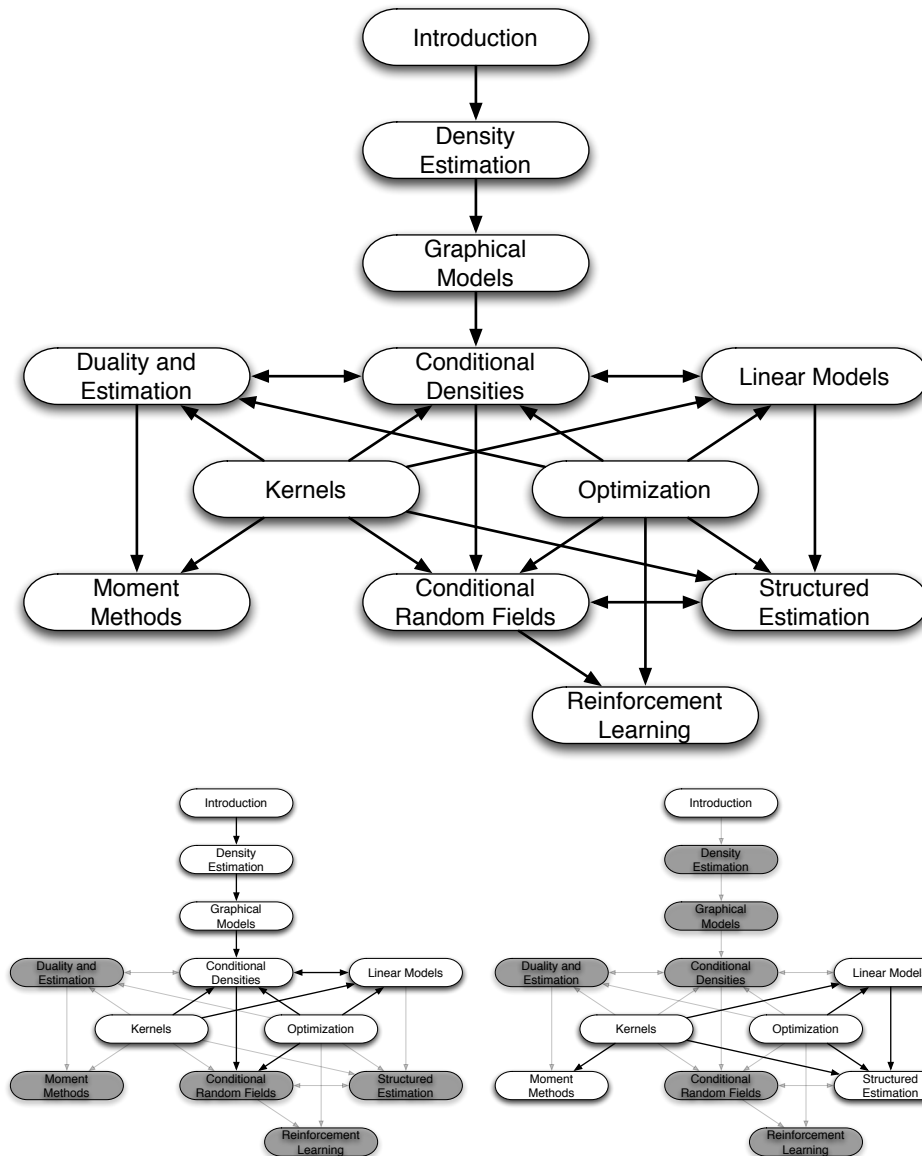


Structure of the Book



Canberra, August 2008

Introduction

Over the past two decades Machine Learning has become one of the mainstays of information technology and with that, a rather central, albeit usually hidden, part of our life. With the ever increasing amounts of data becoming available there is good reason to believe that smart data analysis will become even more pervasive as a necessary ingredient for technological progress.

The purpose of this chapter is to provide the reader with an overview over the vast range of applications which have at their heart a machine learning problem and to bring some degree of order to the zoo of problems. After that, we will discuss some basic tools from statistics and probability theory, since they form the language in which many machine learning problems must be phrased to become amenable to solving. Finally, we will outline a set of fairly basic yet effective algorithms to solve an important problem, namely that of classification. More sophisticated tools, a discussion of more general problems and a detailed analysis will follow in later parts of the book.

1.1 A Taste of Machine Learning

Machine learning can appear in many guises. We now discuss a number of applications, the types of data they deal with, and finally, we formalize the problems in a somewhat more stylized fashion. The latter is key if we want to avoid reinventing the wheel for every new application. Instead, much of the *art* of machine learning is to reduce a range of fairly disparate problems to a set of fairly narrow prototypes. Much of the *science* of machine learning is then to solve those problems and provide good guarantees for the solutions.

1.1.1 Applications

Most readers will be familiar with the concept of web page **ranking**. That is, the process of submitting a query to a search engine, which then finds webpages relevant to the query and which returns them in their order of relevance. See e.g. Figure 1.1 for an example of the query results for “machine learning”. That is, the search engine returns a sorted list of webpages given a query. To achieve this goal, a search engine needs to ‘know’ which

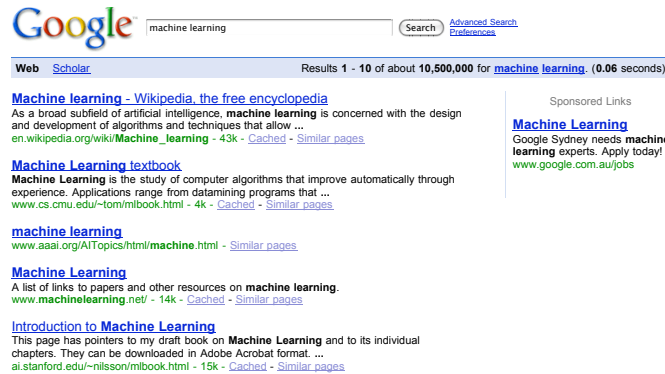


Fig. 1.1. The 5 top scoring webpages for the query “machine learning”

pages are relevant and which pages match the query. Such knowledge can be gained from several sources: the link structure of webpages, their content, the frequency with which users will follow the suggested links in a query, or from examples of queries in combination with manually ranked webpages. Increasingly machine learning rather than guesswork and clever engineering is used to *automate* the process of designing a good search engine [RPB06].

A rather related application is **collaborative filtering**. Internet bookstores such as Amazon, or video rental sites such as Netflix use this information extensively to entice users to purchase additional goods (or rent more movies). The problem is quite similar to the one of web page ranking. As before, we want to obtain a sorted list (in this case of articles). The key difference is that an explicit query is missing and instead we can only use past purchase and viewing decisions of the user to predict future viewing and purchase habits. The key side information here are the decisions made by *similar* users, hence the collaborative nature of the process. See Figure 1.2 for an example. It is clearly desirable to have an automatic system to solve this problem, thereby avoiding guesswork and time [BK07].

An equally ill-defined problem is that of **automatic translation** of documents. At one extreme, we could aim at fully *understanding* a text before translating it using a curated set of rules crafted by a computational linguist well versed in the two languages we would like to translate. This is a rather arduous task, in particular given that text is not always grammatically correct, nor is the document understanding part itself a trivial one. Instead, we could simply use *examples* of translated documents, such as the proceedings of the Canadian parliament or other multilingual entities (United Nations, European Union, Switzerland) to *learn* how to translate between the two

languages. In other words, we could use examples of translations to learn how to translate. This machine learning approach proved quite successful [?].

Many security applications, e.g. for access control, use face recognition as one of its components. That is, given the photo (or video recording) of a person, recognize who this person is. In other words, the system needs to **classify** the faces into one of many categories (Alice, Bob, Charlie, ...) or decide that it is an unknown face. A similar, yet conceptually quite different problem is that of verification. Here the goal is to verify whether the person in question is who he claims to be. Note that differently to before, this is now a yes/no question. To deal with different lighting conditions, facial expressions, whether a person is wearing glasses, hairstyle, etc., it is desirable to have a system which *learns* which features are relevant for identifying a person.

Another application where learning helps is the problem of **named entity recognition** (see Figure 1.4). That is, the problem of identifying entities, such as places, titles, names, actions, etc. from documents. Such steps are crucial in the automatic digestion and understanding of documents. Some modern e-mail clients, such as Apple's Mail.app nowadays ship with the ability to identify addresses in mails and filing them automatically in an address book. While systems using hand-crafted rules can lead to satisfactory results, it is far more efficient to use examples of marked-up documents to learn such dependencies automatically, in particular if we want to deploy our system in many languages. For instance, while 'bush' and 'rice'



Fig. 1.2. Books recommended by Amazon.com when viewing Tom Mitchell's Machine Learning Book [Mit97]. It is desirable for the vendor to recommend relevant books which a user might purchase.



Fig. 1.3. 11 Pictures of the same person taken from the Yale face recognition database. The challenge is to recognize that we are dealing with the same person in all 11 cases.

HAVANA (Reuters) - The European Union's top development aid official left Cuba on Sunday convinced that EU diplomatic sanctions against the communist island should be dropped after Fidel Castro's retirement, his main aide said.

```
<TYPE="ORGANIZATION">HAVANA</> (<TYPE="ORGANIZATION">Reuters</>) - The
<TYPE="ORGANIZATION">European Union</>'s top development aid official left
<TYPE="ORGANIZATION">Cuba</> on Sunday convinced that EU diplomatic sanctions
against the communist <TYPE="LOCATION">island</> should be dropped after
<TYPE="PERSON">Fidel Castro</>'s retirement, his main aide said.
```

Fig. 1.4. Named entity tagging of a news article (using LingPipe). The relevant locations, organizations and persons are tagged for further information extraction.

are clearly terms from agriculture, it is equally clear that in the context of contemporary politics they refer to members of the Republican Party.

Other applications which take advantage of learning are **speech recognition** (annotate an audio sequence with text, such as the system shipping with Microsoft Vista), the recognition of handwriting (annotate a sequence of strokes with text, a feature common to many PDAs), trackpads of computers (e.g. Synaptics, a major manufacturer of such pads derives its name from the synapses of a neural network), the detection of failure in jet engines, avatar behavior in computer games (e.g. Black and White), direct marketing (companies use past purchase behavior to guesstimate whether you might be willing to purchase even more) and floor cleaning robots (such as iRobot's Roomba). The overarching theme of learning problems is that there exists a nontrivial dependence between some observations, which we will commonly refer to as x and a desired response, which we refer to as y , for which a simple set of deterministic rules is not known. By using learning we can infer such a dependency between x and y in a systematic fashion.

We conclude this section by discussing the problem of **classification**, since it will serve as a prototypical problem for a significant part of this book. It occurs frequently in practice: for instance, when performing spam filtering, we are interested in a yes/no answer as to whether an e-mail contains relevant information or not. Note that this issue is quite user dependent: for a frequent traveller e-mails from an airline informing him about recent discounts might prove valuable information, whereas for many other recipients this might prove more of an nuisance (e.g. when the e-mail relates to products available only overseas). Moreover, the nature of annoying e-mails might change over time, e.g. through the availability of new products (Viagra, Cialis, Levitra, . . .), different opportunities for fraud (the Nigerian 419 scam which took a new twist after the Iraq war), or different data types (e.g. spam which consists mainly of images). To combat these problems we