram example

The expected value of our empirical risk when choosing the number of bins automatically performs almost also as good as the best k (number of bins) possible! Since:

$$\mathbb{E}[R(\hat{f}_n)] \leq \inf_{k \in \mathbb{N}} \left\{ \min_{f \in \mathcal{F}_k} R(f) + \sqrt{\frac{(k + k^d)\log 2 + \frac{1}{2}\log n}{2n}} + \frac{1}{\sqrt{n}} \right\}$$

- $\hat{f}_n = \hat{f}_n^{(\hat{k}_n)}$ - Our best k we could automatically determine depending on the data using the empricial risk minimizer where the $\hat{k}_n$ is found by minimizing:
  - $\hat{k}_n = \arg\min_{k \in \mathbb{N}}$ ERM of each subclass + penalized term.

Chapter 9 - The Histogram Classifier revisited

Chapter 10 - Decision Trees and Classification

Chapter 11 - VC Bounds

Chapter 12 - Denoising of Piecewise Smooth Functions