# Chapter 1
# Introduction

***The beginning is the most important part of the work.***
*Plato, The Republic*

Suppose one is confronted with a medical classification problem. What trustworthy technique should one then use to solve it? Support vector machines (SVMs) are known to be a smart choice. But how can one make a personal, more flexible implementation of the learning engine that makes them run that well? And how does one open the black box behind their predicted diagnosis and explain the reasoning to the otherwise reluctant fellow physicians? Alternatively, one could choose to develop a more versatile evolutionary algorithm (EA) to tackle the classification task towards a potentially more understandable logic of discrimination. But will comprehensibility weigh more than accuracy?

It is therefore the goal of this book to investigate how can both efficiency as well as transparency in prediction be achieved when dealing with classification by means of SVMs and EAs. We will in turn address the following choices:

1. Proficient, black box SVMs (found in chapter 2).
2. Transparent but less efficient EAs (chapters 3, 4 and 5).
3. Efficient learning by SVMs, flexible training by EAs (chapter 6).
4. Predicting by SVMs, explaining by EAs (chapter 7).

The book starts by reviewing the classical as well as the state of the art approaches to SVMs and EAs for classification, as well as methods for their hybridization. Nevertheless, it is especially focused on the authors' personal contributions to the enunciated scope.

Each presented new methodology is accompanied by a short experimental section on several benchmark data sets to get a grasp of its results. For more in-depth experimentally-related information, evaluation and test cases the reader should consult the corresponding referenced articles.

Throughout this book, we will assume that a classification problem is defined by the subsequent components:

- a set of $m$ training pairs, where each holds the information related to a data sample (a sequence of values for given attributes or indicators) and its confirmed target (outcome, decision attribute).

C. Stoean and R. Stoean, *Support Vector Machines and Evolutionary Algorithms for Classification*, Intelligent Systems Reference Library 69,
DOI: 10.1007/978-3-319-06941-8_1, © Springer International Publishing Switzerland 2014

- every sample (or example, record, point, instance) is described by $n$ attributes: $x_i \in [a_1,b_1] \times [a_2,b_2] \times ... \times [a_n,b_n]$, where $a_i, b_i$ denote the bounds of definition for every attribute.
- each corresponding outcome $y_i \in \{0,1,...,k-1\}$, where there are $k$ possible classes.
- a set of $l$ validation couples $(x_i^v, y_i^v)$, in order to assess the prediction error of the model. Please note that this set can be constituted only in the situation when the amount of data is sufficiently large [Hastie et al, 2001].
- a set of $p$ test pairs of the type $(x_i^{'}, y_i^{'})$, to measure the generalization error of the approach [Hastie et al, 2001].
- for both the validation and test sets, the target is unknown to the learning machine and must be predicted.

As illustrated in Fig. 1.1, learning pursues the following steps:

- A chosen classifier learns the associations between each training sample and the acknowledged output (training phase).
- Either in a black box manner or explicitly, the obtained inference engine takes each test sample and makes a forecast on its probable class, according to what has been learnt (testing phase).
- The percent of correctly labeled new cases out of the total number of test samples is next computed (accuracy of prediction).
- Cross-validation (as in statistics) must be employed in order to estimate the prediction accuracy that the model will exhibit in practice. This is done by selecting training/test sets for a number of times according to several possible schemes.
- The generalization ability of the technique is eventually assessed by computing the test prediction accuracy as averaged over the several rounds of cross-validation.
- Once more, if we dispose of a substantial data collection, it is advisable to additionally make a prediction on the targets of validation examples, prior to the testing phase. This allows for an estimation of the prediction error of the constructed model, computed also after several rounds of cross-validation that now additionally include the validation set [Hastie et al, 2001].

Note that, in all conducted experiments throughout this book, we were not able to use the supplementary validation set, since the data samples in the chosen sets were insufficient. This was so because, for the benchmark data sets, we selected those that were both easier to understand for the reader and cleaner to make reproducing of results undemanding. For the real-world available tasks, the data was not too numerous as it comes from hospitals in Romania, where such sets have been only recently collected and prepared for computer-aided diagnosis purposes.

What is more, we employ the repeated random sub-sampling method for cross-validation, where the multiple training/test sets are chosen by randomly splitting the data in two for the given number of times.

As the task for classification is to achieve an optimal separation of given data into classes, SVMs regard learning from a geometrical point of view. They assume the existence of a separating surface between every two classes labeled as -1 and

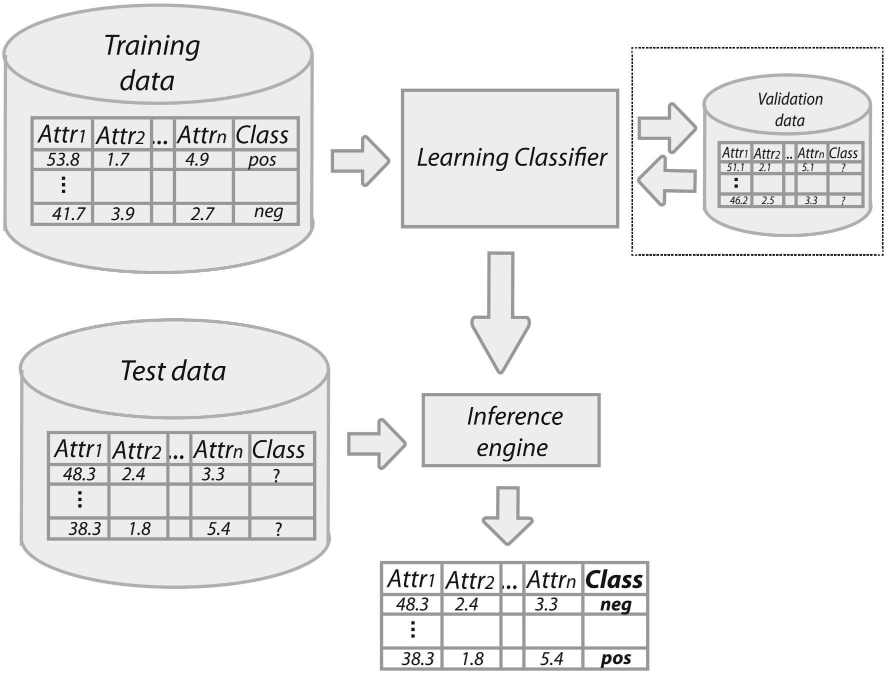

**Fig. 1.1** The classifier learns the associations between the training samples and their corresponding classes and is then calibrated on the validation samples. The resulting inference engine is subsequently used to classify new test data. The validation process can be omitted, especially for relatively small data sets. The process is subject to cross-validation, in order to estimate the practical prediction accuracy.

1. The aim then becomes the discovery of the appropriate decision hyperplane. The book will outline all the aspects related to classification by SVMs, including the theoretical background and detailed demonstrations of their behavior (chapter 2).

EAs, on the other hand, are able to evolve rules that place each sample into a corresponding class, while training on the available data. The rules can take different forms, from the IF-THEN conjunctive layout from computational logic to complex structures like trees. In this book, we will evolve thresholds for the attributes of the given data examples. These IF-THEN constructions can also be called rules, but we will more rigorously refer to them as class prototypes, since the former are generally supposed to have a more elaborate formulation. Two techniques that evolve class prototypes while maintaining diversity during evolution are proposed: a multimodal EA that separates potential rules of different classes through a common radius means (chapter 4) and another that creates separate collaborative populations connected to each outcome (chapter 5).

Combinations between SVMs and EAs have been widely explored by the machine learning community and on different levels. Within this framework, we outline approaches tackling two degrees of hybridization: EA optimization at the core of SVM learning (chapter 6) and a stepwise learner that separates by SVMs and explains by EAs (chapter 7).

Having presented these options – SVMs alone, single EAs and hybridization at two stages of learning to classify – the question that we address and try to answer through this book is: what choice is more advantageous, if one takes into consideration one or more of the following characteristics:

- prediction accuracy
- comprehensibility
- simplicity
- flexibility
- runtime