I CONCEPTS AND TOOLS

The generic can be more intense than the concrete.

J. L. Borges[1]

We now embark on a more systematic presentation of the concepts and tools underlying Support Vector Machines and other kernel methods.

In machine learning problems, we try to discover structure in data. For instance, in pattern recognition and regression estimation, we are given a training set $(x_1, y_1), \ldots, (x_m, y_m) \in \mathcal{X} \times \mathcal{Y}$, and attempt to predict the outputs y for previously unseen inputs x. This is only possible if we have some measure that tells us how (x, y) is related to the training set. Informally, we want similar inputs to lead to similar outputs.[2] To formalize this, we have to state what we mean by *similar*.

A particularly simple yet surprisingly useful notion of similarity of *inputs* — the one we will use throughout this book — derives from embedding the data into a Euclidean feature space and utilizing geometrical concepts. Chapter 2 describes how certain classes of kernels induce feature spaces, and how one can compute dot products, and thus angles and distances, without having to explicitly work in these potentially infinite-dimensional spaces. This leads to a rather general class of similarity measure to be used on the inputs.

1. From *A History of Eternity*, in *The Total Library*, Penguin, London, 2001.
2. This procedure can be traced back to an old maxim of law: *de similibus ad similia eadem ratione procedendum est* — from things similar to things similar we are to proceed by the same rule.

On the *outputs*, similarity is usually measured in terms of a *loss function* stating how "bad" it is if the predicted y does not match the true one. The training of a learning machine commonly involves a *risk functional* that contains a term measuring the loss incurred for the training patterns. The concepts of loss and risk are introduced in depth in Chapter 3.

This is not the full story, however. In order to generalize well to the test data, it is not sufficient to "explain" the training data. It is also necessary to control the complexity of the model used for explaining the training data, a task that is often accomplished with the help of *regularization* terms, as explained in Chapter 4. Specifically, one utilizes objective functions that involve both the empirical loss term and a regularization term. From a *statistical* point of view, we can expect the function minimizing a properly chosen objective function to work well on test data, as explained by statistical learning theory (Chapter 5). From a *practical* point of view, however, it is not at all straightforward to *find* this minimizer. Indeed, the quality of a loss function or a regularizer should be assessed not only on a statistical basis but also in terms of the feasibility of the objective function minimization problem. In order to be able to assess this, and in order to obtain a thorough understanding of practical algorithms for this task, we conclude this part of the book with an in-depth review of optimization theory (Chapter 6).

The chapters in this part of the book assume familiarity with basic concepts of linear algebra and probability theory. Readers who would like to refresh their knowledge of these topics may want to consult Appendix B beforehand.