University of Bristol

DEPARTMENT OF COMPUTER SCIENCE

On the state of the art in Machine Learning: a personal review

Peter A. Flach

December 2000 | CSTR-00-018

On the state of the art in Machine Learning: a personal review

Peter A. Flach

December 15, 2000

Abstract

This paper reviews a number of recent books related to current developments in machine learning. Some (anticipated) trends will be sketched. These include: a trend towards combining approaches that were hitherto regarded as distinct and were studied by separate research communities; a trend towards a more prominent role of representation; and a tighter integration of machine learning techniques with techniques from areas of application such as bioinformatics. The intended readership has some knowledge of what machine learning is about, but brief tutorial introductions to some of the more specialist research areas will also be given.

1 Introduction

This paper reviews a number of books that appeared recently in my main area of expertise, machine learning. Following a request from the book review editors of *Artificial Intelligence*, I selected about a dozen books with which I either was already familiar, or which I would be happy to read in order to find out what's new in that particular area of the field. The choice of books is therefore subjective rather than comprehensive (and also biased towards books I liked). However, it does reflect a personal view of what I think are exciting developments in machine learning, and as a secondary aim of this paper I will expand a bit on this personal view, sketching some of the current and anticipated future trends. This is something of an experiment (both for the journal and for me) and it is hoped that readers find this kind of survey-review useful. The intended readership has some knowledge of what machine learning is about, but I will also give brief tutorial introductions to some of the more specialist research areas covered below.

One trend that I have identified after reading up on the state of the art is towards combining approaches that were hitherto regarded as distinct and were studied by separate research communities. I will briefly mention a few examples that will be discussed at greater length below. Support Vector Machines have roots in the neural information processing community, because of their perceptron-like core, but also in the computational learning theory field, in particular through the work of Vladimir Vapnik, and they hold great promise for the field of symbolic learning, by virtue of their kernel-based feature space transformation. Graphical probabilistic models provide a very attractive marriage of logic and probability theory, an old problem in artificial intelligence that

has for a long time been abandoned or defined away. They can also very naturally be seen as a way of combining reasoning and learning. Molecular biologists recognise the need for sophisticated machine learning methods for solving their sequence identification and prediction problems, and contribute to the development of such methods. After a period in which artificial intelligence seemed to become ever more specialised, these are very encouraging developments hinting at a renewed vitality of the field.

Another trend is the use of richer representations, or at least to explicitly include a logical component in the representation that can be adapted to the domain. Traditionally, machine learning methods have used the so-called attribute-value format, where an instance is described by a vector of attribute values (attributes are defined implicitly by their position), and learned hypotheses combine constraints on attribute values in various ways (e.g., decision trees or if-then rules). Inductive logic programming (ILP) research aimed to develop machine learning techniques employing logic programming representations, but for a long time the connection was rather indirect and ILP techniques were often more familiar to logic programmers than to machine learners. Nowadays, the connection between representation and learning is explored in two ways: either one uses a feature generation method (where features could for instance be the numbers of occurrences of all possible subsequences in a given sequence) and a learning method which is able to deal with very large numbers of features, such as Support Vector Machines; or one exploits the fact that attribute-value vectors are in fact instances of a Cartesian product datatype, which suggests ways to extend the learner to richer datatypes such as lists and sets. Either way, the key point is that the representation formalism should not be compiled into the learning algorithm, but rather be treated as an independent ingredient.

A third trend is the 'absorption' of machine learning techniques into application disciplines. Machine learning has always been concerned with applications, but typically these were carried out by machine learning researchers wanting to demonstrate the viability of their approach on a real-life problem. Nowadays, researchers in areas such as bioinformatics are increasingly aware of the usefulness of machine learning techniques alongside the more traditional statistical techniques that they were using thus far. This is related to the previous point about representations, since the weak point of statistical methods is that they disregard representation almost completely. In parametric statistics, a 'model' is, e.g., a polynomial of a certain degree, and 'learning' or model fitting is estimating the parameters of the model. Symbolic machine learning employs a more sophisticated notion of a model, and learning involves both model search and parameter estimation. This used to be frowned upon by statisticians (the term 'data mining' originally had a derogatory meaning: 'torturing the data until they confess') because it would lead to overfitting, but although this is a real danger for any learning approach, nowadays machine learning researchers use sophisticated statistical techniques to validate their results. Ideally, machine learning becomes 'just' another tool in the data analyst's toolbox, which would make it one of the most successful subdisciplines of artificial intelligence. While we are not quite there yet, I do believe that significant progress has been made towards this goal. A key point in the success of machine learning applications will be the extent to which they can be fine-tuned to the domain of investigation. The strength of off-the-shelf, domain-independent methods is a weakness at the same time, and the more sophisticated the application domain is,

the more the weakness will be felt. We do not need, e.g., a general-purpose distance metric for lists; we need one which works for DNA-sequences, and another one which works for text documents. The connection between machine learning and application areas will be mutual: not only do the application people need to know about machine learning, machine learners need to know about the application area – not just at the moment of application, but already at the research level.

1.1 Plan of the paper

The outline of the paper is as follows. In Section 2 I review a number of introductory books on machine learning and data mining, varying from academic textbooks to practical how-to guides and books written from a business perspective. Sections 3 and 4 review research areas that have recently become popular and hold promise for the future, namely Support Vector Machines and graphical probabilistic models. Section 5 looks at some recent books that are written from the joint perspective of machine learning and an area of application, in this case bioinformatics and ecology. Finally, Section 6 concludes.

Before continuing, a few words of caution are perhaps in order. I have already indicated that the sample of books reviewed below is unrepresentative, because they have been selected according to my interests and competences. In addition, the sample is biased because of the class distribution, which is very skewed: I didn't include books I didn't like. This is not because I like all machine learning books, but because this paper is also intended as an overview of part of the state of the art in machine learning (and also to limit the paper's length). However, even though I generally liked all the books featuring in this paper, I have tried to be honest and indicate any shortcomings I found. As always when subjective judgements are involved, not everybody will agree. While it is often customary to give the authors or editors the opportunity to add their response to a published review of their book, I have not pursued this for the present paper because of its scope. That being said, the book review editors have informed me that they encourage authors or editors of reviewed books to submit their response, should they have the wish to do so, which could then be bundled and published in a future issue of *Artificial Intelligence*. Finally, I have taken the liberty of varying the amount of words spent on each book. This is again based on subjective judgement but I have also tried to take the reader's interests into account, by highlighting material that is likely to be of interest to a wider audience.

2 Introductory texts

I will start with a review of some of the introductory (text)books on machine learning and data mining that are available nowadays. Three relatively recent textbooks on machine learning are Pat Langley's *Elements of Machine Learning* (Morgan Kaufmann, 1996), Tom Mitchell's *Machine Learning* (McGraw-Hill, 1997), and *Data mining: practical machine learning tools and techniques with Java implementations* by Ian Witten and Eibe Frank (Morgan Kaufmann, 2000). Among these three, Mitchell's book is a textbook in a more conventional sense, and also seems the more established

one.[1] Langley, on the other hand, provides a much more personal perspective on the field of machine learning; while I wouldn't recommend it as the main text in a machine learning course, it provides excellent background reading and also would be very suitable for an advanced machine learning course. The book by Witten and Frank is accompanied by a suite of Java-implemented machine learning tools called Weka, which is freely available including source code.[2] While the style of presentation is more leisurely and less academic than Mitchell's, the Weka toolbox is eminently usable for lab assignments and student projects. In addition, the book covers some subjects (such as association rule learning and support vector machines) that are left uncovered by the other two texts.

2.1 Mitchell's Machine Learning

Mitchell's text follows the main paradigms in machine learning. Thus, one finds chapters on decision tree learning, neural networks, Bayesian learning, instance-based learning, genetic algorithms, rule learning, analytical learning, and reinforcement learning. In addition, there are more theoretical chapters on concept learning and the generality ordering (Chapter 2), statistics (Chapter 5), and computational learning theory (Chapter 7). The exposition is usually lucid, with many examples and step-by-step derivations of formulae.

Among the strongest part of the book I would count Chapter 2, which discusses the generality ordering underlying almost any form of symbolic learning from examples in detail, including the version space (the set of all hypotheses consistent with the examples seen so far), the representation of the version space in terms of its most specific and most general boundaries, the candidate elimination algorithm for updating the version space with a new example, and the conditions under which this algorithm converges to the intended hypothesis. While perhaps somewhat limited in practical value, this provides a nice conceptual introduction to more practical learning algorithms. At several points in the book Mitchell refers back to this chapter, e.g., when discussing the hypothesis spaces searched by decision tree or rule learning algorithms, or when showing that any hypothesis in the version space is a maximum *a posteriori* hypothesis in the Bayesian sense, if we assume a uniform prior distribution over hypotheses and deterministic noise-free data. Also, I liked chapter 5 on evaluating hypotheses and learning algorithms (although it would improve with some re-organisation of the material) and Chapter 13 on reinforcement learning, where the problem is to learn from indirect, delayed feedback (e.g., a win at the end of a sequence of moves in a game) rather than from labelled examples.

Perhaps inevitably in a book of this scope, there are some weaker parts. The chapter on Bayesian learning is over-ambitious with its 13 sections, and tries to cover learning of Bayesian belief networks without properly explaining what these networks are used for, i.e., probabilistic inference. Also, the chapter on rule learning is too sketchy

[1] From an internet search of undergraduate and graduate courses on machine learning and data mining that we recently conducted in Bristol, it appears that a majority of the courses use Mitchell's book as the main text.

[2] The URL is `http://www.cs.waikato.ac.nz/ml/weka/`.

on propositional learners like CN2, while the remainder on inductive logic programming could have been improved by dealing with one system only (e.g., FOIL) in some more detail, and throwing out the unnecessarily complicated part on inverting resolution. There is a separate chapter on analytical learning which focuses on Prolog-style partial evaluation, yet a complicated procedural description is given instead of the 4-clause Prolog meta-interpreter which does the same job. The most obvious omission is unsupervised learning (e.g., clustering).

These criticisms aside, this is a thoroughly usable and enjoyable textbook that has already proved to be a classic. There is some supporting material available, including a basic set of slides, from the author's website.[3]

2.2 Elements of Machine Learning

When I called Mitchell's book above a conventional textbook, I meant that in the best possible sense of the phrase: it aims at providing an accessible introduction to the state of the art in machine learning, with many pointers to the relevant literature. Langley is more ambitious: not content with "writing a text that simply reflected the main paradigms within the machine learning community", he "aimed for an organization that would cut across the standard paradigm boundaries, in an attempt to cast the field in a new, and hopefully more rational, light" (p.ix). The material is cleverly organised, reflecting the central role of knowledge representation in learning. Thus, a separation is made between learning simple representations such as logical conjunctions (Chapter 2), threshold concepts including *m*-of-*n* concepts and perceptrons (Chapter 3), and competitive concepts including instance-based learning and probabilistic concepts (Chapter 4) on the one hand, and learning hierarchical organisations of such simple representations on the other (Chapters 5 to 7). The latter include decision lists (ordered lists of e.g. simple conjunctive rules, where the first rule that fires is chosen), inference networks (including logical theories, multi-layer neural networks, and probabilistic networks), and concept hierarchies (including decision trees and unsupervised cluster hierarchies, which are not covered by Mitchell). Throughout the text, emphasis is placed on high-level algorithms, given as pseudo-code, that can be instantiated to yield more familiar instances from the literature. The use of the same set of example problems greatly helps in pointing out the similarities and differences between the approaches, and illustrates the care with which the author has composed the text. The remaining chapters cover more advanced material, including numeric prediction (regression), learning transition networks such as hidden Markov models, and the acquisition of search control knowledge and macro-operators. A brief chapter on prospects for machine learning concludes the book.

I have learned a lot from reading Langley's book, as it provides a beautifully crafted unifying view on machine learning from an established authority in the field. However, this kind of rational reconstruction of the field may not be fully appreciated by someone without any knowledge of machine learning, which is why I would have reservations in recommending it as the only textbook for an introductory machine learning course. One option is to accompany it with hands-on lab exercises introducing the student to

[3] `http://www.cs.cmu.edu/~tom/mlbook.html`

the available machine learning systems and toolboxes such as C4.5, CN2, MLC++, and Weka, but the book itself provides hardly any pointers to such implemented systems or to practical problems that can be solved with machine learning techniques. It is also not obvious that somebody who has only read this book will be able to read the scientific machine learning literature. In comparison with Mitchell's book, the most obvious omission is the lack of material on statistics, heuristics and evaluation methods that have become so important in contemporary machine learning research. Having said that, I warmly recommend this book to anyone with a basic understanding of machine learning, who is interested in a novel, coherent and insightful perspective on the field.

2.3 Data Mining with Java implementations

The book by Witten and Frank is the most recent of the three (although the year 2000 given as publication date in the colofon is not entirely accurate, since I remember browsing through a copy late 1999). It is clearly more up-to-date in its coverage of some topics that gained popularity relatively recently: association rule learning, support vector machines, and ROC curves for cost-sensitive learning. On the other hand, some approaches covered in the other two books are absent from this one: neural networks, genetic algorithms, reinforcement learning, Bayesian networks, and inductive logic programming. There is also a distinctly different flavour to the book: whereas Mitchell's book is an academic introduction to the field, and Langley's is a scholarly reconstruction, this book is a how-to guide with, as the authors put it, "a practical, non-academic, unpretentious approach". There are some marked passages with more technical material that may be skipped without loss of continuity. The writing style is deliberately casual but pleasant.

The organisation of the material is also different from either of the two books discussed above. Following an introductory chapter, there are chapters on the input to a machine learning algorithm, its output (trees, rules, etc.), the basic learning algorithms, quantitative evaluation of what has been learned, more detailed algorithms, engineering input and output (e.g., attribute selection, discretisation, and combining multiple models through bagging, boosting and stacking), and more details on the Java toolbox Weka. The concluding chapter surveys some of the expected future developments. While this organisation of the book makes sense, it also leads to a slight imbalance, as there is certainly a lot more to say about detailed algorithms (Chapter 6, 72 pages) than about datasets as such (Chapter 2, 20 pages).

All main algorithms discussed in the book (decision tree learning, rule learning, support vector machines, instance-based learning, clustering, association rule learning) are also (re-)implemented in the Weka toolbox. The availability of this toolbox is clearly a great asset, and I am convinced that Weka will become very popular among machine learning lecturers, students, and practitioners (especially now that MLC++ has been turned into a commercial product and its public-domain version is no longer supported). On the other hand, I was surprised to find not, e.g., CN2 or Ripper as the rule learning algorithm of choice, but a rather more obscure one which constructs one rule at a time from a partial decision tree, extracting from it the path leading to the leaf with the largest number of instances. While the authors report accuracies as good as C4.5, I would have liked to see an implementation of a more mainstream rule learning

algorithm alongside this one. Another minor quibble with Weka: it is unclear to me why the authors chose to introduce yet another file format (the "ARFF" format) for datafiles, where they could have chosen, e.g., the widely-used C4.5 format.

All in all, this is another excellent introductory book on the topic. It is an primarily a text for self-study; it is not impossible to use it as a textbook, but in my opinion this would require quite some re-organisation of the material (the authors give no suggestions for this, nor do they provide supporting teaching material). Personally, I would suggest using Mitchell's book as the main textbook for an introductory machine learning course, with Witten and Frank's book as background reading and the lab projects designed around the Weka toolbox, and use Langley's book as one of the texts in an advanced topics course, perhaps letting the students study and present some of the more advanced material. But there is no doubt that all three books deserve a prominent place in the library of anyone who takes machine learning seriously.

2.4 Data Mining techniques for marketing and sales

From the business perspective, data mining has developed into something of a hype. While commercial interest in data mining is in itself a good thing, as it generates interest in and funds for data mining research, the hype-factor will undoubtedly lead to wild expectations and subsequent disappointments and loss of interest (similar to the expert systems hype of the 1970s and early 80s). There is a bewildering range of business-oriented books on data mining (Amazon reported 190 hits when I keyed in 'data mining', and Barnes and Noble 111). However, some of these books are good reads, even for a seasoned machine learning researcher, because they are problem-oriented rather than technique-oriented and provide a wealth of practical and sometimes very interesting data mining problems, some of which can fuel novel research. In this context, I would like to mention Michael Berry and Gordon Linoff's *Data mining techniques for marketing, sales, and customer support* (John Wiley, 1997) which has been recommended to me by several people. I can hardly improve upon the following short review submitted to `www.amazon.com` by an eminent machine learning researcher from Australia, who writes that "[the authors'] descriptions of the techniques are clear and accurate, and the case studies provide excellent models. The book is very well written and has a comprehensive index."

The book discusses seven techniques in relative detail: clustering, instance-based learning, market basket analysis (association rule learning), neural networks, genetic algorithms, decision trees, and link analysis (the latter is not really a learning technique, but a collection of standard computer science algorithms for analysing graph data). Each technique is analysed in terms of their strengths and weaknesses, and when to apply them. In addition, the authors discuss data warehousing and on-line analytical processing, and various other topics such as measuring effectiveness of data mining techniques, and data mining methodology. In the latter chapter, we find the following passage which in my opinion clearly illustrates that this is a very well-written book:

> "People often find it hard to understand why the training set and test set are "tainted" once they have been used to build a model. An analogy may help: Imagine yourself back in the 5th grade. The class is taking a spelling test.

> Suppose that, at the end of the test period, the teacher asks you to estimate your own grade on the quiz by marking the words you got wrong. You will give yourself a very good grade, but your spelling will not improve. If, at the beginning of the period, you thought there should be an 'e' at the end of "tomato", nothing will have happened to change your mind when you grade your paper. No new data has entered the system. You need a test set!
>
> Now, imagine that at the end of the test the teacher allows you to look at the papers of several neighbors before grading your own. If they all agree that "tomato" has no final 'e', you may decide to mark your own answer wrong. If the teacher gives the same quiz tomorrow, you will do better. But how much better? If you use the papers of the very same neighbors to evaluate your performance tomorrow, you may still be fooling yourself. If they all agree that "potatoes" has no more need of an 'e' then "tomato", and you have changed your own guess to agree with theirs, then you will overestimate your actual grade on the second quiz as well. That is why the evaluation set should be different from the test set." [3, pp.76–7][4]

I wish that all computer science textbooks were written like this.

2.5 Other introductory Data Mining books

The final introductory texts I would like to mention take a slightly different perspective, by including the preparation of data as an essential step in mining large volumes of data. Sholom Weiss and Nitin Indurkhya's *Predictive data mining: a practical guide* (Morgan Kaufmann, 1998) aims at providing a roadmap for developing data mining applications. In slightly over 200 pages it covers data preparation, data reduction, and learning. It also compares different data reduction and mining methods on experimental data, and discusses some case studies regarding the data preparation and reduction required, and the most suitable mining methods. Apart from some statistical notation (there is a separate chapter explaining the necessary statistical background) the exposition is kept non-technical. For instance, the authors refer to the one-table data format most machine learning methods assume as the "spreadsheet data format", and Boolean attributes are called "true-or-false variables" (the term 'Boolean' does crop up occasionally, however). If you are willing to tolerate a mild form of hype ("The big data revolution has begun") and want to know more about the practical aspects of mining large volumes of data, this might be the book for you. There is software available at `http://www.data-miner.com`, which has to be purchased separately.

A lighter introduction to the subject is provided by Pieter Adriaans and Dolf Zantinge's *Data mining* (Addison-Wesley, 1996). The authors are directors of Syllogic, an IT company specialising in systems and data management. The book pays particular attention to setting up a data mining environment, illustrated by real-life applications of data mining developed by Syllogic. The intended readership is described as "general

[4]The authors are referring, of course, to a recent USA vice-president who famously, when visiting a school, corrected a pupil who had written 'potato' on the blackboard by adding an 'e' at the end.

management and IT managers", and the tone is fairly pedestrian – although the authors clearly have a predilection for the more philosophical side of things, leading to a sometimes curious mix of practical advice and interesting asides (surely this is the only book for managers containing a definition of Kolmogorov complexity?). Nevertheless, this is a good book to read if you want to hear the data mining story from the horse's mouth.

3 Support Vector Machines

Support Vector Machines (SVMs) are a good example of the way in which current machine learning research combines ideas from different research areas. SVMs were introduced in the early nineties by Vapnik and co-workers [4] – bringing together ideas that had been around since the 1960s – and the topic has developed into a very active research area. Support Vector Machines combine two key ideas. The first is the concept of an *optimum margin classifier*, which is a linear classifier which constructs a separating hyperplane which maximises the distance to the training points. While linear classifiers date as far back as Rosenblatt's perceptron, the important point here is maximisation of the margin. This turns the under-specified learning problem into an optimisation problem (without local optima) and has been shown to give very good generalisation performance. Margin maximisation provides a useful trade-off with classification accuracy, which can easily lead to overfitting of the training data, and thus poor performance on test data. As a consequence, SVMs are well-suited to deal with learning tasks where the number of attributes is large with respect to the number of training examples. In general the optimum margin hyperplane will be a linear combination of the input vectors; *support vectors* are those training examples which obtain a non-zero coefficient, i.e., the ones that lie closest to the separating hyperplane.

The second key concept underlying SVMs is the concept of a *kernel*. In its simplest form, a kernel is a function which calculates the dot product of two training vectors. This dot product arises because during training we need to evaluate whether each training vector is correctly classified by the current hypothesis (a weight vector which is a linear combination of training vectors as described above). Many linear classifiers can be described in a dual form where the data only appears through the Gram matrix which contains the dot products of each pair of training vectors. Intuitively, this dot product expresses the similarity of the two training points in terms of the given attributes (which are, for the moment, assumed to be numerical). Now, consider the use of a feature transformation, which reformulates the input vectors in terms of new features. Provided we have a way of calculating dot products in feature space, this leaves the linear classifier unaffected. Kernels calculate these dot products in feature space, often without explicitly calculating the feature vectors, operating directly on the input vectors instead. For instance, a kernel which calculates the n-th power of the dot product of the input vectors is equivalent to the dot product of feature vectors whose components are proportional to products of n attribute values. So, while the implicit feature vectors may have much higher dimensionality than the input vectors, this doesn't incur computational overhead, nor do we suffer from overfitting if an optimum margin classifier is used. Also, notice that the resulting classifier is linear in feature

space but not necessarily in input space.

Because of the similarity between the optimum margin classifier and the perceptron algorithm, Support Vector Machines are often seen as an extension of neural networks. However, whereas neural networks can only indirectly incorporate background knowledge through their topology and initial choice of weights, SVMs offer a much more sophisticated mechanism to incorporate domain knowledge by means of the kernel. In a protein sequence domain, for instance, similarity between two sequences is measured in terms of their common sub-sequences (allowing gaps which result from mutation). There are well-known dynamic programming algorithms for calculating the longest common sub-sequences of two given sequences, which run in time proportional to the product of the lengths of the input sequences. These algorithms can be used to define a kernel whose implicit feature vector counts the number of occurrences of all possible subsequences of length n – a feature space clearly too large to be constructed explicitly! This example illustrates several points: (1) SVMs can deal with non-numerical, symbolic data; (2) SVMs have, to some extent, already been upgraded to first-order representations such as lists; (3) SVMs are not an off-the-shelf technology, but need to be engineered to fit the underlying domain; and (4) existing algorithms can be reformulated in terms of kernel methods. As another example of the last point, by taking a Gaussian kernel which operates on the difference between the two input vectors, one in fact implements radial basis function networks.

3.1 An introduction to Support Vector Machines

Some of the points made above have been taken from the book *An introduction to Support Vector Machines and other kernel-based learning methods* by Nello Cristianini and John Shawe-Taylor (Cambridge University Press, 2000). This slim volume (less than 200 pages) gives an excellent introduction to this rapidly developing field. The book is well-written and contains many pointers to the literature (the bibliography contains 180 references). In addition, there is a website `www.support-vector.net` "which will be kept up to date with new work, pointers to software and papers that are available on-line", as the authors remind us at the end of every chapter. The book consists of 8 chapters, of which the theoretical Chapters 2–6 deal with linear learning machines, kernel-induced feature spaces, generalisation theory, optimisation theory, and Support Vector Machines, Chapter 7 deals with implementation techniques, and Chapter 8 covers selected applications of SVMs. Each chapter ends with some exercises (a reasonable starting point, but probably not sufficient for the classroom) and a section on further reading and advanced topics.

Following a general introduction, Chapter 2 explores the perceptron algorithm as a simple linear classifier, developing its dual form for use with kernels. Other issues addressed include multi-class discrimination and linear regression. Chapter 3 gives a very clear introduction to kernels and feature spaces, including a fairly technical discussion of necessary conditions for a function to be a kernel (Mercer's conditions), how to build kernels from kernels (closure properties) and from features (the above subsequence kernel was taken from this chapter, following Watkins [19]). Chapter 4 explores the generalisation performance of margin-based classifiers in the context of the PAC learning model (probably approximately correct learning) and Vapnik-Chervonenkis (VC)

theory, which relates generalisation performance to the capacity of a hypothesis space. The PAC model is distribution-free, in the sense that the learner is required to perform well without knowing the distribution governing the selection of training and test examples. This model is often felt to be too restrictive, as many learnability results are negative. In contrast, SVMs are designed to take advantage of benign distributions: the margin measures how helpful the example distribution is. This leads to data-dependent error bounds that involve the margin but not the dimensionality of the feature space.

It has already been mentioned that Support Vector Machines are optimisation algorithms, and Chapter 5 is devoted to optimisation theory, including Lagrange multipliers and quadratic programming. Chapter 6 then puts everything together: "Support Vector Machines are a system for efficiently training the linear learning machines introduced in Chapter 2 in the kernel-induced feature spaces described in Chapter 3, while respecting the insights provided by the generalisation theory of Chapter 4, and exploiting the optimisation theory of Chapter 5". This chapter also deals with soft-margin classifiers, which are able to deal with data that are not linearly separable in feature space because of noise. Chapter 7 deals with implementation techniques, concentrating on Platt's Sequential Minimal Optimisation (SMO). Finally, Chapter 8 describes a number of successful applications of SVMs in the domains of text categorisation, image recognition, hand-written digit recognition, and bioinformatics. The results are generally impressive, but the chapter does not contain enough detail to really understand what is going on. The book is concluded by two appendices containing pseudo-code for the SMO algorithm and background mathematics, an extensive bibliography, and a not so extensive index.

On the whole, the book has a mathematical flavour, as both authors come from computational learning theory. That being said, I found the material generally accessible and well-explained. In particular the first three chapters already equip the reader with a good understanding of the general issues, but there is also a wealth of more advanced material throughout the book. I would have liked to see a better separation between introductory and advanced material (e.g., starred subsections). There is also some material towards the end of Chapters 3, 4 and 6 connecting Bayesian learning with Gaussian processes which is not well explained and seems unrelated to the core of the book. As a final indication that the organisation of the material could have been improved, I mention the fact that the term 'support vector' is only explained on p.97, half-way through the book! In other words, the book gives a rational reconstruction of the topic, which is fine for somebody already familiar with it but probably less appropriate for a novice. But these are minor quibbles: this is an excellent and timely introduction to an exciting subject, with all the potential of becoming a standard reference.

3.2 Advances in kernel methods

More advanced material can be found in *Advances in Kernel Methods: Support Vector Learning*, edited by Bernhard Schölkopf, Christopher Burges and Alexander Smola (MIT Press, 1999). The book arose from a workshop on SVMs held at the 1997 Neural Information Processing Systems conference. The book starts off with a nice and short (15 pages) introductory chapter on Support Vector learning, and a 6-page roadmap to

the remaining papers in the collection, both written by the editors. The book consists of 4 parts: Theory, Implementations, Applications, and Extensions of the algorithm. The first part opens with a paper by Vladimir Vapnik, somewhat enigmatically entitled "Three remarks on the support vector method of function estimation". The three remarks concern possible extensions of the method (the chapter could therefore also have been included in Part 4). The first of these is related to the geometry of the support vectors:

> "...there may exist more advanced models of generalization then that based on maximization of the margin. The bound on the error depends on the expectation of the ratio of two random variables: the radius of the sphere that contains the support vectors, and the margin. It is quite possible that by minimizing this ratio one can control the generalization better than by maximizing the margin. Note that in high dimensional feature spaces, where the SV machine constructs hyperplanes, the training set is very sparse and therefore the solution that minimizes this ratio can be quite different from the one that maximizes the margin." [17, p.35]

The second possible extension is transductive inference, which improves prediction performance on a given test set by taking the distribution of test examples into account; and the third is to use support vector regression for estimating conditional probabilities or densities (in Vapnik's words: to use the solution of a simple problem to solve a more difficult one). Other chapters in the Theory part study generalisation performance, Bayesian voting schemes, and geometry and invariance in kernel-based methods, among others. As might be expected in a part dealing with theory, some of these chapters are very technical.

The second part of the book deals with Implementations, and contains three papers. The first, by Linda Kaufman, deals with methods for solving the quadratic programming (QP) problem underlying support vector learning. The second, by Thorsten Joachims, describes the techniques used in SVM^{light}, an SVM implementation capable of dealing with large training sets, with memory requirements linear in the number of training examples and support vectors, and a least-recently-used caching strategy for kernel evaluations.[5] Finally, John Platt's chapter details his Sequential Minimal Optimization (SMO) training algorithm, which the editors generously describe as SVM's backpropagation:

> "Here is an algorithm that is easy to understand, easy to implement (the chapter even lists pseudocode), and trains an order of magnitude faster (in some cases) than a conjugate-gradient implementation of a QP-optimizer. It is our hope that, due to its simplicity and speed, this algorithm will make SVMs accessible to a much larger group of users, and that the encouraging results will withstand the test of a wide range of data sets." [16, p.20]

The SMO algorithm operates by heuristically breaking the optimisation problem down into a series of smallest possible QP problems, each involving two Lagrange multipliers, which are then solved analytically. Again, the memory requirements are linear in

[5] SVM^{light} is available at `http://www-ai.cs.uni-dortmund.de/svm_light/`.

the training set size. Note that the pseudocode listed in this chapter is the same as that included in an appendix to [5]. All three chapters demonstrate the improvements their optimisations yield on various benchmark problems.

The third part of the book concerns applications such as dynamic reconstruction of chaotic processes and time series prediction. This part also contains a nice short paper by Ulrich Kreßel about using SVMs for multi-class prediction, not – as is usually done – by training on each class against all others, but by pairwise classification. The author shows empirical improvements on the often-used handwritten digit recognition dataset. This paper is one of the few in the volume with direct relevance outside the SVM approach – in fact, I wouldn't be surprised if it had been tried before in other machine learning approaches as well (the author does not discuss related work).

Part 4 deals with extensions of the algorithm. Here, I would like to mention two chapters: Kristin Bennett's work on decision trees where the decisions are SVMs; and Schoölkopf *et al.*'s chapter on Kernel Principal Component Analysis, i.e., computing principal components in high-dimensional feature spaces that are related to input space by a non-linear map. The book is concluded by an extensive bibliography and a basic index. Generally, this is a well-produced book (witness the combined bibliography which requires a lot of effort from both authors and editors). Perhaps inevitably in a book of this kind, there is some repetition of material. Also, I am not quite sure how well this book succeeds in covering all and only most important developments in SVMs (as its title suggests). On the whole, however, this is a very useful resource for people wanting to find out what current-day SVM research is about.

3.3 Statistical learning theory

For a really in-depth study of the support vector method and associated techniques, there is Vladimir Vapnik's book *Statistical learning theory* (John Wiley & Sons, 1998). I will not attempt a review of this mighty volume, but I would like to draw the reader's attention to the final chapter. Rather than collecting historical and bibliographical remarks at the end of each chapter, they have been assembled together here, resulting in a personal, very readable and highly instructive overview of the development of the field, from Popper, Kolmogorov and Fisher as intellectual fathers, via perceptrons, VC theory, Ockham's razor, Bayesian inference, and backpropagation, to support vectors, kernel methods, and an epilogue proposing to concentrate on inference from sparse data as the key problem, taking into consideration "physical factors" which hold for our specific world, in which we have to solve our applied tasks.

I am convinced that the ideas embodied in Support Vector Machines, and kernel methods more generally, will prove to be very fruitful for the further development of machine learning as a field of research. Kernels allow non-symbolic learning methods to deal with symbolic and structured data. While SVMs are black-box classifiers, they assign weights to features in high-dimensional feature spaces and thus could be used for feature construction in combination with symbolic approaches such as inductive logic programming. Also, because of the close connection between similarity (as measured by a kernel) and distance, kernels could be used in distance-based clustering and classification methods. In this respect I was slightly disappointed that the issue of distances, and more generally the intuitive interpretation of what a kernel computes (beyond the

mathematical intuition), is almost completely ignored in the books referred to above. As a more general critique, the development of some of the material is sometimes overly (and unnecessarily) mathematically involved, and some of the terminology (e.g. 'structural risk minimisation') is fairly obscure for somebody coming from a symbolic machine learning viewpoint. Nevertheless, I encourage anyone interested in current developments in machine learning to take a look at the SVM literature, and the books I mentioned provide good starting points.

4 Graphical probabilistic models

Just as kernels can be used to add logical structure to non-symbolic learning methods, graph representations can be employed to combine logical and probabilistic knowledge. The resulting models are usually called graphical probabilistic models, or *graphical models* for short. Graphical models became popular after Judea Pearl published his seminal book *Probabilistic reasoning in intelligent systems* in 1988 [14], and have been a very active research topic for the last decade.

In general, a probabilistic model is some encoding of a joint probability distribution over a set of random variables. Given such an encoding, probabilistic inference then amounts to calculating the conditional distributions of the dependent variables, given the values of the observed variables and marginalising out the remaining non-observed variables. Of course, such brute-force inference is exponential in the number of random variables, and therefore intractable in the general case. Most graphical models explicitly encode independence assumptions among subsets of the variables, allowing decomposition of the joint probability distribution. For instance, according to the chain rule of probability $p(ABC) = p(A)p(B|A)p(C|AB)$; if we know that C is independent of B given A, this reduces to $p(ABC) = p(A)p(B|A)p(C|A)$. This independence assumption can be graphically depicted as the directed acyclic graph $B \leftarrow A \rightarrow C$. Assuming that all variables are Boolean, this reduces 7 entries in the joint probability distribution of ABC (the 8th one can be determined from the requirement that they all sum to 1) to 1 entry for A, 2 entries for $B|A$ (1 for each value of A) and likewise 2 entries for $C|A$).[6]

A variety of graph representations is used to encode independence assumptions, and the terminology is diverse and somewhat bewildering. Directed acyclic graphs are referred to as *Bayesian networks*, *belief networks*, *causal networks* or *influence diagrams*, while their undirected cousins are called *Markov random fields*. Notice that a totally unconnected graph represents independence among all variables and thus the biggest computational gain, whereas a fully connected graph (relative to an ordering of the variables, i.e., each node is connected to all its descendants in the ordering) simply represents the chain rule of probability. The complexity of probabilistic inference in graphical models is related to the number of (undirected) paths that exist between nodes

[6] Notice that we can equivalently write $p(ABC) = p(B)p(A|B)p(C|A)$ or $p(ABC) = p(C)p(B|A)p(A|C)$, corresponding to the directed acyclic graphs $B \rightarrow A \rightarrow C$ and $B \leftarrow A \leftarrow C$, respectively. In general, the same set of independence assumptions can be encoded by different graphs – independence is encoded by *missing* arrows, rather than dependence being encoded by arrows present. Causal interpretation of directed graphical models therefore requires additional knowledge, e.g., an ordering of the variables (more on causality in Section 4.3).

(a particularly efficient subclass of Bayesian networks is the *Markov chain*, imposing a linear dependency order on the variables). Many inference algorithms for graphical models convert the original graph into some kind of tree, whose nodes represent sets of random variables whose distributions cannot be decomposed (cliques). A major advantage of graphical models is that we can exploit well-known graph algorithms in probabilistic inference.

4.1 Machine learning and digital communication

A compact but comprehensive introduction to different kinds of graphical models, and probabilistic inference in those models, is given by Brendan Frey in the first two chapters of his book *Graphical models for machine learning and digital communication* (MIT Press, 1998). Frey uses his own construction *factor graphs*, which are bipartite graphs with one type of node for random variables and another for conditional probability distributions, to explain both Bayesian networks and random Markov fields. The book is of particular interest because it draws connections between classification, unsupervised learning, data compression, and channel coding. Such connections are not new: ever since William of Ockham's dictum "entities should not be multiplied beyond necessity" people realised that it often pays off to choose a (syntactically) simple hypothesis, culminating in approaches such as minimum description length and minimum message length. Also, the idea that efficient codes employ statistical knowledge about the source, by assigning shorter codes to more frequent messages, has been well-known since Shannon's ground-breaking work on information theory. The interesting aspect of Frey's book is not the use of probability theory as the unifying framework, but the use of graphical models.

For example, in Chapter 6 of his book Frey considers the problem of channel coding, where a binary vector transmitted over a non-ideal channel is corrupted by noise (additive white Gaussian noise for simplicity). Problems associated with channel coding are to devise an appropriate code that allows error detection and correction (by allowing sufficient distance between the codewords, i.e., adding redundancy) and to devise a decoder that performs the error correction in a near-optimal way. While many channel decoders are algebraic, quantising the corrupted codeword and algebraically determining the codeword that is closest in Hamming distance, probabilistic decoders aim to make as much as possible of the unquantised channel output. Similar to algebraic decoders which take advantage of the algebraic structure of the code, probabilistic decoders employ probabilistic structure, and this is where graphical models come in. For instance, a Hamming code with n input bits and p parity bits can be translated into a Bayesian network consisting of two parts: one part connects the n unobserved input variables to n observed output variables where the conditional probabilities are Gaussian, and the other part connects the n input variables to p unobserved variables for the parity bits, which are then connected to p observed (and corrupted) output variables. While for simple Hamming codes exact probabilistic inference can be used to compute the most likely codeword, in the case of more elaborate codes approximate probability propagation algorithms need to be used. Codes that are analysed in this chapter from the perspective of graphical models include convolutional codes and the rather amazing turbocodes, which come very close to Shannon's theoretical limit.

The outline of the book is as follows. The first chapter gives succinct introductions to pattern classification, unsupervised learning, data compression, and channel coding using a uniform conceptual framework and notation. It also introduces graphical models and gives some examples of the kind of problems to which the book is devoted. Chapter 2 deals with exact and approximate inference in graphical models, including Monte Carlo methods, variational inference, and Helmholtz machines (which couple a generative Bayesian network with a recognition network meant to quickly recognise the values of the hidden variables). Chapter 3 is devoted to classification tasks. Unconventionally, most of the approaches studied in this chapter learn one model for each class of training data, which may ignore similarities between classes (I found the justification of this approach less than convincing). The Bayesian network architectures considered include autoregressive networks (while these are fully connected relative to an ordering of the variables and thus do not encode any independence assumptions, this is balanced by the absence of hidden variables), multiple-cause networks which postulate a hidden layer of unobserved variables connected to all observed variables, stochastic Helmholtz machines, hierarchical networks, and ensembles of networks. These methods are experimentally tested on the well-known US Postal hand-written digits dataset, and compared with other machine learning methods (naive Bayes, CART, and k-nearest neighbour). The best performing methods were the autoregressive classifier and an ensemble of stochastic Helmholtz machines (with the former requiring much less training time). The other machine learning methods performed poorly on this task, possibly because on this dataset the use of one model per class is advantageous – but this is not analysed. Incidentally, one of the things I learned from this chapter is the existence of the DELVE system (data for evaluating learning in valid experiments), an environment for evaluating learning algorithms [15].

Chapter 4 studies unsupervised learning tasks. In the context of probabilistic models, such tasks are naturally viewed as tasks in which there are unobserved variables which 'cause' the data, whose values corresponds to clusters. Stochastic Helmholtz machines are trained using the wake-sleep algorithm on an artificial dataset involving noisy images containing horizontal and vertical bars. Other learning techniques employed are slice sampling and variational methods. Chapter 5 is concerned with data compression. If we have a Bayesian network approximating the probability distribution over the binary vectors to be encoded, this network can be used to compress the messages in a straightforward way. In case the network has hidden variables, this can be viewed as a multi-valued source code, where there are many codewords for each input vector depending on the values of the hidden variables. Frey shows convincingly how auxiliary data can be used to select the codeword, achieving a better communication rate than when always the shortest codeword would be chosen, as demonstrated by two sets of experiments. Chapter 6 (with its 42 pages the longest in the book) has been described above. In the final chapter, Frey sketches what he sees as the main research directions in machine learning and digital communication, using citations of some of the experts in the field.

Books which grew out of PhD dissertations do not always make good research monographs, but this one does. In less than 200 pages Frey manages to convey a strong sense of enthusiasm and excitement about recent advances in the use of graphical probabilistic models, to which he himself has contributed significantly. This is not a text-

book, and depending on the reader's knowledge some further background reading may be necessary, but those interested in the state of the art in graphical models and their use in machine learning and digital communication should certainly read this book. It is well-written and beautifully designed. Software used for some of the experiments in the book can be found at `http://mitpress.mit.edu/book-home.tcl?isbn=026206202X`.

4.2 Learning in graphical models

One volume which provides a lot of background on graphical models is *Learning in graphical models*, edited by Michael Jordan (originally published in hardback by Kluwer Academic Publishers in 1998, and as a paperback by MIT Press in 1999). The volume arose from the proceedings of the International School on Neural Nets organised in 1996. The connection between graphical models and neural networks may not seem an obvious one, but is in fact explored in several chapters in this volume. A neural network can be treated as a graphical probabilistic model by associating a binary random variable with each node and interpreting the activation of the node as the probability that the associated random variable takes one of its two values.

The book consists of four parts: Inference (8 chapters), Independence (2 chapters), Foundations for Learning (2 chapters), and Learning from Data (11 chapters). With one or two exceptions, the chapters within each part are alphabetically ordered on first author. There are 4 tutorial chapters in Part I and another one in Part III, which are all of high quality and make this book a very useful resource. Most of the remaining chapters are much more technical and concentrate on fairly specific topics, but some others are, although more advanced than the tutorial introductions, of more general interest. Personally I would have preferred a different organisation of the material, since the present one is clearly imbalanced and seems to lack an inherent logic (for what it's worth, I would probably have started with a tutorial part, followed by an advanced theory and algorithms part, followed by a learning part – possibly sub-divided into supervised and unsupervised learning). Also, I don't think the title is wholly appropriate, since only about half of the papers directly concerns learning. Nevertheless, the tutorials can be strongly recommended to anyone desiring to have a deeper understanding of the main issues in graphical models, and the reader is likely to find a few other chapters of direct interest. Below, I will describe my choice of the best chapters, including the tutorials.

The first two chapters in Part I are *Introduction to inference for Bayesian networks* and *Advanced inference for Bayesian networks*, both by Robert Cowell. The first tutorial covers fairly basic material, leading up to 'moralisation' (turning the directed graph into an undirected one, adding links between parents of the same node), triangulation of the moral graph, and construction of the junction tree which has a node for each clique in the triangulated graph. Inference algorithms on the junction tree are local, in the sense that cliques only have to agree on the marginal probability that they assign to nodes they have in common.[7] The second tutorial covers more advanced material, including sampling, most likely configurations, and networks with normally distributed

[7] Junction trees are called *join trees* in relational databases.

continuous variables. The split between the two chapters is somewhat arbitrary, and for all practical purposes they constitute a single, comprehensive introduction to inference in Bayesian networks.

Inference algorithms for Bayesian networks are often portrayed as message passing algorithms. In her chapter in this volume, Rina Dechter proposes *bucket elimination* as a unifying framework for probabilistic inference.[8] The framework accommodates algorithms for belief updating, finding the most probable explanation, the maximum a posteriori hypothesis, and the maximum expected utility.

Exact inference in graphical models is often infeasible. In their tutorial chapter, Jordan, Ghahramani, Jaakkola and Saul provide an introduction to variational methods for obtaining upper and lower bounds on local and global probabilities. The why and how of variational methods is explained particularly clearly, concentrating on the underlying principles rather than just giving the techniques.

The penultimate chapter in Part I, by David MacKay, gives an introduction to Monte Carlo methods for generating samples from a given probability distribution and estimating expectations of functions under that distribution. Techniques covered include importance sampling, rejection sampling, the Metropolis method, and Gibbs sampling.

Part III includes the well-known tutorial by David Heckerman on learning with Bayesian networks, which I believe started life as a Microsoft research report, and since then has been published in various guises. There is a distinction between learning the parameters of a given Bayesian network from data, and learning the structure as well, which is a much harder problem. Heckerman does discuss the second problem (from a Bayesian perspective, i.e., updating a prior probability distribution over possible model structures given the data), but as this is still a very active research area the latest results are not included (the most recent references are from 1996). Heckerman includes a brief discussion of learning causal relationships, a fairly controversial subject (see below).

Part IV contains no tutorial chapters and consists mostly of fairly short, technical chapters. Of the longer chapters I would like to highlight the following two. *Latent variable models* by Christopher Bishop investigates modelling of continuous hidden variables. One interesting technique studied is a probabilistic version of principal component analysis, which seeks to construct a system of orthonormal axes on which the data (under projection) has maximum variance. Bishop demonstrates how this can be applied to clustering, and also suggests a data visualisation method. Nir Friedman and Moises Goldszmidt propose a technique for learning the parameters in a Bayesian network by decomposition. Recall that a Bayesian network requires a probability distribution for each node in the network, conditional on its parents; for a node with n parents, such a table has 2^n entries. Describing such tables by e.g. decision trees does not only compress them, it also leads to probabilistic models with fewer parameters which therefore can be more reliably estimated, and which moreover are more accurate. This is supported by experiments with data generated from three target networks.

All in all, this volume combines aspects of a textbook (the excellent tutorials), a 'Readings in graphical models' (the journal-quality papers I mentioned), and a con-

[8] A version of this chapter has been published as [6].

ference proceedings. On the one hand, this leads to a certain imbalance in form and content. On the other hand, it is obvious that a large audience will find at least part of the book useful. In conclusion, I would like to mention the index which is fairly extensive for a collection of this kind.

4.3 Graphical models and causation

A volume which contains more controversial material is *Computation, causation, and graphical models*, edited by Clark Glymour and Gregory Cooper (AAAI Press and MIT Press, 1999). Whereas standard classification-oriented machine learning is concerned with predicting some features from other features, causal prediction aims at predicting the *change* in some features that will result if an intervention changes other features. This is controversial because the traditional view is that causal inference from statistical data only exhibiting correlations is impossible without performing randomised controlled experiments.[9] Nevertheless, the authors contributing to this collection explore methods of causal prediction that are based on the assumption that the unknown causal structure can be adequately modelled by graphical representations similar to the Bayesian belief networks discussed earlier (but now with a causal interpretation). Such approaches search among large numbers of model structures, and can be called 'data mining for causal relationships'. Many statisticians distrust search for model structures, claiming that it will lead to overfitting ('torturing the data until they confess' – the term 'data mining' originated as a pejorative term in statistics) and relying on parametric methods, where the model is chosen in advance and only its parameters are estimated from the data.

The book starts with an introductory chapter by Gregory Cooper, followed by eighteen chapters organised in five parts: Causation, representation and prediction (two chapters), Search (four chapters), Controversy over search (four chapters), Estimating causal effects (three chapters), and Scientific applications (five chapters). The introductory chapter includes, besides the usual material on directed acyclic graphs and the (causally interpreted) independence assumptions they encode, a discussion about the nature of causality (i.e., local in time and space, and relative to the variables in a model), and an overview of search methods and algorithms for causal discovery. Broadly, these fall into two groups: constraint-based methods, which use tests of conditional (in)dependence to constrain the causal relationships among model variables, and Bayesian methods, which compute the probability that causal relationships exist.

In Part I, Chapter 2 by Spirtes et al. deals with prediction and experimental design with graphical causal models. It concentrates on a framework for experimental design due to Rubin, in which some features are *dispositional*, i.e., propensities to give a response to a treatment (e.g., fragility is a disposition to break if struck). A random variable is associated with each dispositional feature and each value of the relevant treatment variable. The next chapter by Judea Pearl discusses graphical tools for reasoning about causal relationships, foreshadowing a forthcoming book on causality.

[9] If the value of an attribute A is randomised, the randomising device is the sole cause of A. It is customary to make a causal independence assumption, which states that if A does not cause B, and B does not cause A, and there is no other variable which causes both A and B, then A and B are independent. As the second and third case are excluded in a randomised experiment, correlation between A and B implies that A causes B.

Part II is devoted to search for causal model structures. Heckerman, Meek and Cooper describe a Bayesian approach to causal discovery, using priors for model structures and model parameters. Scheines et al. describe their PC algorithm (implemented in the Tetrad II system), which applies constraint-based search, as well as a different kind of causal model called a *structural equational model*, which includes possibly correlated error terms. Chapter 6 by Spirtes, Meek and Richardson investigates how to deal with hidden variables, which are particularly consequential in causal modelling as two variables which share a hidden cause may appear independent if the cause is not taken into account. Chapter 7 by Richardson and Spirtes deals with discovery of (cyclic) feedback models. I found that this chapter repeated (sometimes verbatim) quite some material from Chapters 5 and 6, which could easily have been avoided given that its authors contributed to all three chapters.

Part III, entitled 'Controversy over search', is certainly one of the most entertaining and informative parts of the book. It starts with a chapter by James Robins and Larry Wasserman entitled 'On the impossibility of inferring causation from association without background knowledge' and is followed by a reply by Clark Glymour, Peter Spirtes and Thomas Richardson with the title 'On the possibility [...]'. This in turn is followed by a 'Rejoinder to Glymour and Spirtes' by the first two authors (note that Richardson has mysteriously disappeared), and concluded by a brief 'Response to rejoinder' by Glymour, Spirtes and Richardson. The editors summarise the discussion thus:

> "Robins and Wasserman say for any sample size there exist prior probabilities for which, conditional on the data, a true hypothesis about causation has small probability; they show this by describing a procedure for revising priors as a function of sample size so that the truth is driven towards a zero posterior probability as the sample size increases without bound. Glymour, Spirtes, and Richardson reply that the revision procedure is un-Bayesian, and that the priors needed are unrealistic. Robins and Wasserman reply that epidemiologists in fact have such priors. Glymour, Spirtes, and Richardson reply that if epidemiologists really used such priors, then their methods would be unreliable." [10, p.303]

Part IV contains three chapters on parameter estimation in causal models. This involves not only estimating parameters of the distributions governing the random variables in the models, but also estimating distributions arising from possible interventions. Finally, Part V is devoted to applications of causal discovery in various fields, including evolutionary biology, satellite data analysis, spectral data analysis, market analysis, and database mining. All of these applications make use of the Tetrad II system. The book concludes with an extensive 12-page index.

In conclusion, this book provides a fascinating overview of a relevant and rapidly evolving research area. Causal discovery is obviously central to many fields of science, requiring a multi-disciplinary approach, an open mind, and a will to confront controversy rather than swiping it under the carpet. All of these ingredients are present in this volume, making it an entertaining read. In addition, the book is beautifully produced. My only slight reservation is that, from the papers cited and the distribution of authors (several of whom contribute to four or even five chapters), one gets the impression

that the perspective provided is perhaps biased towards particular approaches (some pointers to work not covered in the book are provided in the preface).

5 Application areas

I would now like to (briefly) draw attention to three books that are concerned with the application of machine learning techniques in other fields of scientific inquiry: bioinformatics and ecology. For a machine learning researcher, such books are important for at least two reasons: they demonstrate that machine learning has gained respectability as a useful form of data analysis, and they provide new focal points of machine learning research.

The first book is *Bioinformatics: the machine learning approach* by Pierre Baldi and Søren Brunak (MIT Press, 1998). The authors define bioinformatics as the computational analysis of biological sequences, i.e., linear descriptions of protein, DNA and RNA molecules. Through concerted research efforts such as the human genome project there is now an abundance of data awaiting interpretation, including sequence classification, similarity detection, recognising protein coding regions in DNA sequences, predicting molecular structure and function, and reconstructing evolutionary history. One of the unifying forces behind bioinformatics is the use of sequences (i.e., lists) as the primary datastructure describing objects of interest. This also implies that standard machine learning algorithms are not immediately applicable, as they often assume that examples are described by fixed-size attribute-value vectors. Bioinformatics thus underlines the need for richer knowledge representation formalisms in machine learning, for instance the first-order logic representations employed in inductive logic programming. There are also clear analogies with speech recognition, natural language processing, and information retrieval.

An important vehicle in bioinformatics is the Hidden Markov Model (HMM), which is a probabilistic automaton generating a probability distribution over sequences, by specifying for each state transition probabilities to other states, and emission probabilities governing the symbol output in that state. A random walk through the HMM, from start state to stop state, generates one of several possible sequences of symbols (the adjective 'hidden' refers to the fact that one cannot infer the state sequence from the symbol sequence). There are a number of well-known algorithms for HMMs, including the Viterbi algorithm for determining the most likely state sequence that generated a given symbol sequence, the forward algorithm for generating a particular sequence, and the Baum-Welch algorithm for estimating the probability parameters from a training set of sequences (a special case of the Expectation Maximisation or EM algorithm). Each of these techniques is briefly explained in Chapter 7. The next chapter then gives a number of applications of HMMs in bioinformatics. I found this one of the most valuable chapters of the book, although without a certain sophistication in molecular biology it is not always easy to understand the salient points of the application. There is also a short chapter on stochastic grammars, a generalisation of HMMs (Chapter 11).

The other chapters in the book deal with machine learning techniques. However, I was a bit surprised to see, besides the Bayesian methods required for HMMs, only coverage of neural networks and a bit of genetic algorithms. In their preface, the au-

thors write that "an often-met criticism of machine learning techniques is that they are 'black box' approaches: one cannot always pin down exactly how a complex neural network, or hidden Markov model, reaches a particular answer". This is true for neural networks, but certainly less so for symbolic machine learning approaches like decision trees or rule learning. Quite possibly, the choice of neural networks is the fact that they provide an easily used 'off-the-shelf' technology: just specify the number of hidden nodes and off you go. Also, it does lend further unity to the book, since the authors offer a reconstruction of neural networks in terms of graphical probabilistic models (the same point was noted in Section 4.2 above). However, I believe that the time of off-the-shelf machine learning methods is over: machine learning methods will become increasingly domain-dependent, requiring close collaboration between domain experts and machine learning researchers, each of them having quite some understanding of the other field.

The book *Biological sequence analysis: probabilistic models of proteins and nucleic acids* by Durbin, Eddy, Krogh and Mitchison (Cambridge University Press, 1998) concentrates almost exclusively on HMMs and related methods, but does so in much greater depth. The style of the book is fairly technical, with emphasis on the probabilistic aspects, although there is discussion of practical sequence alignment problems and many methods are illustrated by results obtained on real data. The authors have provided scattered exercises (although many of them are of the type 'prove the step that we left out of the derivation of equation (X.Y)') and thus seemed to have had a textbook in mind. In the preface, they write: "Computational biology is an interdisciplinary field. Its practitioners, including us, come from diverse backgrounds, including molecular biology, mathematics, computer science, and physics. Our intended audience is any graduate or advanced undergraduate student with a background in one of these fields. We aim for a concise and intuitive presentation that is neither forbiddingly mathematical nor too technically biological."

Quite apart from its merits as a textbook, this book is a rich source of information on DNA sequence analysis. Compared with Baldi and Brunak's book, this one has much more detail on the methods and algorithms used, but less on the biological side of things. I would recommend Baldi & Brunak to those seeking a general introduction to bioinformatics (it also has an extensive listing of web resources), and Durbin *et al.* to those considering to make a serious study of sequence analysis.

Before concluding, I would like to mention the collection *Machine learning methods for ecological applications* (Kluwer Academic Publishers, 1999), edited by Alan Fielding. It is aimed at introducing machine learning methods to a readership of professional ecologists. Except for one chapter on equation discovery which is written by a group of machine learning researchers from Slovenia, all chapters have been written by ecologists and biologists discussing the application of machine learning to problems such as the identification of species, optimal mate choice, predicting species distributions and modelling landscape features. The breadth of machine learning techniques covered is considerable, and there is a wealth of ecological and biological case studies. While the machine learning methods required in ecological applications are perhaps less specialised than in the case of biological sequence prediction, there are issues, such as unequal misclassification costs, that require understanding of the ecological backgrounds of the problem.

6 Concluding remarks

Many approaches, developments and application areas have been left out of this review. Inductive logic programming is one of my own main areas of research; while I certainly believe the approach holds great promise for the future, I felt a review from a somewhat greater distance would be more beneficial. Other areas of active research include case-based reasoning and instance-based learning methods, where predictions are based on a distance metric which is used to retrieve training instances that are most similar to the newly observed one; constructive induction, which involves the construction of intermediate concepts or theoretical terms that are meaningful in the domain; cost-sensitive classification and subgroup discovery, where the emphasis is on identifying interesting sub-populations rather than classification accuracy; meta-learning, which is aimed at predicting which algorithm works best on a particular dataset; and multi-strategy learning, which combines different learning approaches to solve practical problems. Promising application areas include information retrieval, text and web mining, and data analysis in medicine. I am looking forward to future literature reviews and surveys of the many facets of machine learning.

Acknowledgements

I am grateful to Nada Lavrač for encouragement and suggestions which helped improve an earlier draft, to Edward Ross for assisting in the internet search for machine learning and data mining courses, and to Tony Cohn and Don Perlis for coaxing me into writing this paper. Part of this work was supported by the Esprit V project IST-1999-11495 *Data Mining and Decision Support for Business Competitiveness: Solomon Virtual Enterprise*.

References

[1] Pieter Adriaans and Dolf Zantinge. *Data mining*. Addison-Wesley, 1996.

[2] Pierri Baldi and Søren Brunak. *Bioinformatics: the machine learning approach*. MIT Press, 1998.

[3] Michael J. A. Berry and Gordon Linoff. *Data mining techniques for marketing, sales, and customer support*. John Wiley & Sons, 1997. `http://www.data-miners.com/`.

[4] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992.

[5] Nello Cristianini and John Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000. `http://www.support-vector.net/`.

[6] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113:41–85, 1999.

[7] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[8] Alan H. Fielding, editor. *Machine learning methods for ecological applications*. Kluwer Academic Publishers, 1999.

[9] Brendan J. Frey. *Graphical models for machine learning and digital communication*. MIT Press, 1998. `http://mitpress.mit.edu/book-home.tcl?isbn=026206202X`.

[10] Clark Glymour and Gregory F. Cooper, editors. *Computation, causation, & discovery*. AAAI Press/MIT Press, 1999.

[11] Michael I. Jordan, editor. *Learning in graphical models*. MIT Press, 1999. Originally published in 1998 by Kluwer Academic Publishers.

[12] Pat Langley. *Elements of Machine Learning*. Morgan Kaufmann, 1996.

[13] Tom M. Mitchell. *Machine Learning*. McGraw-Hill, 1997. `http://www.cs.cmu.edu/~tom/mlbook.html`.

[14] Judea Pearl. *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann, 1988.

[15] C.E. Rasmussen, R.M. Neal, G.E. Hinton, D. van Camp, M. Revow, Z. Ghahramani, R. Kustra, and R. Tibshirani. The DELVE manual. Technical report, University of Toronto, 1996. `http://www.cs.utoronto.ca/~delve/`.

[16] Bernhard Schölkopf, Christopher J.C. Burges, and Alexander J. Smola, editors. *Advances in kernel methods: Support Vector learning*. MIT Press, 1999.

[17] Vladimir Vapnik. *Three remarks on the support vector method of function estimation*, chapter 3, pages 25–41. In Schölkopf et al. [16], 1999.

[18] Vladimir N. Vapnik. *Statistical learning theory*. John Wiley & Sons, 1998.

[19] Chris Watkins. Kernels from matching operations. Technical Report CSD-TR-98-07, Royal Holloway, University of London, July 1999.

[20] Sholom M. Weiss and Nitin Indurkhya. *Predictive data mining: a practical guide*. Morgan Kaufmann, 1998. `http://www.data-miner.com/`.

[21] Ian H. Witten and Eibe Frank. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann, 2000. `http://www.cs.waikato.ac.nz/ml/weka/`.