3 Risk and Loss Functions

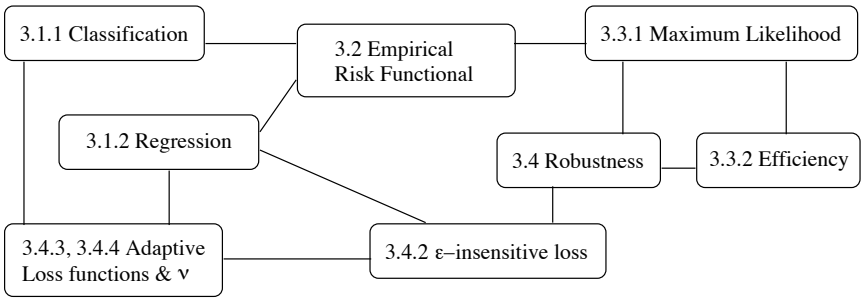
One of the most immediate requirements in any learning problem is to specify what exactly we would like to achieve, minimize, bound, or approximate. In other words, we need to determine a criterion according to which we will assess the quality of an estimate $f : \mathcal{X} \to \mathcal{Y}$ obtained from data.

This question is far from trivial. Even in binary classification there exist ample choices. The selection criterion may be the fraction of patterns classified correctly, it could involve the confidence with which the classification is carried out, or it might take into account the fact that losses are not symmetric for the two classes, such as in health diagnosis problems. Furthermore, the loss for an error may be input-dependent (for instance, meteorological predictions may require a higher accuracy in urban regions), and finally, we might want to obtain probabilities rather than a binary prediction of the class labels -1 and 1. Multi class discrimination and regression add even further levels of complexity to the problem. Thus we need a means of encoding these criteria.

Overview

The chapter is structured as follows: in Section 3.1, we begin with a brief overview of common loss functions used in classification and regression algorithms. This is done without much mathematical rigor or statistical justification, in order to provide basic working knowledge for readers who want to get a quick idea of the default design choices in the area of kernel machines. Following this, Section 3.2 formalizes the idea of risk. The risk approach is the predominant technique used in this book, and most of the algorithms presented subsequently minimize some form of a risk functional. Section 3.3 treats the concept of loss functions from a statistical perspective, points out the connection to the estimation of densities and introduces the notion of efficiency. Readers interested in more detail should also consider Chapter 16, which discusses the problem of estimation from a Bayesian perspective. The later parts of this section are intended for readers interested in the more theoretical details of estimation. The concept of robustness is introduced in Section 3.4. Several commonly used loss functions, such as Huber's loss and the ε-insensitive loss, enjoy robustness properties with respect to rather general classes of distributions. Beyond the basic relations, will show how to adjust the ε-insensitive loss in such a way as to accommodate different amounts of variance automatically. This will later lead to the construction of so-called ν Support Vector Algorithms (see Chapters 7, 8, and 9).

While technical details and proofs can be omitted for most of the present chapter, we encourage the reader to review the practical implications of this section.



Prerequisites

As usual, exercises for all sections can be found at the end. The chapter requires knowledge of probability theory, as introduced in Section B.1.

3.1 Loss Functions

Let us begin with a formal definition of what we mean by the loss incurred by a function f at location x, given an observation y.

Definition 3.1 (Loss Function) *Denote by $(x, y, f(x)) \in \mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$ the triplet consisting of a pattern x, an observation y and a prediction $f(x)$. Then the map $c : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$ with the property $c(x, y, y) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$ will be called a loss function.*

Note that we require c to be a nonnegative function. This means that we will never get a payoff from an extra good prediction. If the latter was the case, we could always recover non-negativity (provided the loss is bounded from below), by using a simple shift operation (possibly depending on x). Likewise we can always satisfy the condition that exact predictions ($f(x) = y$) never cause any loss. The advantage of these extra conditions on c is that we know that the minimum of the loss is 0 and that it is obtainable, at least for a given x, y.

Minimized Loss $\neq$ Incurred Loss

Next we will formalize different kinds of *loss*, as described informally in the introduction of the chapter. Note that the incurred loss is not always the quantity that we will attempt to minimize. For instance, for algorithmic reasons, some loss functions will prove to be infeasible (the binary loss, for instance, can lead to NP-hard optimization problems [367]). Furthermore, statistical considerations such as the desire to obtain confidence levels on the prediction (Section 3.3.1) will also influence our choice.

3.1.1 Binary Classification

Misclassification Error

The simplest case to consider involves counting the misclassification error if pattern x is classified wrongly we incur loss 1, otherwise there is no penalty.:

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ 1 & \text{otherwise} \end{cases} \tag{3.1}$$