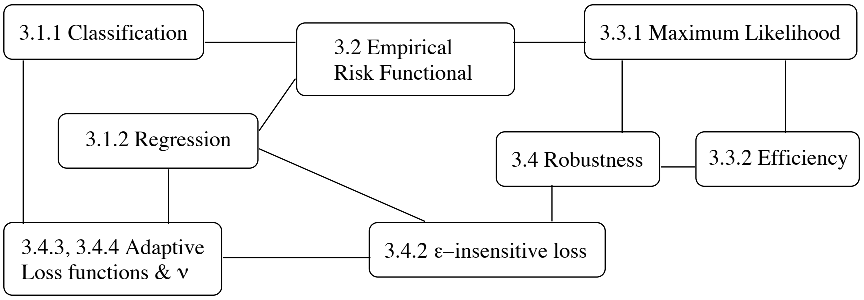

Prerequisites

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.:

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

This definition of c does not distinguish between different classes and types of errors (false positive or negative).[1]

Asymmetric and Input-Dependent Loss

A slight extension takes the latter into account. For the sake of simplicity let us assume, as in (3.1), that we have a binary classification problem. This time, however, the loss may depend on a function $\tilde{c}(x)$ which accounts for input-dependence, i.e.

$$c(x, y, f(x)) = \begin{cases} 0 & \text{if } y = f(x) \\ \tilde{c}(x) & \text{otherwise} \end{cases} \tag{3.2}$$

A simple (albeit slightly contrived) example is the classification of objects into rocks and diamonds. Clearly, the incurred loss will depend largely on the weight of the object under consideration.

Analogously, we might distinguish between errors for $y = 1$ and $y = -1$ (see, e.g., [331] for details). For instance, in a fraud detection application, we would like to be really sure about the situation before taking any measures, rather than losing potential customers. On the other hand, a blood bank should consider even the slightest suspicion of disease before accepting a donor.

Confidence Level

Rather than predicting only whether a given object x belongs to a certain class y, we may also want to take a certain confidence level into account. In this case, $f(x)$ becomes a real-valued function, even though $y \in \{-1, 1\}$.

Soft Margin Loss

In this case, $\operatorname{sgn}(f(x))$ denotes the class label, and the absolute value $|f(x)|$ the confidence of the prediction. Corresponding loss functions will depend on the product $yf(x)$ to assess the quality of the estimate. The *soft margin* loss function, as introduced by Bennett and Mangasarian [40, 111], is defined as

$$c(x, y, f(x)) = \max(0, 1 - yf(x)) = \begin{cases} 0 & \text{if } yf(x) \geq 1, \\ 1 - yf(x) & \text{otherwise.} \end{cases} . \tag{3.3}$$

In some cases [348, 125] (see also Section 10.6.2) the squared version of (3.3) provides an expression that can be minimized more easily;

$$c(x, y, f(x)) = \max(0, 1 - yf(x))^2. \tag{3.4}$$

Logistic Loss

The soft margin loss closely resembles the so-called *logistic* loss function (cf. [251], as well as Problem 3.1 and Section 16.1.1);

$$c(x, y, f(x)) = \ln\left(1 + \exp\left(-yf(x)\right)\right). \tag{3.5}$$

We will derive this loss function in Section 3.3.1. It is used in order to associate a probabilistic meaning with $f(x)$.

Note that in both (3.3) and (3.5) (nearly) no penalty occurs if $yf(x)$ is sufficiently large, i.e. if the patterns are classified correctly with large confidence. In particular, in (3.3) a minimum confidence of 1 is required for zero loss. These loss functions

[1] A *false positive* is a point which the classifier erroneously assigns to class 1, a *false negative* is erroneously assigned to class -1.

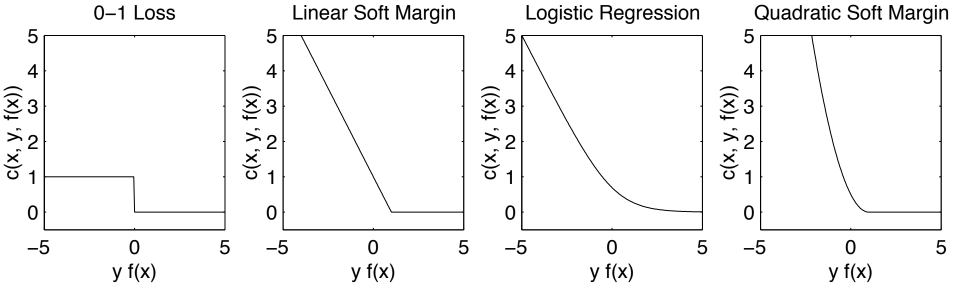

Figure 3.1 From left to right: 0-1 loss, linear soft margin loss, logistic regression, and quadratic soft margin loss. Note that both soft margin loss functions are upper bounds on the 0-1 loss.

led to the development of *large margin classifiers* (see [491, 460, 504] and Chapter 5 for further details). Figure 3.1 depicts various popular loss functions.[2]

Multi Class Discrimination

Matters are more complex when dealing with more than two classes. Each type of misclassification could potentially incur a different loss, leading to an $M \times M$ matrix (M being the number of classes) with positive off-diagonal and zero diagonal entries. It is still a matter of ongoing research in which way a confidence level should be included in such cases (cf. [41, 311, 593, 161, 119]).

3.1.2 Regression

When estimating real-valued quantities, it is usually the size of the difference $y - f(x)$, i.e. the amount of misprediction, rather than the product $yf(x)$, which is used to determine the quality of the estimate. For instance, this can be the actual loss incurred by mispredictions (e.g., the loss incurred by mispredicting the value of a financial instrument at the stock exchange), provided the latter is known and computationally tractable.[3] Assuming location independence, in most cases the loss function will be of the type

$$c(x, y, f(x)) = \tilde{c}(f(x) - y). \tag{3.7}$$

See Figure 3.2 below for several regression loss functions. Below we list the ones most common in kernel methods.

[2] Other popular loss functions from the generalized linear model context include the inverse complementary log-log function. It is given by

$$c(x, y, f(x)) = 1 - \exp(-\exp(yf(x))). \tag{3.6}$$

This function, unfortunately, is not convex and therefore it will not lead to a convex optimization problem. However, it has nice robustness properties and therefore we think that it should be investigated in the present context.

[3] As with classification, computational tractability is one of the primary concerns. This is not always satisfying from a statistician's point of view, yet it is crucial for any practical implementation of an estimation algorithm.

Squared Loss

The popular choice is to minimize the sum of squares of the residuals $f(x) - y$. As we shall see in Section 3.3.1, this corresponds to the assumption that we have additive normal noise corrupting the observations y_i. Consequently we minimize

$$c(x, y, f(x)) = (f(x) - y)^2 \text{ or equivalently } \tilde{c}(\xi) = \xi^2. \tag{3.8}$$

For convenience of subsequent notation, $\frac{1}{2}\xi^2$ rather than ξ^2 is often used.

ε-insensitive Loss and ℓ_1 Loss

An extension of the soft margin loss (3.3) to regression is the *ε-insensitive* loss function [561, 572, 562]. It is obtained by symmetrization of the "hinge" of (3.3),

$$\tilde{c}(\xi) = \max(|\xi| - \varepsilon, 0) =: |\xi|_\varepsilon. \tag{3.9}$$

The idea behind (3.9) is that deviations up to ε should not be penalized, and all further deviations should incur only a linear penalty. Setting $\varepsilon = 0$ leads to an ℓ_1 loss, i.e., to minimization of the sum of absolute deviations. This is written

$$\tilde{c}(\xi) = |\xi|. \tag{3.10}$$

We will study these functions in more detail in Section 3.4.2.

Practical Considerations

For efficient implementations of learning procedures, it is crucial that loss functions satisfy certain properties. In particular, they should be cheap to compute, have a small number of discontinuities (if any) in the first derivative, and be convex in order to ensure the uniqueness of the solution (see Chapter 6 and also Problem 3.6 for details). Moreover, we may want to obtain solutions that are computationally efficient, which may disregard a certain number of training points. This leads to conditions such as vanishing derivatives for a range of function values $f(x)$. Finally, requirements such as outlier resistance are also important for the construction of estimators.

3.2 Test Error and Expected Risk

Now that we have determined how errors should be penalized on specific instances $(x, y, f(x))$, we have to find a method to combine these (local) penalties. This will help us to assess a particular estimate f.

In the following, we will assume that there exists a probability distribution $\mathrm{P}(x, y)$ on $\mathcal{X} \times \mathcal{Y}$ which governs the data generation and underlying functional dependency. Moreover, we denote by $\mathrm{P}(y|x)$ the *conditional* distribution of y given x, and by $d\mathrm{P}(x, y)$ and $d\mathrm{P}(y|x)$ the integrals with respect to the distributions $\mathrm{P}(x, y)$ and $\mathrm{P}(y|x)$ respectively (cf. Section B.1.3).

3.2.1 Exact Quantities

Unless stated otherwise, we assume that the data (x, y) are drawn iid (independent and identically distributed, see Section B.1) from $\mathrm{P}(x, y)$. Whether or not we have