Department of Computer Science
Series of Publications A
Report A-1997-1

Plausible Prediction by Bayesian Inference

Henry Tirri

University of Helsinki
Finland

Department of Computer Science
Series of Publications A
Report A-1997-1

Plausible Prediction by Bayesian Inference

Henry Tirri

To be presented, with the permission of the Faculty of Science of the University of Helsinki, for public criticism in Auditorium III, Porthania, on June 3rd, 1997, at 12 o'clock.

University of Helsinki
Finland

Contact information

Postal address:

Department of Computer Science
P.O.Box 26 (Teollisuuskatu 23)
FIN-00014 University of Helsinki
Finland

Email address: postmaster@cs.Helsinki.FI (Internet)

URL: http://www.cs.Helsinki.FI/

Telephone: +358 9 708 51

Telefax: +358 9 708 44441

ISSN 1238-8645
ISBN 951-45-7746-9
Computing Reviews (1991) Classification: G.3, I.2.3, I.2.6
Helsinki 1997
Helsinki University Press

Plausible Prediction by Bayesian Inference

Henry Tirri

Department of Computer Science
P.O. Box 26, FIN-00014 University of Helsinki, Finland
Henry.Tirri@cs.helsinki.fi, http://www.cs.helsinki.fi/~tirri/

PhD Thesis, Series of Publications A, Report A-1997-1
Helsinki, June 1997, 122+36 pages
ISSN 1238-8645, ISBN 951-45-7746-9

Abstract

The capability to perform inference with uncertain and incomplete information is characteristic to intelligent systems. Many of the research issues in artificial intelligence and computational intelligence can actually be viewed as topics in the "science of uncertainty," which addresses the problem of plausible inference, i.e., optimal processing of incomplete information. The various different approaches to model and implement intelligent behavior such as neural networks, fuzzy logic, non-monotonic (default) logics and Bayesian networks all address the same problem of finding an appropriate language and inference mechanism to perform plausible inference, needed to implement such activities as prediction, decision making, and planning.

In this work we study the problem of plausible prediction, i.e., the problem of building predictive models from data in the presence of uncertainty. Our approach to this problem is based on the language of Bayesian probability theory both in its traditional and information theoretic form. We study Bayesian prediction theoretically and empirically with finite mixture models. Such models are interesting due to their ability to accurately model complex distributions with few parameters. In addition, finite mixture models can be viewed as a probabilistic formulation of many model families commonly used in machine learning and computational intelligence.

We first address the question of how an intelligent system should predict given the available information. We present three alternatives for probabilistic prediction: single model based prediction, evidence based prediction, and minimum encoding based prediction. We then compare the empirical performance of these alternatives by using a class of finite mixture models. The empirical results demonstrate that, especially for small data sets, both the evidence and the minimum encoding approaches outperform the traditionally used single model approach.

We then focus on the problem of constructing finite mixture models from the given data and a priori information. We give the Bayesian solution for inferring both the most probable finite mixture model structure, i.e., the proper number of mixture components, and the most probable model within the class. For general mixture models the exact solution in both problems is computationally infeasible. Thus we also evaluate the quality of approximate approaches.

The Bayesian predictive approach presented can be applied to a wide class of prediction problems appearing in various application domains, e.g., medical and fault diagnostic problems, design problems and sales support systems. Using publicly available data sets, we demonstrate empirically that Bayesian prediction with finite mixtures is highly competitive when compared to the results achieved with other popular non-Bayesian approaches using, for example, neural network and decision tree models. The Bayesian prediction method presented constitutes the kernel of the `D-SIDE/C-SIDE` software currently used in industrial applications.

Computing Reviews (1991) Categories and Subject Descriptors:

G.3. [Probability and Statistics]: Probabilistic algorithms
I.2.3 Deduction and Theorem Proving [Artificial Intelligence]: Uncertainty, "fuzzy," and probabilistic reasoning
I.2.6 Learning [Artificial Intelligence]: Concept learning, Induction

General Terms:
Theory, Algorithms

Additional Key Words and Phrases:
Bayesian inference, prediction, classification, intelligent systems

Acknowledgements

I am most grateful to my advisor, Professor Esko Ukkonen, for guiding me through the doctoral studies. The work at hand has also directly benefitted from the insightful and detailed comments of Jyrki Kivinen and the discussions with Professor Pekka Orponen.

I am indebted to all those colleagues I have been fortunate enough to collaborate with during the years. In particular I owe a great deal to those people that have contributed to the research that is reported in this dissertation. I want to extend this special gratitude to Petri Kontkanen, Jussi Lahtinen, Petri Myllymäki and Tomi Silander from the Complex Systems Computation Group at the Department of Computer Science of the University of Helsinki, and to Peter Grünwald from CWI. I feel privileged to have been able to work with such a stimulating and active research group.

From the long list of other colleagues and friends that have made these years of study enjoyable and intellectually challenging, I would like to especially mention Timo Alanko, Professor Ralph Back, true Bayesian Peter Cheeseman, Hannu Erkiö, Professor Witold Litwin, Patrik Floréen, Professor David Madigan, Professor Heikki Mannila, Robert Morris, Professor Erkki Oja, Professor Kari-Jouko Räihä, and Professor Jari Veijalainen. I am also grateful to the Head of Department, Professor Martti Tienari, and to all the personnel at the department for the inspiring and lively working environment. In particular I want to express my gratitude to Petri Kutvonen and other system support people without whom computer science at our department would only be science.

My deepest gratitude, however, goes to my four ladies: my dear wife Kirsi, and my daughters Karoliina, Katariina and Vilhelmiina, to whom I dedicate this dissertation as a minuscule compensation for the hours of attention they should have deserved, but did not get.

The financial support of the Technology Development Center (TEKES), the Academy of Finland and the Helsinki Graduate School of Computer Science and Engineering is gratefully acknowledged.

Contents

Chapter 1

Introduction

"I am HAL Nine Thousand computer Production Number 3. I became operational at the HAL Plant in Urbana, Illinois, on January 12, 1997."

—Arthur C. Clarke and Stanley Kubrick in 2001: A Space Odyssey

Throughout the history of the science of computing, building intelligent systems has been one of the fundamental objectives, this desire being so great that it has given birth to such multidisciplinary research fields as *artificial intelligence* [121] and *computational intelligence* [10]. In addition to appealing to computer scientists, these areas have attracted researchers from many areas including linguistics, mathematics, physics, neurosciences, psychology, cognitive science and philosophy. Unfortunately, the informal objective of building an intelligent artifact seems to be the only issue that is commonly agreed upon. The notion of "intelligence," and even the possibility of artificial entities exhibiting intelligent behavior, has been a subject of heated debate among computer scientists and philosophers since the beginning of the computing era (see the discussions in [27, 41, 42, 62, 125, 126], e.g.). One of the early landmarks of this debate, the Turing Test [141], is already approaching its 50th anniversary.

On machine intelligence and plausible inference. For the convenience of presenting the ideas, let us focus our interest on a hypothetical computing system which we will call `HAL`. The purpose of introducing `HAL` is not only notational; it allows us to avoid some of the verbal confusions of philosophical discussions and perhaps to make some of the points clearer. We will start by inspecting two recent definitions for `HAL` to exhibit intelligence—one from computational intelligence and the other from artificial intelligence. According to Bezdek [10], `HAL` exhibits computational intelligence when it

> "...deals only with numerical (low-level) data, has a pattern recognition component, and does not use knowledge in the AI sense; and additionally when it (begins to) exhibit (i) computational adaptivity; (ii) computational fault tolerance; (iii) speed approaching humanlike turnaround, and (iv) error rates that approximate human performance."

The other definition comes from the recent unified framework for artificial intelligence, which adopts the view that intelligence is concerned mainly with rational action. In this approach `HAL` is considered intelligent if it takes the best possible action in a situation, i.e., it approximates well the behavior of an ideal rational agent defined by Russel and Norvig as follows [121]:

> "For each possible percept sequence, an *ideal rational agent* should do whatever action is expected to maximize its performance measure, on the basis of the evidence provided by the percept sequence and whatever built-in knowledge the agent has."

These definitions reflect the two prevailing philosophical approaches to machine intelligence: the descriptive approach and the normative approach. The descriptive approach is interested in modeling the intelligent behavior of biological organisms and then reproducing this behavior in computers. The normative approach aims at a prescription for "correct" behavior whose value is judged purely on performance grounds. Neither of these approaches is central to the work at hand, however; if forced to take sides, we tend to adopt the normative approach. In spite of their apparent differences, both of the definitions implicitly imply the existence of a common component of machine intelligence: `HAL` should be capable of making use of the available information to perform inferences, which help it to work towards a goal of maximizing some given performance measure. Vague as it is, let us for a moment study the implications of this observation.

In an idealized situation, `HAL` would have complete information relevant to its performance maximization task and should perform logical inference (with predicate calculus). An example of this type of a task is automated theorem proving, where `HAL` uses its information about the axioms and syllogisms to infer theorems of interest. Even in this complete information case there are theoretical and practical limits to `HAL`'s capabilities, as demonstrated by the well-known results of Gödel [57] and Turing [140].

Now let us assume that we want `HAL` to play "Five- Card Stud Poker," and we define the performance measure to be the amount of money won from its opponents (whether or not playing poker exhibits intelligent behavior is of course debatable). `HAL` definitely does not have all the necessary information available to perform logical inference. On one hand, it meets with a great amount of uncertainty: it is uncertain which card will be dealt in the next round, it is uncertain about the values of the cards dealt face down, it is uncertain of the betting strategy of its opponents, it is uncertain whether a meteorite will hit the clubhouse and end the game abruptly before the next round (in which case maximizing the performance would require `HAL` to quit immediately), etc.

On the other hand, it has information available in the cards dealt face up, general information about the composition of the card deck, information about the betting behavior of the opponents from the earlier games, information about celestial mechanics and statistics of a meteorite hitting a particular place, etc. Since in this context not doing anything is also an act, `HAL` has to perform inference regardless of the fact that all the necessary information to do logical inference is not available. This type of inference, which `HAL` does with incomplete information, is called *plausible inference* [109]. The observation that real problems encountered in practice almost invariably require plausible inference, serves as a starting point of this work.

From this perspective, many of the research issues in artificial and computational intelligence can actually be viewed as topics in the "science of uncertainty,"[1] which addresses the problem of optimal processing of incomplete information. The various different approaches to model and implement intelligent behavior such as neural networks [63, 80], fuzzy logic [92, 148, 149], non-monotonic (default) logics [97] and Bayesian networks [64, 107] all address the same fundamental question: what is the appropriate language and inference mechanism for `HAL` to perform plausible inference, which is needed to implement such activities as prediction, decision making, and planning? Naturally, the answer varies depending on the approach adopted. Many of these answers are quite involved and intertwine procedural components, e.g., "learning algorithms", and representational mechanisms such as graphical models.

In this work our purpose is to gradually introduce into the reader's mind a perhaps less familiar, Bayesian approach to plausible inference. This Bayesian language of inference is based on Bayesian probability theory [9, 8] which, instead of a long-run frequency, views probability as a degree of belief in an event. Rational, i.e., well-performing, decisions typically rely on good predictions of unknown information. Therefore we will be interested in a specific capability of `HAL`'s: *prediction* of unknown quantities based on the information available to it. During this process we end up developing the technical contributions of this work: computational machinery offering a

[1]Term introduced by Peter Cheeseman.

viable solution to a wide class of prediction problems appearing in various application domains. When this process is finished, however, we will hope that we have also provided the reader a demonstration of the usability of having a firm theoretical basis for handling inference under uncertainty as well as and given a flavor of what the Bayesian inference approach can offer to a practitioner in machine learning and computational intelligence.

HAL's specification and Bayesian inference. Let us now return to the design of HAL and give some intuitively sensible specifications for HAL's representation of uncertainty.

First of all, HAL should be able to express its "view of the world" as well-defined propositions (possibly quantified) together with associated plausibility values, which describe HAL's belief that the corresponding proposition is true. Furthermore, its plausibility values should be representable as real numbers in HAL's memory. We would like HAL's beliefs to be context dependent: the plausibility of a proposition can depend explicitly on plausibilities of other propositions. In addition, HAL should also be capable of hypothetical inference, i.e., it should be able to assign plausibilities to conjunctions of propositions conditioned on the truth of other propositions. In its beliefs HAL should show consistency—it should have equal plausibilities for propositions that are believed to have the same logical truth value. Finally, HAL should have a notion of complementarity; its plausibility in the negation of a proposition should be a monotonically decreasing function of the plausibility of the proposition itself.

Interestingly, after this rather informal specification, HAL can be given the rules telling how it should calculate the plausibilities of new compound propositions from the plausibilities of the original propositions, i.e., the "grammar" of its plausible inference. In Chapter 2 we will see that these rules are the axioms of probability theory, now as a formalization of the notion of "belief" rather than a frequency. Therefore, in this work we will equate plausibility ("belief in a proposition") with the notion of conditional probability, and plausible inference with Bayesian inference [9, 70, 71][2]. The rewriting of one of the axioms in a form known as the Bayes' theorem then gives HAL an "update rule," which it can apply when new information is available: the new probability of a proposition can be calculated by combining the current probability of a proposition and the probability of new evidence given the proposition. This, of course, raises the question of the

[2] It should be observed that with the specification given above, HAL's inference cannot be based on some alternative approaches such as fuzzy logic [149] or the Dempster-Shafer theory [127] since they violate one or more of the requirements [68].

initial state of this process, i.e., how does HAL assign the initial probabilities so that this update process can be started? In the Bayesian terminology these initial probabilities are called priors. Chapter 2 concludes with a brief discussion on principles for assigning both non-informative and informative priors.

HAL's inferences are based on conditional probabilities. Therefore, in order to perform inferences HAL has to be able to determine what is the relevant information to condition on its probabilistic conclusions—in other words, HAL needs *models*. Without any loss of generality we can assume that HAL can model anything of interest with probabilistic models. Thus HAL's models describe what the relevant information is and what independence assumptions it makes. For prediction purposes, it is useful to assume that HAL's models come from a specified set of models. We will call such a set a model family; it will describe the general structure (e.g., the number of parameters and their types) of HAL's models. Naturally, HAL can use many different model families. Bayesian probabilistic modeling has the interesting property that it can also be cast in an information theoretic form, where the models can be viewed as codes, in which case HAL is interested in short encodings of the model and the (observed) data together [116, 145]. All these modeling issues are discussed in Chapter 3.

The main emphasis of Chapter 3, however, is on how HAL implements prediction as *deductive Bayesian inference*, i.e., how HAL uses the model family and the available data to compute the probability of an unknown quantity. We will describe three alternative methods: a single model based prediction, evidence (model averaging) based prediction, and prediction based on the minimum encoding approach by Rissanen [117].

To avoid confusion, one final comment on Bayesian modeling is in order. All inference HAL performs is conditioned on a model or a set of models. The fact that HAL uses a specific probabilistic model or a model family for its inference does not mean that HAL (or we) believe that probabilistic models are somehow an inherent property of the reality from which the data observed by HAL are generated. A model family is only a language which HAL can use to express constraints for the data of interest. For any given model family and the results HAL exhibits, there is no way of knowing that there is not another model family which HAL could use, and still show similar performance in its predictions. In fact, in most cases such an alternative model family exists. Reality, of course, is neither way.

Up to this point we have discussed very generic aspects of HAL's design relative to a non-identified model family. In Chapter 4 we fix HAL's model family to be the set of finite mixture models [138], with the restriction that the observed data are discrete. The choice of this particular model family is not an arbitrary one—using finite mixture models corresponds to providing HAL with a set of probabilistic concepts [47] from which the attributes (or features) of data are the part HAL is able to observe. In addition, many model families commonly used in machine learning and computational intelligence can be given a probabilistic interpretation as mixture models. Chapter 4 discusses how HAL's three predictive methods can be implemented in the case of finite mixtures. The chapter is concluded by giving a comparison of the empirical prediction performance of all the methods for a particular subclass of finite mixtures called the Naive Bayes family, where all the methods can be implemented without the approximations needed for the general mixture models. As far as we know, the comparison performed is the first of its kind—in fact, we are not even aware of any other application of Rissanen's new encoding approach.

Chapters 3 and 4 discuss the use of Bayesian inference for deductive purposes, i.e., how HAL's predictions should be performed given a model (or a set of models). Still, HAL has to somehow establish these models. There is nothing in the Bayesian inference formalism that restricts its applicability to probabilistic deduction. It is also possible to do *Bayesian induction* and let HAL use the update rule to infer probabilistic models from its data. Consequently, Chapter 5 applies the general Bayesian modeling framework of Chapter 3 for the construction of finite mixture models from the data. This construction is essentially a search process for high probability models from the model family. As opposed to the more restricted Naive Bayes model family discussed in Chapter 4, we will demonstrate that for the general mixtures this search can only find approximations of the most probable models.

After the theoretical introduction to HAL's Bayesian predictive framework with finite mixture models, one can question how the developed machinery performs in practice. HAL's design was motivated by the need to give it the capability to predict well. In many practical applications, HAL has to resort to building predictive models using only domain data with very little or no prior information. The advantage of discussing predictive modeling is that the methods can be validated, at least to a degree, by inspecting HAL's actual predictive performance, either with benchmark data sets or with real applications. The Bayesian predictive framework described in this work has been tested in both respects. It constitutes the kernel of the D-SIDE/C-SIDE software currently applied for industrial applications. In Chapter 6 we report empirical results with publicly available common classification benchmarks and evaluate HAL's Bayesian predictive framework against the results achieved with other popular model families such as neural networks and decision trees using non-Bayesian approaches.

From the discussion above, it should not be concluded that Bayesian inference is simple and understood in all its details. On the contrary, the Bayesian approach is open-ended, and new Bayesian analysis methods are constantly being developed, as witnessed by the various conference series on topics such as "Bayesian Statistics" and "Maximum Entropy and Bayesian Methods". Bayesian inference applied to real problems requires instantiation of the general principles in the case of a particular model family, which can be a complex (and sometimes even controversial) task involving computational issues (see, e.g., [56]). In Chapter 7 we discuss how `HAL`'s design could be improved with several extensions of the work described here.

Contributions and research history. For building predictive models, numerous alternatives to the approach discussed in this work are available, including non-probabilistic and probabilistic neural networks, decision trees, rule-based systems, statistical discrimination techniques, and general Bayesian network modeling. Recently other multiple-model approaches such as boosting [50] and bagging [16] have also been introduced. Although the Bayesian prediction described in this work provides a consistent and theoretically justified formal framework for prediction, one can ask what our work has to offer to the practitioner developing predictive models. There are several answers to this question:

- *Principled approach to avoid overfitting of data when the model is constructed.* Bayesian methods with any model family have an "in-built" mechanism for the appropriate tradeoff between the model complexity and fit to the data. Thus the *ad hoc* approaches to selection of the model structure, or extra regularization and penalization terms used in many machine learning and neural network approaches, are not necessary.

- *Time-efficient prediction.* Although general Bayesian networks use conditional independence to simplify Bayesian prediction, even approximate prediction using an arbitrary Bayesian network for discrete variables is NP-hard [34]. By restricting our approach to finite mixture models (corresponding to a one-level Bayesian tree), exact Bayesian prediction can be performed efficiently.

- *Controlled model construction time.* For the general mixture models learning time, i.e., the time to search for the best model, is relative to the number of local maximum models explored by stochastic search.

Thus the more time one has, the better the resulting models are. For fast results, one can limit the number of searches and use the best model found up to that point. If one restricts the search to models from the Naive Bayes model family, the model construction requires only one pass through the data and consequently is very efficient.

- *Good prediction performance with small data sets.* As opposed to traditional single-model based approaches such as feed-forward neural networks with backpropagation learning or decision trees, the possibility to use multiple-model prediction allows construction of models that predict well even with small data sets. This is important in many practical applications, where gathering of observational data is expensive. Both the evidence based prediction and the stochastic complexity based prediction with the Naive Bayes model family exhibit this behavior.

- *Natural handling of incomplete data.* Many traditional model construction approaches produce inaccurate models in the presence of missing data, and typically omit data if part of the values are missing. The Bayesian approach described in this work models the full joint probability distribution of the domain; thus it can handle missing values in inputs for both model construction and the prediction phases.

- *Very few "technical" parameters to be provided by the user.* Many methods to construct predictive models require the user to specify a substantial number of technical parameters that control the model search process, feed-forward backpropagation neural networks being perhaps the most notorious example. In our approach the user needs to provide very few parameters if non-informative priors are used and the prediction method is fixed. For the general mixture models only the convergence criteria for the search algorithm, the maximum number of mixture components and the number of individual searches for each mixture size are required. The Naive Bayes prediction exhibits the extreme case by requiring only the information about which one of the variables is the class variable.

Many of the ideas and results in this monograph have been published in preliminary form as joint works with various members in the Complex Systems Computation Group. This monograph aims to be more approachable than those research papers by containing some introductory material as well and by presenting the Bayesian inference especially to the computer science audience. Furthermore, in some cases the published ideas and results appear here in an improved and polished form.

The main contribution of Chapters 3 and 4, the formulation and comparison of the three alternative methods for predicting with the Naive Bayes model family, is joint work with Peter Grünwald from CWI and was published in [86]. To our knowledge, this comparison is the first of its nature, in particular concerning the formulation and application of the predictive form of Rissanen's recent version of stochastic complexity.

The empirical comparisons of the evidence approximations discussed in Chapter 5 for the complete data evidence case appeared in [88], and those for the incomplete data evidence case in [91]. The latter is unique in the respect that it evaluates the Cheeseman-Stutz approximation against cross-validation, an empirical technique commonly used for model class selection in machine learning, neural network community and statistics. In addition, both of these studies use natural data sets instead of synthetic data sets such as the ones used in [65].

Many of the empirical prediction results presented in Chapter 6 were published in [90, 136]. Some of the results have been improved since then, and we have added here the results achieved with the Naive Bayes model family. The latter paper also discusses the novel idea of viewing lazy learning from Bayesian perspective (the related discussion on Bayesian Case-Based Reasoning appeared in [135]). The details of the EM algorithm for the specific Bayesian finite mixture case for multinomial distributions were presented in [89]. The updated results reported in Chapter 6 also indicate that our `C-SIDE/D-SIDE` implementation of the Bayesian finite mixture approach outperforms any Naive Bayes classifier and general Bayesian network learning algorithm of which we are currently aware.

Chapter 2

Bayesian language for inference

"Inside every nonBayesian there is a Bayesian struggling to get out."

—Dennis V. Lindley

In any realistic application, `HAL` cannot have perfect information on which it could base its inference. Therefore, to be able to perform as well as possible, `HAL` has to be provided with a plausible inference framework, which allows it to evaluate the plausibility of unknown facts given its incomplete "view of the world." In particular it needs a procedure to change its beliefs when new observable data arrive. This chapter discusses the Bayesian alternative among such frameworks.

We start by defining a set of simple desirable properties `HAL` should exhibit when performing plausible inference. For our purposes it is sufficient to assume that `HAL` reasons about two-valued logic propositions. From the properties defined in Section 2.1 the "grammar of Bayesian inference" arises as a natural consequence (Section 2.2). Therefore the rules that tell `HAL` how to calculate the plausibilities of new compound propositions from the plausibilities of the original propositions are the product and sum rules in the Bayesian probability theory, on which all the work here is based.

The grammar, however, is only half of of the required theory. `HAL` still faces the problem of initial plausibility assignments, i.e., how to assign numerical values for the propositions in the first place. This topic is briefly addressed in Section 2.3. We conclude the chapter by discussing Bayes' theorem, which gives `HAL` an update rule to adjust its plausibility assessments when the state of knowledge regarding the proposition of interest changes through the acquisition of new data (Section 2.4). This central viewpoint of Bayesian inference—that belief is fundamentally an update process—is quite natural one for computer science; however, historically it has been the main reason for the rejection of the Bayesian approach because the starting point of this process, the use of priors, is controversial.

Many different sets of desirable properties for plausible inference, akin to the ones used here for HAL, have been proposed in the literature [36, 58, 113, 122]. Perhaps the most famous of these is the widely cited work by Cox [32, 33] (for a recent refined version of the original formal derivation by Cox see [106] and for the counterexample to the original proof see [61]), the properties used here follow the ones introduced by Jaynes in [69, 70] (see also the discussion in [22, 68, 94]). Interestingly, each of these sets of properties leads to the same set of rules—the rules of Bayesian probability theory. In the presence of alternative frameworks for plausible inference, such as fuzzy logic [148] or Dempster-Shafer theory [127], in our view this insensitivity to the choice of properties provides a particularly compelling argument for using Bayesian probability as a measure for plausibility. It is also notable that a rigorous mathematical framework can be erected upon a such an apparently vague notion of a measure of degree of plausibility.

2.1 The desiderata for plausible inference

Let us now define the *qualitative properties* to be satisfied by HAL's plausible inference.

Desideratum 2.1 *Degrees of plausibility are represented by real numbers.*

This representation property is essentially motivated by the requirement that, to be able to manipulate plausibilities, HAL has to be able to store and modify them, and thus they must be associated with some physical quantity. Therefore there has to be some kind of association between degrees of plausibility and real numbers. More theoretical justifications for using real numbers as plausibilities can be found in [70, 107]. We take it as a convention that propositions with the same truth value must have equal plausibility, and that a greater plausibility will correspond to a greater number. In addition we assume also a continuity property, a very small increase in plausibility ought to correspond only to a slightly greater number.

In general HAL assigns the plausibility to some proposition H given the truth of proposition D. Following the common notation we indicate this by the symbol $H|D$ which can be called the *(conditional) plausibility* that H is true, given that D is true[1]. It should be noticed that plausibilities are always conditional, i.e., they are always relative to HAL's state of knowledge. We frequently need several conditioning propositions, thus logical conjunction is

[1] If $H|D$ appears in the running text, for clarity of expression we often add parentheses.

denoted by HD and logical disjunction by $H + D$. In addition, the logical negation of H is denoted by $\bar{H}$. Finally $(H|DI)$ is not defined when D and I are mutually contradictory.

Desideratum 2.2 *Direction of inference has a qualitative correspondence with common sense.*

This second desideratum is related to `HAL` being able to perform inference that does not contradict common sense. It should be noticed that for us this does not mean an attempt to model human common sense, we just want to include a property which we argue "rational" inference should have—after all, our approach to `HAL`'s design is normative. Accordingly if `HAL` has old information I which gets updated to I' in such a way that the plausibility of H is increased, i.e.,

$$(H|I') > (H|I),$$

and the plausibility of D given H is not changed, i.e.,

$$(D|HI') = (D|HI),$$

this can produce only an increase in the plausibility that both H and D are true. In other words for such a situation

$$(HD|I') \geq (HD|I),$$

and the plausibility that H is false has to be decreased

$$(\bar{H}|I') < (\bar{H}|I).$$

This qualitative property of inference simply gives `HAL` a sense of direction in its inference. It should be observed that the property says nothing about how much the plausibilities change, except that our continuity assumption now requires that if the change in $(H|D)$ is small, it can induce only a small change in $(HD|I)$ and $(\bar{H}|I)$.

The last set of desiderata is related to consistency of inference. This consistency requirement can be described with three different properties.

Desideratum 2.3 *(Internal consistency.) If a conclusion can be inferred in more than one way, every possible way must lead to the same result.*

Desideratum 2.4 *(Propriety.)* `HAL` *always takes into account all of the information that is relevant to a question.*

Desideratum 2.5 *(Jaynes consistency.) Equivalent states of knowledge must be represented by equivalent plausibility assignments.*

Of these desiderata the last two properties can be seen as input/output requirements for HAL 's inference. The "Propriety" desideratum requires that in plausible inference all the relevant evidence has to be used, in particular that all the relevant input data has to be taken into account. From the "Jaynes consistency" property (introduced by Jaynes in [69]) it follows that if in two problems HAL 's state of knowledge is the same (excluding proposition labeling), then the same plausibilities must be assigned in both cases.

Perhaps somewhat surprisingly, the set of properties listed above, together with some technical properties of monotonic functions, uniquely determines how HAL must perform inference, i.e., there is only one set of mathematical operations for manipulating plausibilities which has all these properties.

2.2 The probability axioms: the grammar of Bayesian inference

Having now formulated the requirements for plausible inference it is a matter of straightforward mathematics to work out the consequences of our desiderata. Given two or more propositions, other more complicated propositions can be built by considering them together. Thus we need rules to tell us how the plausibilities of the new compound propositions can be calculated from the the plausibilities of the original propositions, i.e., we would like to find out the "grammar" for our theory. Here we assume that the original plausibilities are given. How such initial assignments are done is the topic of Section 2.3.

In order to be useful, the plausibility calculus has to be powerful enough to enable HAL to calculate the plausibility of any proposition built from other propositions using Boolean algebra. Since it is well known that only a subset of the common logical operations is needed to generate all possible propositions, we can restrict our discussion on only two of them: conjunction and negation. The properties listed in Section 2.1 are sufficient to specify the rules for calculating the plausibility of a negated proposition and of the conjunction of two propositions leading to *the sum rule* and *the product rule*, correspondingly. Full detailed derivation of these rules is quite involved and lengthy and thus omitted, as our interest here is to focus on developing Bayesian inference for a particular domain, i.e., for the finite mixture model family. The argumentation in the proofs, however, is interesting for the intuitive justification of Bayesian inference, and since the details of the original proofs by Cox [32, 33], Aczél [1] and Jaynes [69, 70] are not widely known, we will outline the intuitive idea in the derivation of the product rule, and present results of the analogous derivation of the sum rule. For further details the references mentioned in the beginning of this chapter may be consulted.

2.2.1 The product rule

We first seek a consistent rule relating the plausibility of the logical product (HD) to the plausibilities of H and D separately. In particular, let us find $(HD|I)$. The separate plausibilities of H and D that may be known to us include the four quantities

$$u = (H|I),\ x = (D|I),\ y = (H|DI),\ v = (D|HI).$$

By the "Propriety" Desideratum, we should use all of these assuming they are relevant.

Now Desideratum 2.2 can be used to determine if only a subset of these four quantities is actually relevant. For example, common sense indicates that $(HD|I)$ cannot depend on only one of x, y, u, or v. This leaves eleven combinations of two or more of these plausibilities. A little deeper thought reveals that most of these combinations violate common sense. For example, if $(HD|I)$ depended only on u and x we would have no way of expressing the possibility that H and D are exclusive. Carrying out this somewhat tedious analysis, Tribus [139] shows that all but two of the possibilities can exhibit qualitative violations of common sense in some extreme case. The only possible relevant combinations are x and y, or u and v. Intuitively this follows from the fact that there are two ways a decision about the truth of (HD) can be broken down into decisions about H and D. We can either decide that H is true, and then, accepting the truth of H decide that D is true. Or, vice versa, we first decide that D is true and then make our decision about H given the truth of D. By the commutativity of logical conjunction, we can exchange H and D in all the quantities. Consequently, since the different pairs x, y and u, v merely reflect the ordering of H and D, we may focus on one pair.

Let us now denote $z = (HD|I)$. Thus we seek for a function F such that

$$z = F(x, y). \tag{2.1}$$

At this point the "Internal consistency" (Desideratum 2.3) can be used by setting up a problem that can be solved two different ways, and by requiring the solutions to be identical. Suppose we try to find the plausibility that three propositions H, D, I would be true simultaneously. The joint proposition

(HDI) can be built in two different ways: $(HDI) = H(DI) = (HD)I$. From the former of these equations and (2.1) we know that

$$(HDI|J) = F((DI|J), (H|DIJ)),$$

where (DI) is treated as a single proposition. Similarly from the second equality we get

$$(HDI|J) = F((I|J), (HD|IJ)).$$

Repeated application of (2.1) and the "Internal consistency" now requires that the function F obeys the equation

$$F(F(x, y), z) = F(x, F(y, z)),$$

for all real values of x, y, and z, i.e., "the associativity equation" of Aczél [1] who derives the general solution in Equation (2.2). The better known shorter proof presented by Cox [33] assumes differentiability. The general solution to this functional equation is

$$F(x, y) = w^{-1}(w(x)w(y)), \tag{2.2}$$

where $w(x)$ is any positive, continuous, monotonic function of plausibility. Thus we have not uniquely specified F, but constrained its form. Using this solution with (2.1), the consistency requirement tells us that

$$w(HD|I) = w(H|DI)w(D|I) \tag{2.3}$$

Equation (2.3) is called henceforth the *product rule*. The result has been derived as a necessary condition for the Internal consistency property. Conversely, it is evident that (2.3) is also sufficient to ensure this consistency for any number of joint propositions.

The requirements of qualitative correspondence with common sense impose further conditions on the function $w(x)$. For example, suppose that in Equation (2.3) H is certain, given I. Then in the context produced by knowledge of I, the propositions (HD) and D are the same, in the sense that one is true if and only if the other is true. By the most primitive property of all, propositions with the same truth value must have equal plausibility, hence

$$(HD|I) = (D|I)$$

and also we will have

$$(H|DI) = (H|I)$$

because if H is already certain given I, then given any other information D which does not contradict I, it is still certain. In this case, (2.3) reduces to

$$w(D|I) = w(H|I)w(D|I)$$

and this must hold no matter how plausible or implausible D is to HAL. Thus the function $w(x)$ must have the property that certainty is represented by $w(H|I) = 1$.

Similarly we can suppose that H is impossible, given I. It follows that the proposition (HD) is also impossible given I:

$$(HD|I) = (H|I)$$

and if H is already impossible given I, then given any further information D which does not contradict I, H would still be impossible:

$$(H|DI) = (H|I).$$

In this case, (2.3) reduces to

$$w(H|I) = w(H|I)w(D|I) \tag{2.4}$$

and again this equation must hold no matter what plausibility D might have. There are only two possible values of $w(H|I)$ that could satisfy this condition; it could be 0 or $+\infty$ (the choice $-\infty$ is ruled out because then by continuity $w(D|I)$ would have to be capable of negative values and (2.4) would then be a contradiction).

In summary, qualitative correspondence with common sense requires that $w(x)$ be a positive continuous monotonic function. It may be either increasing or decreasing. If increasing, the function must range from zero for impossibility up to one for certainty, if decreasing, it must range from ∞ for impossibility down to one for certainty. Thus far, our conditions say nothing at all about how it varies between these limits. However, these two possibilities of representation are not different in content. Given any function $w_1(x)$ which is acceptable by the above criteria and represents impossibility by ∞, we can define a new function $w_2(x) \equiv 1/w_1(x)$, which will be equally acceptable and represents impossibility by zero. Therefore, there will be no loss of generality in adopting the familiar choice $0 \leq w(x) \leq 1$, but one should observe that it is *chosen here as a convention.*

2.2.2 The sum rule

The propositions being considered are of the Aristotelian logical type which must be either true or false, hence we know that the logical product $(H\bar{H})$

is always false, and the logical sum $(H + \bar{H})$ always true. The plausibility that H is false must depend in some way on the plausibility that it is true. In other words, if we define $r \equiv w(H|D)$ and $s \equiv w(\bar{H}|D)$, there must exist some functional relation $s = S(r)$. Similar to the derivation in the previous section Desiderata 2.2 and 2.3 lead to a functional equation whose solution implies that for some positive m (see [69, 70])

$$w^m(H|D) + w^m(\bar{H}|D) = 1. \tag{2.5}$$

From the discussion above we know that associativity of the logical product requires that some monotonic function $w(x)$ of the plausibility $x = (H|D)$ must obey the product rule 2.3. Now this same function must also obey the *sum rule* in (2.5). We can write the product rule equally well as

$$w^m(HD|I) = w^m(H|I)w^m(D|HI) = w^m(D|I)w^m(H|DI)$$

which shows that the value of m is actually irrelevant; for whatever value is chosen, we are free to make a simple change of variables from $w(x)$ to the different monotonic function $p(x) \equiv w^m(x)$, so that we may always write

$$p(HD|I) = p(H|I)p(D|HI) = p(D|I)p(H|DI) \tag{2.6}$$

and

$$p(H|D) + p(\bar{H}|D) = 1. \tag{2.7}$$

It should be observed that this entails no loss of generality, for the only requirement imposed on the function $w(x)$ is that it is a continuous monotonic increasing function ranging from $w = 0$ for impossibility to $w = 1$ for certainty. But if $w(x)$ satisfies this, then so does $w^m(x)$, $0 < m < \infty$. Consequently allowing different values of m does not give us any freedom that did not exist already in the arbitrariness of $w(x)$. All possibilities allowed by our desiderata are contained in (2.6) and (2.7) in which $p(x)$ is any continuous monotonic increasing function with the range $0 \leq p(x) \leq 1$.

2.2.3 Relation to logical inference

We have seen that the equations (2.6) and (2.7) follow as natural consequences from the properties assumed for `HAL`'s plausible inference. Just as conjunction and negation are an adequate set for logical inference, `HAL` is able to derive all legitimate relationships between plausibilities from these product and sum rules.

Plausible inference allows `HAL` to reason in the presence of uncertainty. In the extreme case where all the necessary information is available, however, HAL's inference should not contradict logical inference. Let us now briefly examine how the plausible inference based on equations (2.6) and (2.7) relates to logical inference.

From (2.7) it is obvious that in the limit as $p(H|D)$ approaches 0 or $p(H|D)$ approaches 1, if H is true, then $\bar{H}$ must be false. Logical inference is based on the two syllogisms

$$\frac{\begin{array}{c} H \Rightarrow D \\ H \text{ true} \end{array}}{D \text{ true}} \qquad \frac{\begin{array}{c} H \Rightarrow D \\ D \text{ false} \end{array}}{H \text{ false}} \tag{2.8}$$

and their consequences. From (2.6) we get

$$p(D|HI) = \frac{p(HD|I)}{p(H|I)} \qquad p(H|\bar{D}I) = \frac{p(H\bar{D}|I)}{p(\bar{D}|I)} \tag{2.9}$$

Now if we define $I \equiv (H \Rightarrow D)$, from (2.8) it follows that $p(HD|I) = p(H|I)$ and $p(H\bar{D}|I) = 0$ and thus (2.9) becomes

$$p(D|HI) = 1 \qquad p(H|\bar{D}I) = 0 \tag{2.10}$$

as stated in the original syllogisms. Therefore HAL will perform logical inference "in the limit", i.e., when it becomes more and more certain on its conclusions.

After this exercise the connection between plausible inference and the Bayesian probability theory becomes evident. Equations (2.6) and (2.7) are the familiar "axioms" of probability theory, and thus we can identify the quantity $p(H|D)$ as the probability (i.e., belief) of H given D. That is, *probability in this study is taken to be a technical term referring to a monotonic function of plausibility obeying the equations (2.6) and (2.7).* One should not be deceived by the intuitive simplicity of this result—it shows that assuming the desiderata presented in Section 2.1 together with some technicalities required during the derivation *every allowed extension of Aristotelian logic to plausibility theory is isomorphic to Bayesian probability theory*[2]. The different choices of the monotonic function $p(x)$ correspond

[2] One should observe that one of the underlying assumptions is that we extend two-valued logic to plausible inference. A proposition H can only be true or false, but the Bayesian probability theory tells us how *plausible* the truth of H is. Therefore our discussion does not directly address approaches to plausible inference which violate the "law of excluded middle", in particular it does not address the fuzzy approaches [92]. In fact, some of the fuzzy approaches violate already our desiderata by assuming independence in the product rule [21, 68].

only to different ways HAL stores plausibility values internally; HAL's externally observable behavior would be just the same. In fact, since $p(x)$ is a monotonic function of x, the plausibility x is a monotonic function of p defined in $0 \leq p(x) \leq 1$. Of all the possible functions $x(p)$ one can then choose the one specified by $x(p) = p$, since this leads to the simplest rules of combination, equations (2.6) and (2.7).

Hereinafter we will move from talking about plausibilities to using probabilities, and state the product rule and sum rule as axioms. In the following let H, D, I be propositions.

Axiom 2.1 (Product rule) *The probability of the logical product* $(HD|I)$ *is the (arithmetic) product of the probabilities* $(H|I)$ *and* $(D|HI)$*, i.e.,*

$$p(HD|I) = p(H|I)p(D|HI).$$

Axiom 2.2 (Sum rule) *The sum of the probability of a proposition* $(H|I)$ *and the probability of its logical negation* $(\bar{H}|I)$ *is 1, i.e.,*

$$p(H|I) + p(\bar{H}|I) = 1.$$

As noted already earlier, this isomorphism of allowed plausibility theories to probability theory can be derived for similar sets of properties other than the one used here for HAL. It should be observed, however, that alternative choices for implementing HAL's plausible inference exist, the most notable ones being the various fuzzy logic approaches [92, 149], and the Dempster-Shafer theory [127]. There is an ongoing debate about the advantages and disadvantages of these approaches with respect to Bayesian inference, but reviewing this discussion is beyond the scope of our work (for more details an interested reader is referred to the recent monograph by Paris [106] and the articles by Cheeseman [21, 22]).

We are interested in building intelligent systems capable of inference with incomplete information. From this perspective, the logical approach for the Bayesian inference presented is the most natural one. We should point out, however, that the foundations of Bayesian probability theory can also be based on other than logical aspects, such as the operational considerations presented in [9].

2.3 Assigning probabilities: the vocabulary of inference

We have discussed the rules by which HAL can manipulate plausibilities, and observed that the equations derived are in fact the axioms of Bayesian probability theory. We have left open, however, the problem how `HAL` assigns actual numerical values to propositions. The situation is analogous to logical inference, where the truth of certain propositions can be inferred given the truth or falseness of other assumed propositions, which are provided as input to the theory. Similarly, for plausible inference we need to assume that some "input" probabilities are assigned directly, as opposed to being derived from other probabilities using Axioms 2.1 and 2.2. Thus we seek rules for converting information about propositions into numerical assignments of probabilities. *In Bayesian probability theory all probabilities are conditioned by propositions that indicate exactly what was assumed in the assignment of a probability*, i.e., the probabilities are not properties of the propositions themselves. In this sense they are "subjective" and describe `HAL`'s state of knowledge, not states of nature. But they are "objective" in that all conditioning information has to be taken into account (Desideratum 2.4) and that equivalent states of knowledge are represented by equal probabilities (Desideratum 2.5).

Finding general rules for converting information D into a probability assignment $p(H|D)$ is an important part of the Bayesian probability theory and present in the choice of prior distributions (see discussion in Section 2.4). Development of such rules for different types of information is still a topic of ongoing research. Here we will only briefly outline some of the most elementary approaches (for more detailed discussion see e.g., [8, 55, 70]).

2.3.1 Non-informative probabilities

The simplest type of information we can have about some proposition H_1 is a specification of the alternatives $\{H_2, H_3, \ldots\}$ to it. Probability assignments that use only this information are referred to as *non-informative probabilities.*

For example assume Problem I where we have two propositions H_1 and H_2 and we only know that these propositions form an exhaustive set of exclusive alternatives. For notational reasons let us define $I \equiv H_1 + H_2$ to indicate the conditioning information. The propositions are exclusive, thus $p(H_1 H_2|I) = 0$ and Equation (2.7) implies that $p(H_2|I) = 1 - p(H_1|I)$. Let us now consider Problem II which differs from Problem I only in relabeling of the propositions in such a way that H_2 is replaced by H_1' and H_1 by H_2'. The "new" conditioning information is then

$$I' = H_1' + H_2' = H_1 + H_2 = I.$$

Evidently $p(H_1'|I) = p(H_2|I)$, and $p(H_2'|I) = p(H_1|I)$. Now, if information I is indifferent to H_1 and H_2, the state of knowledge in Problem II regarding H_1' and H_2' (including their labeling), is the same as that in Problem I. By

Desideratum 2.5 we have assumed that HAL represents equivalent states of knowledge by equivalent probability assignments, hence

$$p(H_1'|I) = p(H_1|I). \tag{2.11}$$

But from this observation it follows that $p(H_2|I) = p(H_1|I)$ which by Axiom (2.2) implies

$$p(H_1|I) = p(H_2|I) = 1/2,$$

i.e., we can assign a numerical value for the probabilities. It is easy to see that the above line of reasoning with symmetry equations (as equations similar to (2.11) are called [70]) can be generalized to a set $\{H_1, \ldots, H_n\}$ of n exclusive, exhaustive propositions, leading to the assignments

$$p(H_i|I) = 1/n, \qquad (1 \leq i \leq n), \tag{2.12}$$

better known as the *Principle of Indifference.* As noted by Jaynes [70], this intuitive result has important consequences for HAL's behavior; if it were to assign any values different from (2.12), by mere permutation of labels there would exist another problem where HAL's state of knowledge is the same, but it would assign different probabilities. In the experimental part of the work described here this Principle of Indifference is present in the use of uniform prior densities for model parameters.

Principle of Indifference is useful to HAL when the set of possibilities is finite, but the analysis becomes much more difficult when the set of alternatives is infinite, e.g., when we want to assign probabilities to the possible values of continuous parameters. For such an infinite case the "transformation trick" of transforming the original problem to an equivalent one may be very hard. For the finite case above, the only transformation that preserves the identity of the alternatives is permutation. In the continuous case, there is an infinite number of possible re-parameterizations. Obvious generalization of Laplace's Principle of Indifference is problematic as it lacks invariance under transformation (see [9, 118] for discussion). In one of the continuations of the work reported here [87] we illustrate the differences when more sophisticated non-informative probability assignments, such as Jeffreys' invariance principle [72], are used instead of uniform densities. For a more detailed discussion on rules for assigning non-informative probabilities in continuous cases the reader is encouraged to consult the excellent book by Berger (Chapter 3) [8] and the references therein.

2.3.2 Informative probabilities and the Principle of Maximum Entropy

In addition to the specification of alternatives $\{H_1,H_2, \dots\}$, we may have some additional information which leads us to probability assignments different from the non-informative ones discussed above. It is obvious that some types of prior information are too vague to be translatable into mathematical terms usable by HAL. Therefore in practice the minimum requirement of testability of information is needed before HAL can assign prior probabilities. The information I_E is *testable* with respect to H_i if, given any prior probability assignment over the H_i, there exists a procedure that determines unambiguously whether the assignment is consistent with the information I_E. In general there may exist several distributions consistent with testable information I_E. A classical example of testable information is the case where the prior mean is specified (e.g., the mean value of many rolls of dice being 2.5) and among the prior distributions with this mean the most non-informative distribution is sought. If we denote by $\mathcal{F}$ the operation of altering a non-informative distribution to reflect testable information I_E, we can write

$$p(H|II_E) = \mathcal{F}(p(H|I); I_E).$$

Interestingly it can be shown [70, 128] that the desiderata presented in Section 2.1 are sufficient to uniquely specify the operation $\mathcal{F}$. Thus for such case HAL will select among all the possible normalized distributions $(p_1, \dots, p_n)$ satisfying the constraints imposed by I_E, the one with maximum information entropy as defined (in the finite case) by

$$\mathcal{H}(p_1, \dots, p_n) = -\sum_{i=1}^{n} p_i \lg p_i. \tag{2.13}$$

This *Maximum Entropy Principle* assignment represents the most objective description of what HAL knows about the propositions $\{H_1,H_2, \dots\}$. Like the Principle of Indifference, also Equation (2.13) can be generalized to the infinite case, but its use comes more complicated due to the lack of a completely natural definition of entropy for continuous spaces. For illustrative worked examples of the maximum entropy assignments see [118, pp. 210–314].

Above we have discussed the justification of the Principle of Indifference based on the desiderata we assumed for HAL's inference. When HAL assigns uniform prior probabilities, it is in fact applying maximum entropy (albeit in such a simple case that the formalization and derivation in [128] are unnecessarily complex). In practice, however, much more information is commonly used in prior probability assignments by assuming that the distribution has a specific (parametric) form, which in fact expresses independence assumptions. The motivation for such assumptions is often computational and mathematical convenience, which is one of the main advantages of the use of the so called conjugate priors discussed later on.

2.4 Bayes' theorem

`HAL` can derive all legitimate relationships between probabilities from the product and sum rules in Axioms 2.1 and 2.2. There is, however, one straightforward consequence of Axiom 2.1 which has proven to be so useful that it is usually stated as a theorem.

Theorem 2.1 (Bayes' theorem, Bayes 1763, Laplace 1774) Let us assume that $\{H_1, \ldots, H_n\}$ is a set of n exclusive, exhaustive propositions, and $1 \leq j \leq n$. Then

$$p(H_j|DI) = \frac{p(H_j|I)p(D|H_jI)}{\sum_{j=1}^{n} p(D|H_jI)p(H_j|I)}.$$

Proof 2.1 Let us first derive a generalized sum rule from Axioms 2.1 and 2.2.

$$\begin{aligned} p(H+D|I) &= 1 - p(\bar{H}\bar{D}|I) \\ &= 1 - p(\bar{H}|I)p(\bar{D}|\bar{H}I) \\ &= 1 - p(\bar{H}|I)(1 - p(D|\bar{H}I)) \\ &= p(H|I) + p(\bar{H}D|I)) \\ &= p(H|I) + p(D|I)p(\bar{H}|DI)) \\ &= p(H|I) + p(D|I) - p(HD|I)). \end{aligned}$$

The equation

$$p(H+D|I) = p(H|I) + p(D|I) - p(HD|I)) \tag{2.14}$$

is clearly a generalization of the sum rule in Axiom 2.2.

Now to the actual derivation. Propositions (HD) and (DH) are obviously identical as the ordering is irrelevant. Axiom 2.1 implies that

$$p(H_iD|I) = p(H_i|DI)p(D|I) = p(D|H_iI)p(H_i|I)$$

from which we can solve

$$p(H_i|DI) = \frac{p(H_iD|I)}{p(D|I)} = \frac{p(D|H_iI)p(H_i|I)}{p(D|I)}.$$

The theorem now follows from the generalized sum rule (2.14) when it is applied to proposition $D = DH_1 + DH_2 + \ldots DH_n$. □

The four elements, $p(H_i|I), p(D|H_iI), p(H_i|DI)$ and $p(D|I)$ present in Bayes' theorem appear throughout Bayesian statistics with different names. In order to have a standard terminology we define them as follows.

Definition 2.1 Let $\{H_1, \ldots, H_n\}$ be a set of n exclusive, exhaustive propositions, and D, I propositions. Then $(1 \leq i \leq n)$

- $p(H_i|I)$ are called the *prior probabilities* of the H_i,
- $p(D|H_iI)$ are called the *likelihoods* of the H_i given D,
- $p(H_i|DI)$ are called the *posterior probabilities* of the H_i given D,
- $p(D|I)$ are called the *evidence* of the D.

The likelihood term is also known as the *sampling distribution*, and the evidence is sometimes called the *marginal likelihood* or *predictive probability* [9]. This last term such not be confused with the predictive distribution as used in this work.

Theorem 2.1 has proven to be so important in the Bayesian probability theory that the whole field is so-called due to its wide use of Bayes' theorem to assess hypotheses. This becomes obvious by giving proper interpretations to the propositions H, D and I. Typically H_i is a hypothesis `HAL` wants to assess, D some relevant data (to the hypothesis) that `HAL` has, and I some background information indicating the way in which H_i and D are related, also specifying any alternatives that may exist for H_i. Here we can see the intimate relationship between Bayes' theorem and `HAL`'s *learning*. Bayes' theorem is an "update rule" that indicates how `HAL` adjusts its plausibility assessments when the state of knowledge regarding the hypothesis changes through the acquisition of new data.

One should be careful not to misinterpret the terms "prior probability" and "posterior probability" necessarily to mean "earlier in time" and "later in time". Their semantics is only with respect to the particular chain of inference being made. The distinction is only conventional, not fundamental. Similarly the separation of the information available to prior information and data is a choice made by the modeler, and used only to organize a chain of inferences. Thus there is only one kind of probability; the different names refer only to a particular way of organizing `HAL`'s calculation in a particular case.

Up to this point `HAL`'s inference framework has only been used with a finite set of hypotheses $\{H_1, \ldots, H_n\}$. In addition the arguments of the formal probability symbol $p(\cdot)$ have been propositions such as

$H \equiv$ "Class CS101 has N students, of which M are blue-eyed. We pick one student randomly and observe."

$D \equiv$ "The student selected is blue-eyed."

From next chapter on, we will be continuously interested in hypotheses with real-valued parameters, and thus follow the custom of most contemporary works and relax our notation to allow also real-valued variables such as θ, or numerical values such as x_i as arguments, i.e., we allow for example notation

$$p(x_1, x_2, \ldots, x_{m-1}|\theta).$$

In addition, when using variables, values and sets as arguments of the probability p, to avoid ambiguity in denoting the conjunctions, we will move from using catenation to using "," instead, i.e., $p(x_i|\theta^b, M_k)$ means the conditional probability of $X_i = x_i$ given both $\theta = \theta^b$ and $M = M_k$. Finally, HAL's inference can be extended to allow continuous variables (values) θ as arguments (see the discussion in [70, p. 419]), in which case we in fact use probability density functions. For simplicity we will use the same notation for continuous density functions and discrete probability mass functions, in addition different distributions in the same expression will each be denoted by $p(\cdot)$. This slight abuse of strict mathematical accuracy will result in a compact standard notation used in much of the Bayesian literature (see e.g., [9, 55]).

Chapter 3

Models and plausible predictions

"All models are false, but some are useful."

—José Bernardo and Adrian Smith in Bayesian Theory

Chapter 2 established the general Bayesian framework for the study of plausibilities and their evolution in the light of new information. We will now turn to the detailed development of these ideas, and return to our original motivation of giving `HAL` a means for predicting yet unobserved quantities conditional on having observed some other quantities (data). Naturally `HAL`'s predictions should be based on logical inference whenever enough information is at hand to allow it. In the case of real problems, however, almost invariably the necessary information is not available and `HAL`'s "intelligence" is dependent on how optimal its processing of incomplete information is. From this chapter on we will move our focus from general plausible inference to prediction using probabilistic theories, with the criterion that the best theory is the one with the greatest predictive power (Section 3.1).

The predictive Bayesian inference discussed in this work is deeply related to the minimum encoding approaches of Wallace [144, 145], Rissanen [116, 117] and Solomonoff [130]. In Section 3.3 we briefly discuss this information theoretic formulation, which in many cases provides an interesting complementary view of the Bayesian approach, and demonstrates some advantages over the traditional approach.

3.1 Predictive models

3.1.1 The prediction problem

`HAL`'s predictions are based on conditional probabilities. This conditioning, however, needs information about on which propositions (variables) the conditionings are made. This information is expressed as a *model.* Obviously this is not particular only to Bayesian modeling, any predictive system such as HAL has to use a model of the quantities to be predicted. In very broad sense a set of models, defined for example by structural equations, can be understood as the language used to describe the constraints that the observed quantities satisfy. In our case this language is the language of computable probability mass functions or density functions, thus all HAL's models are probabilistic.

Let us now consider that HAL's events of interest are defined explicitly in terms of quantities $\vec{d}_1, \ldots, \vec{d}_L$. These quantities are typically vector-valued with continuous or discrete component values, and will be hereinafter called (observable) *data.* Furthermore, let us assume that HAL wants to predict the as yet unobserved data $D_q = \vec{d}_{N+1}, \ldots, \vec{d}_L$ on the basis of its background information and the observed data $D = \vec{d}_1, \ldots, \vec{d}_N$, $(1 \leq N < L)$. Following the discussion in Chapter 2, HAL's plausibilities for data are derived from the specification of a joint probability density function $p(\vec{d}_1, \ldots, \vec{d}_L | \mathcal{U})$, where $\mathcal{U}$ is a "universal" model family, i.e., a model space containing all the models that are available to HAL to express conditioning information. The explicit conditioning with model space $\mathcal{U}$ is usually dropped for notational simplicity; consequently, unless there is a possibility for confusion, this joint probability density will be denoted by $p(\vec{d}_1, \ldots, \vec{d}_L)$. In our Bayesian framework HAL's prediction problem can be expressed as the problem of identifying the joint conditional density $p(\vec{d}_{N+1}, \ldots, \vec{d}_L | \vec{d}_1, \ldots, \vec{d}_N)$. From Axiom 2.1 we have

$$p(\vec{d}_{N+1}, \ldots, \vec{d}_L | \vec{d}_1, \ldots, \vec{d}_N) = \frac{p(\vec{d}_1, \ldots, \vec{d}_L)}{p(\vec{d}_1, \ldots, \vec{d}_N)}, \tag{3.1}$$

i.e., HAL's (posterior) *predictive density* for new data D_q can be reduced to calculating the ratio of the joint densities $p(D_q, D)$ and $p(D)$. Following the convention adopted in the computational intelligence and machine learning communities, we call the data sets D_q and D *test data* and *training data*, respectively.

Intuitively the Bayesian ideal to calculate the predictive density $p(\cdot)$ would require HAL to use all the models in the universal model family $\mathcal{U}$, which clearly is not feasible in practice. In Sections 3.2 and 3.3 we will give three alternative definitions of a predictive density for HAL to use for prediction, all of which approximate the ideal predictive density $p(\cdot)$ by using different sets of models from the model space $\mathcal{U}$. However, first we proceed by introducing some concepts and notation used in Bayesian modeling.

3.1.2 Model families, model classes and models

Prediction and learning, whether Bayesian or not, are always relative to a set of possible models. There seems to be sometimes confusion in the machine learning literature about this issue, as for example the pure instance-based methods only store the training data D, and do not form an explicit model during learning. In such approaches the model is, however, formed for each prediction dynamically, as indicated by the more appropriate recent term "lazy learning" [3].

Let us now introduce some general notation, used subsequently in our context of predictive modeling. The training data D denotes a (random) sample of N independent and identically distributed (i.i.d.) data vectors $\vec{d}_1, \ldots, \vec{d}_N$. For simplicity, we assume here that the data is coded by using only finite-valued attributes $X_1, \ldots, X_m$. More precisely, we regard each attribute X_i as a random variable with possible values from the set $\{x_{i1}, \ldots, x_{in_i}\}$. Consequently, each data vector $\vec{d}$ is represented as a value assignment of the form $(X_1 = x_1, \ldots, X_m = x_m)$, where $x_i \in \{x_{i1}, \ldots, x_{in_i}\}$. The approach described here extends naturally to continuous attributes as well, the only exceptions being the predicted attributes for certain types of definitions of the predictive density.

In the following, let $\mathcal{M} \subseteq \mathcal{U}$ denote a *model family*, a set of models each determining some probability distribution on the problem domain. Examples of model families include the set of feed-forward neural network models [12, 95], the set of Bayesian networks [66], and the set of decision trees [74], which all can be viewed as sets of models representing probability distributions. For notational convenience, it is often useful to partition the models within a model family $\mathcal{M}$ to some finite number of subsets, *model classes* M_i, where all the models within a model class share the same parametric form, i.e., the same number of parameters. Consequently, the model classes usually correspond to some specific model structure. Examples of such structures are the topology of a feed-forward neural network or a Bayesian network. A *model* θ is here defined as a parameter instantiation within some parametric model class M_i, fully determining a probability distribution in the data vector space[1]. It should be noted that in some of the literature what we call a "model class", is called a "model", and our "model selection" (see the next section) is called "parameter estimation".

Let us now reflect all this modeling discussion to the general Bayesian framework discussed in Chapter 2. A model can be seen as a hypothesis H_i

[1]We will also sometimes use the notation $\theta(D')$ to indicate the data set D' from which the model is constructed. Usually $D' = D$, and in such cases for notational simplicity we drop D and use simply θ instead of $\theta(D)$.

from a continuous range of hypotheses, the data D is the "problem-specific" information given to HAL, and the choice of the set of models to be considered ($\mathcal{M}$ and partitioning $\{M_i\}$) is the background information I.

One of the advantages of the Bayesian approach is that the assumptions of the model family are made explicit. This helps us to avoid the confusion often present in the computational intelligence and machine learning literature, where in many cases it is difficult to see which set of models is searched during the learning process, and what models are preferred in the first place. The latter point corresponds to the Bayesian priors, where the preferences, or the lack of them, are also made explicit. In addition the Bayesian framework also allows clear conceptual separation between the criteria of the quality of a model and the algorithm searching for good models, instead of coding the quality measure into the search algorithms itself. This separation of the search mechanism and the model quality measure offers HAL more flexibility—it can use different search algorithms for the same criteria. Although all the work reported here is based on a particular search algorithm and its variants, various competitive alternatives exist [84].

3.1.3 Prediction and the posterior density for models

Let us assume that we have programmed HAL by defining a model family $\mathcal{M}$ with models θ partitioned into model classes $\{M_i\}$, where each model class has the same parametric form θ_{M_i}. Furthermore, we provide HAL with the prior information $p(\theta|\mathcal{M})$. We now give HAL some additional information in the form of training data D, and ask it to give a predictive distribution for a new data vector $\vec{d}$, i.e., to give $p(\vec{d}|D)$ (since we have fixed HAL's model family, all the discussion is relative to $\mathcal{M}$ and for notational simplicity we drop it from the conditioning part). How should HAL predict?

The answer is somewhat surprising, especially for those used to thinking learning in terms of building non-probabilistic models. HAL's predictive distribution for a new data vector is given by

$$\begin{aligned} p(\vec{d}|D) &= \int_{\mathcal{M}} p(\vec{d},\theta|D)d\theta \\ &= \int_{\mathcal{M}} p(\vec{d}|D,\theta)p(\theta|D)d\theta \\ &= \int_{\mathcal{M}} p(\vec{d}|\theta)p(\theta|D)d\theta \qquad (3.2) \end{aligned}$$

The second equation follows from Axiom 2.1 and the third one from the conditional independence of $\vec{d}$ and D given θ. The integrand in (3.2) requires HAL to calculate the posterior density $p(\theta|D)$. Since we provided HAL with the prior $p(\theta|\mathcal{M})$, it can apply Bayes' theorem. Thus the Bayesian predictive distribution is achieved by *model averaging*, i.e., by combining the predictions of all the models in $\mathcal{M}$ weighted by their posterior density. This weighted averaging over models reflects HAL's uncertainty about the model, i.e., this predictive distribution is the expected distribution of $\vec{d}$ over all possible parameter settings (models). Notice that if the training data D severely restricts the set of possible distributions, the posterior density will be highly peaked, and the effect is approximately the same as if only a single model θ were used. Intuitively this is the reason for the observation that with large training data sets, HAL usually can predict well already with a single model—the additional data reduces the uncertainty about the parameters thus making the posterior more peaked, in which case the single model approximation is more viable.

In most practical situations averaging over all the models in a model family is not feasible, and HAL should try to approximate this average. One general approach to approximate the integral in Equation (3.2) is to find a good model structure, and then use all the models with that structure. In our framework this corresponds to the case, where HAL uses model class posterior to find a highly probable model class $\hat{M}$ indicated by the training data D, and then averages over all the models in $\hat{M}$, i.e.,

$$p(\vec{d}|D) = \int_{\hat{M}} p(\vec{d}|\theta)p(\theta|D)d\theta. \tag{3.3}$$

Notice that Equation (3.3) still allows HAL to use an infinite number of models.

Equation (3.3) requires an integral over all models θ of fixed structure. If this integral is hard to compute, HAL can make a further approximation by using model posterior (within a model class) to select a single good model $\hat{\theta}$, and then perform the predictions with that one model. It should be noticed that the prevailing approaches in the traditional machine learning and computational intelligence achieve usually even less, since instead of using the posterior, they typically resort to assigning *ad hoc* selection scores to models. In the light of this hierarchy of approximations it is easy to see why the traditional, single model based approaches in machine learning have only been moderately successful, in particular with small data sets D. It also explains why the new averaging approaches such as bagging [16] or boosting [50] have produced improved results over the use of individual models.

In both of the approximation steps above, HAL faces the problem of selecting the best representative of a posterior distribution. Let us now investigate whether the Bayesian framework introduced in Chapter 2 offers HAL any help for solving also this selection problem.

Summarizing the posterior density by using an estimate is a nontrivial problem, since its solution is outside the pure inference framework and requires Bayesian decision theory [8]. Bayesian probability theory alone can only solve the inference problem, i.e., can provide only a probability density representing HAL's state of knowledge. The criteria for selecting a single model to represent this density is dependent on how the model will be used. Thus the task of selecting a single model θ' summarizing the posterior is dependent on a loss function $l(\theta', \theta)$, which describes the "loss" occurring if θ' is used instead of the model θ. The Bayesian approach to finding the best representative model θ^b requires then the minimization of the expected loss [9, pp. 255–258]:

$$\theta^b = \arg\min_{\theta'} \int_{\mathcal{M}} l(\theta', \theta) p(\theta|D) d\theta.$$

In many cases the loss function $l(\theta', \theta)$ is hard to define or not known. Therefore it is common practice to use the mode of the posterior $\hat{\theta}$ as θ^b to summarize the posterior distribution[2]. If the posterior is very strongly peaked, the predictive distribution of (3.2) can be replaced by the prediction made using the mode $\hat{\theta}$, with a negligible loss of accuracy. Therefore HAL will use the posterior mode at each level of selection:

i. **Model family selection**
In the extreme case we could imagine that HAL could also choose between alternative model families. In such cases the model family could be chosen by maximizing the posterior probability $p(\mathcal{M}|D, \mathcal{U})$,

$$\hat{\mathcal{M}} = \arg\max_{\mathcal{M}} p(\mathcal{M}|D, \mathcal{U}),$$

where $\mathcal{U}$ is the set of all possible models of any form[3]. This extreme case of computing the model family posterior density for all the possible model families is a hopelessly intractable task. Therefore any $\texttt{HAL}_{\mathcal{M}}$ is designed to work with some model family $\mathcal{M}$ fixed in advance, based on designer's prior knowledge or personal preferences.

ii. **Model class selection**
As opposed to model family selection, model class selection is commonly encountered in practice. The model class to be selected is the

[2] It should be noted that the most commonly used choices for θ^b, the mode and the mean of the posterior, have no fundamental status in Bayesian inference as they both change under nonlinear re-parameterizations.

[3] For this case the model families $\mathcal{M}_i$ become model classes of the "universal" model family $\mathcal{U}$.

class with the maximal posterior probability,

$$\hat{M} = \arg\max_{M} p(M|D, \mathcal{M}).$$

From Bayes' theorem we know that

$$p(M|D, \mathcal{M}) = \frac{p(D|M, \mathcal{M})p(M|\mathcal{M})}{p(D|\mathcal{M})}.$$

The denominator $p(D|\mathcal{M})$ is a constant, and can be ignored. In many cases we can assume that all the model classes are equally probable a priori (according to the Principle of Indifference). Then it is sufficient to maximize $p(D \mid M, \mathcal{M})$, i.e., the evidence (see Definition 2.1) of the data,

$$p(D|M, \mathcal{M}) = \int p(D|\theta, M, \mathcal{M})p(\theta|M, \mathcal{M})d\theta. \tag{3.4}$$

iii. **Model selection**
Finally, selecting a single model within a model class M corresponds to choosing the *maximum a posteriori (MAP)* values $\hat{\theta}$ of the parameters,

$$\hat{\theta} = \arg\max_{\theta} p(\theta|D, M, \mathcal{M}).$$

Again from Bayes' theorem we have

$$p(\theta|D, M, \mathcal{M}) \propto p(D|\theta, M, \mathcal{M})p(\theta|M, \mathcal{M}), \tag{3.5}$$

analogously to the model class selection case.

Due to the update nature of Bayes' theorem in posterior calculation, it has also proven to be useful to introduce the notion of *conjugacy*, and use so called natural conjugate priors [55]. For a given likelihood $p(D|\theta)$ all densities $p'(\cdot)$ having a functional form such that the posterior $p(\theta|D)$ will follow the same form as $p'(\cdot)$, are called conjugate priors of $p(D|\theta)$. In addition to computational convenience, conjugate priors have the practical advantage of being interpretable as additional data. In this work we will always be using conjugate priors, which in our case will be Dirichlet densities [55].

Before concluding the discussion on posterior densities, one common source of confusion should be pointed out. The assumption of uniform priors for model classes does not mean that HAL would consider complex structures equally plausible to simple structures. The data-dependent term $p(D|M, \mathcal{M})$ in Bayes' theorem, i.e., the likelihood, embodies preference for simpler models automatically[4], which can intuitively explained as follows. Complex model classes, by their nature, are capable of making a greater variety of predictions than simple ones. Therefore, if model class M_c has more complex structure than M_s, the likelihood $p(D|M_c, \mathcal{M})$ is spread more thinly over the data space than $p(D|M_s, \mathcal{M})$. In the case where the data are compatible with both model classes, the simpler model class M_s will turn out to be more probable than M_c, without the need to assign an additional "penalizing" prior for complex structures.

3.2 Bayesian predictive inference

Let us now return to `HAL`'s general prediction problem as described by Equation (3.1). Here we formulate a restricted version of the problem, where `HAL` needs to predict the value of a single attribute of a new partially observed data vector. This restricted version corresponds to a typical attribute-based classification problem in machine learning and computational intelligence [75].

As discussed in Section 3.1.1, given the training data D, `HAL` should base its predictions on the conditional distribution of a new *test vector* $\vec{d}$, that is on $p(\vec{d}|D)$, where

$$p(\vec{d}|D) = \frac{p(\vec{d}, D)}{p(D)}. \tag{3.6}$$

We now focus on the following prediction problem: Given the values of the variables $X_1, \ldots, X_{m-1}$, and the training data D, `HAL` needs to predict the value of variable X_m. For notational simplicity, in the sequel we drop the variable names, and denote a value assignment

$$(X_1 = x_1, X_2 = x_2, \ldots, X_{m-1} = x_{m-1})$$

by writing $(x_1, x_2, \ldots, x_{m-1})$. Now for each possible value x_{mi},

$$x_{mi} \in \{x_{m1}, \ldots, x_{mn_m}\}$$

we wish to compute the probabilities

$$p(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D).$$

[4] This preference for simplicity is usually know as the "Ockham's Razor" and credited to William of Ockham (c. 1285-1349).

Let $\vec{d}[x_{mi}]$ denote the vector $(X_1 = x_1, \ldots, X_{m-1} = x_{m-1}, X_m = x_{mi})$. From Axiom 2.1 we know that

$$p(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{p((x_1, \ldots, x_{m-1}), X_m = x_{mi}, D)}{p((x_1, \ldots, x_{m-1}), D)},$$

from which by writing explicitly the marginalization we get

$$p(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{p((x_1, \ldots, x_{m-1}), X_m = x_{mi}, D)}{\sum_{k=1}^{n_m} p((x_1, \ldots, x_{m-1}), X_m = x_{mk}, D)}.$$

Now using the $\vec{d}[x_{mi}]$ notation we have

$$\begin{aligned} p(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) &= \frac{p(\vec{d}[x_{mi}], D)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}], D)} \\ &= \frac{p(\vec{d}[x_{mi}]|D)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}]|D)}. \end{aligned} \tag{3.7}$$

Consequently, the conditional distribution for variable X_m can be computed by using the complete data vector conditional distributions (3.6) for each of the possible complete vectors $\vec{d}[x_{mi}]$. The resulting distribution is called the *predictive distribution* of X_m. This approach can be straightforwardly extended to cases with more than one uninstantiated variable, but it should be noted that in the general case the number of comparisons needed grows exponentially with respect to the number of free variables. In Chapter 4 we will see that for certain model families `HAL` can in fact address this general problem more efficiently.

`HAL`'s ability to express constraints for the data was restricted to the language of some parametric family of models $\mathcal{M}$, where each instantiation of parameters θ corresponds to a single distribution. We now turn to the problem that given this language, how should `HAL` define $p(\cdot)$. In the following we consider two alternative definitions for $p(\cdot)$, and in Section 3.3.1 we will add a third, minimum encoding based approach introduced by Rissanen [117].

3.2.1 The MAP predictive distribution

The first definition for the distribution $p(\cdot)$ is the most elementary one, i.e., the one based on the best model in a fixed model class M. Since here our discussion will always be relative to a fixed model family $\mathcal{M}$, to simplify notation we drop $\mathcal{M}$ from the conditioning part. Given a prior distribution $p(\theta|M)$ over the space of parameters, we can arrive at a posterior distribution $p(\theta|D, M)$ by using Bayes' theorem:

$$p(\theta|D, M) \propto p(D|\theta, M)p(\theta|M). \tag{3.8}$$

In the *maximum a posteriori (MAP) probability* approach, distribution $p(\cdot)$ is replaced by the distribution conditioned by the single model

$$\hat{\theta}(D) = \arg\max_{\theta} p(\theta|D, M),$$

i.e., the mode of the posterior distribution $p(\theta|D, M)$:

$$p_{\text{map}}(\vec{d}, D) = p(\vec{d}, D|\hat{\theta}, M).$$

In orthodox statistical approach, the MAP model is replaced by the *maximum likelihood (ML)* model $\tilde{\theta}$, i.e., by the model maximizing the data likelihood $p(D|\theta, M)$. Since throughout this work we use non-informative priors (see Section 2.3.1), and assume the prior distribution $p(\theta|M)$ to be uniform, the MAP model is equal to the ML model, as can clearly be seen from (3.8). Now the corresponding predictive distribution (3.7) is in this case

$$\begin{aligned} p_{\text{map}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) &= \frac{p_{\text{map}}(\vec{d}[x_{mi}], D)}{\sum_{k=1}^{n_m} p_{\text{map}}(\vec{d}[x_{mk}], D)} \\ &= \frac{p(\vec{d}[x_{mi}], D|\hat{\theta}, M)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}], D|\hat{\theta}, M)}. \end{aligned}$$

Since the data are i.i.d., $\vec{d}[x_{mi}]$ and D are independent given θ, thus

$$\begin{aligned} p_{\text{map}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) &= \frac{p(\vec{d}[x_{mi}]|\hat{\theta}, M)p(D|\hat{\theta}, M)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}]|\hat{\theta}, M)p(D|\hat{\theta}, M)} \\ &= \frac{p(\vec{d}[x_{mi}]|\hat{\theta}, M)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}]|\hat{\theta}, M)}. \end{aligned} \tag{3.9}$$

3.2.2 The evidence predictive distribution

A more sophisticated definition for the distribution $p(\cdot)$ is based on the observation that instead of using a single model from a fixed model class M, `HAL` could use *all the models in that class.* This can be achieved by integrating over the model class M, i.e., by averaging over all the models θ (here we again drop $\mathcal{M}$ from our notation):

$$p_{\text{ev}}(\vec{d}, D) = \int p(\vec{d}, D|\theta, M)p(\theta|M)d\theta. \tag{3.10}$$

The integral in Equation (3.10) is in fact the normalization factor of Bayes' theorem, i.e., the *evidence.* The resulting predictive distribution can now be given as

$$p_{\text{ev}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{p_{\text{ev}}(\vec{d}[x_{mi}], D)}{\sum_{k=1}^{n_m} p_{\text{ev}}(\vec{d}[x_{mk}], D)}.$$

From (3.10) we get

$$p_{\text{ev}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{\int p(\vec{d}[x_{mi}], D|\theta, M)p(\theta|M)d\theta}{\sum_{k=1}^{n_m} \int p(\vec{d}[x_{mk}], D|\theta, M)p(\theta|M)d\theta}. \tag{3.11}$$

Computing the MAP predictive distribution (3.9), or at least a good approximation of it, is often feasible in practice. In particular it is feasible for the model family of finite mixtures discussed in Chapter 4. Calculating the evidence predictive distribution (3.11) —or even good approximations to it—is very hard for most model families. We will show, however, that the evidence predictive distribution can be computed efficiently for a special class of discrete finite mixture models, which corresponds to the Naive Bayes classifier in the machine learning literature (see e.g., [82]).

3.3 Information theoretic view

The Bayesian predictive inference framework we have discussed has an interesting information theoretical formulation—prediction of data with minimal descriptions. From elementary information theory [31] we know that for any complete code C , there is a corresponding probability distribution p_C such that for all sets D, $-\log p_C(D)$ is the length of the encoding of D when the encoding is done using C. Similarly, for all probability distributions p over data sets D there is a code C_P such that for any data set D the code-length of D when encoded with C_P is equal to $\lceil -\log p(D) \rceil$. Thus we can equate HAL's probabilities of data with the lengths (in bits) of messages which communicate data without loss to a receiver.

The intuition behind the inference in the information theoretic approach is that the most probable model has *the shortest encoding of the model and data combined*. We have learned that the most probable model in the Bayesian approach is defined by the mode of the posterior, which is calculated by Bayes' theorem (here we again drop the model family $\mathcal{M}$ from the notation)

$$p(\theta|D) = \frac{p(D|\theta)p(\theta)}{p(D)}. \tag{3.12}$$

Taking the negative logarithm of this expression turns the products into sums, and gives us

$$-\log p(\theta|D) = -\log p(D|\theta) - \log p(\theta) + \text{constant}. \tag{3.13}$$

Since HAL is only interested in the relative probability of the different models θ, the last term in Equation (3.12) can be ignored. Now the connection between Bayesian probability theory and the coding approach becomes clear: from information theory we know that $-\log p(\vec{d}_i)$ is the minimum message length to encode a particular data vector $\vec{d}_i$[5].

The minimum message length in (3.13) is the sum of two terms. The first term is the information required to encode the data, given the model θ, and decreases for suitably selected more complex models. The second term is the information to describe the model θ, which is larger for more complex, and thus less probable, models. This type of codes are called *two-part codes*.

Theoretically this idea of predicting sequences using minimal effective descriptions dates back to the formulation by Solomonoff [130]. In various forms aiming at practical applications, the idea of data modeling in statistical prediction by encodings was proposed by Wallace et al. [144, 145] and Rissanen [114, 115, 116] leading to related two-part code formulations known as *Minimum Message Length (MML)* and *Minimum Description Length (MDL)* principles. These two principles, MML and MDL, are very seldom distinguished in the literature, which is understandable as both involve coding the parameters of models and then selecting the model with the shortest two-part message length.

Although both MML and MDL involve a process of "coding the parameters," the codes used differ in many important respects. MML is a Bayesian approach as it requires an explicit prior distribution on parameters. MDL rejects the use of priors and uses distribution independent universal codes. A universal code tries to assign every possible value of the parameter the same prior probability (for discussion on universal codes see [116]). By not including a prior, MDL codes differ from MML codes by at least a constant, and can affect the model chosen if the data set is small. Asymptotically there is no difference, the model chosen will be the same. Predictive inference in this coding framework now involves a minimization problem, where HAL should select the model θ, which minimizes the code length $-\log p(\theta|D)$.

Given the existing "traditional" Bayesian probabilistic formulation, a natural question is what added value can be gained by this coding framework? First of all, the description length concept is useful for motivating prior probability distributions, in fact in many cases it is easier to express a structural prior as a coding of the model rather than an explicit distribution over the possible model structures. Second, the coding approach has interesting advantages over the traditional Bayesian maximum a posteriori model

[5] We will always assume that the base of the log is 2, and thus the message length is in bits.

selection. As pointed out earlier, the mode of the posterior is not invariant under nonlinear transformations of the parameter space, whereas MML is known to be invariant under one-to-one transformations [145]. Third, MML/MDL has its uses as pedagogical tool—many of the Bayesian concepts can be elegantly introduced, at least for computer scientists, with the coding analogies.

Without going deeply into technical details which lay outside the scope of this work, it is not possible to discuss the existing fine distinctions between the two approaches. For our present purposes it is enough to restate that the use of either of these two-part code approaches for prediction corresponds to `HAL`'s maximum a posteriori prediction with a single (most probable) model. From the Bayesian framework we know, however, that for predictive purposes we can do better when we average predictions over a set of models. The corresponding information theoretic notion for model averaging is the *stochastic complexity (SC)* [114, 116].

There seems to be even more confusion about the relationship between the notion of stochastic complexity and MML/MDL encodings. MML/MDL methods use a single model to encode the data, and thus encode also the parameters, stochastic complexity uses a set of models M to encode the data. Let $L_C(D)$ be the code-length for data D when coded by the code C. It is always true that

$$L_{\mathrm{SC}}(D) < L_{\mathrm{MML/MDL}}(D),$$

although if the data set D is large, it is quite likely that the single best model will do nearly as well as predicting with the full model class (which we of course know already from the Bayesian interpretation). Due to the correspondence of complete codes and probability distributions we know that the stochastic complexity code can be written as $-\log p_{\mathrm{sc}}$ where p_{sc} is a probability distribution that, in a sense to be explained later, gives as much probability as possible to all D. Therefore `HAL` can use p_{sc} for prediction. Let us now discuss this third, and final one, of `HAL`'s alternative definitions for the predictive distribution.

3.3.1 The stochastic complexity predictive distribution

The original definition of the stochastic complexity, as given in [115], is the minus logarithm of the evidence, i.e., $-\log p_{\mathrm{ev}}(D)$. Recently, however, Rissanen [117] has shown that there exists an improved code that is itself not dependent on any prior distributions of parameters, and which in general yields even shorter code-lengths than the code with lengths $-\log p_{\mathrm{ev}}(D)$. Here by shorter we mean that for some data sets the code-length will be considerably shorter, while for most data sets it will be only negligible longer. Hence p_{sc} will give a much higher probability than p_{ev} to some data sets and approximately equal probability to all other ones. In the case of discrete data, the new stochastic complexity for data $(\vec{d}, D)$ with respect to model class M can be written as $-\log p_{\mathrm{sc}}(\vec{d}, D)$ with

$$p_{\mathrm{sc}}(\vec{d}, D) = \frac{P(\vec{d}, D|\tilde{\theta}(\vec{d}, D), M)}{\sum_{\vec{d}', D'} P(\vec{d}', D'|\tilde{\theta}(\vec{d}', D'), M)}, \tag{3.14}$$

where the sum in the denominator goes over all the possible instantiations of the data set $D \cup \vec{d}$, and $\tilde{\theta}(\vec{d}, D) \in M$ denotes the maximum likelihood model for this data $(\vec{d}, D)$.

In [117] it is shown that under some regularity conditions on the class of models, which hold for the finite mixture models studied here, p_{sc} is asymptotically equivalent to p_{ev} when p_{ev} is used with Jeffrey's prior. Therefore the "old" and "new" definition of stochastic complexity are asymptotically the same, but this does not necessarily hold for small data sets.

We can now obtain the stochastic complexity predictive distribution as follows:

$$p_{\mathrm{sc}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{p_{\mathrm{sc}}(\vec{d}[x_{mi}], D)}{\sum_{k=1}^{n_m} p_{\mathrm{sc}}(\vec{d}[x_{mk}], D)}.$$

From (3.14) we get

$$p_{\mathrm{sc}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{\frac{p(\vec{d}[x_{mi}], D|\tilde{\theta}(\vec{d}[x_{mi}], D, M))}{\sum_{\vec{d}', D'} p(\vec{d}', D'|\tilde{\theta}(\vec{d}', D'), M)}}{\sum_{k=1}^{n_m} \frac{p(\vec{d}[x_{mk}], D|\tilde{\theta}(\vec{d}[x_{mk}], D), M)}{\sum_{\vec{d}', D'} p(\vec{d}', D'|\tilde{\theta}(\vec{d}', D'), M)}}. \tag{3.15}$$

At first sight, this probability may seem hard to compute as we have to sum over all the exponentially many possible instantiations of the data set $D \cup \vec{d}$. But a closer inspection reveals that the two exponential sums in the rightmost part of (3.15) cancel out and thus we obtain:

$$p_{\mathrm{sc}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) = \frac{p(\vec{d}[x_{mi}], D|\tilde{\theta}(\vec{d}[x_{mi}], D), M)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}], D|\tilde{\theta}(\vec{d}[x_{mk}], D), M)}.$$

Since data are i.i.d., $\vec{d}[x_{mi}]$ and D are independent given $\tilde{\theta}$, thus

$$\begin{aligned} p_{\mathrm{sc}}(X_m &= x_{mi}|(x_1,\ldots,x_{m-1}),D) \\ &= \frac{p(\vec{d}[x_{mi}]|\tilde{\theta}(\vec{d}[x_{mi}],D),M)p(D|\tilde{\theta}(\vec{d}[x_{mi}],D),M)}{\sum_{k=1}^{n_m} p(\vec{d}[x_{mk}]|\tilde{\theta}(\vec{d}[x_{mk}],D),M)p(D|\tilde{\theta}(\vec{d}[x_{mk}],D),M)}. \end{aligned} \tag{3.16}$$

This formula looks similar to that of the maximum likelihood predictor (3.9). However, it should be noted that the probabilities

$$p(D|\tilde{\theta}(\vec{d}[x_{mk}],D,M)$$

do not cancel out here since the maximum likelihood estimator appearing in the denominator of (3.16) depends on x_{mk} and hence is not a constant. Moreover, the maximum likelihood estimator $\tilde{\theta}(\vec{d}[x_{mi}],D)$ is now computed by using the augmented data set $D \cup \vec{d}$, not just D.

Chapter 4

Predicting with finite mixtures

"In the back of his book," Norman remembered, "Thorp announces in a cryptic sentence or two that he devised, but failed to implement, a system to beat roulette. I thought on reading this that it was utter hogwash. Roulette is a random game. You can't devise any betting scheme that will win. But on rereading Thorp, we realized that you might be able to develop a predictive scheme, and that's what he was talking about."

—Thomas A. Bass in The Newtonian Casino

In the previous chapter we discussed `HAL`'s plausible predictions by predictive Bayesian inference relative to an arbitrary model family $\mathcal{M}$. We will now proceed by fixing a particular set of models, which will give `HAL` a concrete instance of the general predictive framework. In Section 4.1 we first present the family of finite mixtures of multinomials, and briefly discuss its advantages with respect to some commonly used model families in machine learning (Section 4.2). We then proceed to describing in detail, how `HAL` performs Bayesian predictive inference with finite mixtures (Section 4.3).

In Chapter 3 we presented three alternative schemes to perform predictions. If `HAL`'s model family $\mathcal{M}$ is restricted to a subset of the general finite mixture model, all these three prediction methods (together with Bayesian induction of the required models) can be implemented without the approximations needed in the general case. This makes it possible for us to perform an empirical study of the prediction performance and learning rate of these three methods—a comparison which to our knowledge is first of its kind (Section 4.4).

4.1 Finite mixture models

Previously we have already committed `HAL` to model its problem domain by m discrete variables $X_1, \ldots, X_m$ (we can assume that continuous values, if needed, are discretized), and that a data vector $\vec{d_i} \in D$ is sampled from the joint distribution of the variables $X_1, \ldots, X_m$. We now make an additional modeling assumption that the data D can be viewed as if it were generated by K different mechanisms, which all can have a distribution of their own. Furthermore, it is assumed that each data vector originates from exactly one of these mechanisms. Whether or not this actually is the case, is not of importance here. As we have already pointed out, model family is only a language in which `HAL` can express the constraints in data. From these assumptions it follows that the data vector space is divided into K local regions usually called *clusters* or *classes*, each of which consists of the data vectors generated by the corresponding mechanism.

The underlying intuitive idea is that a set of data vectors can be modeled by describing a set of clusters, and then describing the data vectors using these cluster descriptions. Each description gives the distribution of the variables $X_1, \ldots, X_m$, conditioned that the data vector belongs to the cluster. The cluster descriptions should be chosen in such a way that the information required to describe data vectors in the cluster can be significantly reduced because they are similar to the "prototype" described by the cluster. In such "cluster language" a data set D can be described by first giving the cluster index for each data vector, and then by describing the differences between the observed and expected values.

An appropriate model family for this type of modeling is the set of *discrete finite mixtures* ([46], [138]), where the joint domain probability distribution is approximated as a weighted sum of mixture distributions.

Definition 4.1 Let $X_1, \ldots, X_m$ be a set of m $(m \geq 1)$ discrete (random) variables, and $\vec{d} \in D$ is a sample from the joint distribution of the variables $X_1, \ldots, X_m$. Then the *finite mixture* distribution for $\vec{d}$ can be written as $(K \geq 1)$

$$\begin{aligned} p(\vec{d}) &= p(X_1 = x_1, \ldots, X_m = x_m) \\ &= \sum_{k=1}^{K} \left[p(Y = y_k) p(X_1 = x_1, \ldots, X_m = x_m | Y = y_k) \right], \end{aligned}$$

where Y denotes a latent *clustering random variable*, the values of which are not given in the data D, and K is the number of possible values of Y.

Thus in finite mixture models the problem domain probability distribution is approximated by a weighted sum of component distributions, where each mixture component $p(X_1 = x_1, \ldots, X_m = x_m | Y = y_k)$ models one data producing mechanism. The finite mixture model family is universal in the sense that it can approximate any distribution arbitrarily close as long as a sufficient number of components is used [138].

If the variables $X_1, \ldots, X_m$ are independent given the value of the clustering variable Y, (4.1) becomes

$$p(\vec{d}) = \sum_{k=1}^{K} \left(p(Y = y_k) \prod_{i=1}^{m} p(X_i = x_i | Y = y_k) \right). \tag{4.1}$$

In this work we will always make this independence assumption and consequently use (4.1).

Finite mixtures as defined in Equation (4.1) is a generic model family, since we still have to fix the cluster distribution $p(Y)$ and the intra-class conditional distributions $p(X_i | Y = y_k)$[1]. Most commonly used component functions in the literature are the univariate normal distributions (see e.g., [138]). `HAL` models its problem domains by discrete variables X_i, thus it is not necessary to make an assumption of the form of the distribution. Consequently a natural choice for the intra-class conditional distribution is the multivariate generalization of the binomial distribution called the *multinomial distribution* [55], i.e, the likelihood function is given by

$$p(X_i = x_j | \theta) = \theta_j, j = 1, \ldots, x_n,$$

where parameter θ_j represents the probability of the value x_j. Analogously we assume that the cluster distribution $p(Y)$ is multinomial. Thus in order to get a model, `HAL` needs to fix the number of the mixing distributions (K), i.e., the model class M, and determine the values of the model parameters. For technical reasons it will be convenient to make a notational distinction between the mixture weight parameters and the parameters of the intra-class conditional distributions. Therefore let us denote the mixture weight probabilities by

$$\alpha_k = p(Y = y_k),$$

and the conditional distributions by

$$\phi_{kil} = p(X_i = x_{il} | Y = y_k).$$

[1] Here we consider only mixtures in which all the component distributions come from the same parametric class.

Now if we denote

$$\phi_{ki} = (\phi_{ki1}, \ldots, \phi_{kin_i}),$$

we can express HAL's model as $\theta = (\alpha, \phi), \theta \in \mathcal{M}$, where

$$\alpha = (\alpha_1, \ldots, \alpha_K) \quad \text{and} \quad \phi = (\phi_{11}, \ldots, \phi_{1m}, \ldots, \phi_{K1}, \ldots, \phi_{Km})$$

Finally, HAL needs to fix the prior distributions for the parameters. As discussed earlier (see Section 3.1.3), in order to allow HAL to use posteriors as new priors in further applications, the functional form of parameter distribution should remain invariant in the prior-to-posterior transformation. The family of Dirichlet (multivariate Beta) densities is conjugate to the family of multinomials [37, 55], therefore we assume that HAL's prior distributions of the parameters are from the Dirichlet family.

Let k index the mixture components, i the variables, and l the values of a variable, i.e.,

$$1 \leq k \leq K; 1 \leq i \leq m; 1 \leq l \leq n_i.$$

Then the prior distributions for the parameters in θ are

$$(\alpha_1, \ldots, \alpha_K) \sim \mathrm{Di}\,(\mu_1, \ldots, \mu_K) \;\text{ and }\; (\phi_{ki1}, \ldots, \phi_{kin_i}) \sim \mathrm{Di}\,(\sigma_{ki1}, \ldots, \sigma_{kin_i})\,.$$

Here μ_k and σ_{kil} are called the *hyper-parameters* of the corresponding distributions. The general form of the Dirichlet density used above is (see e.g., [55, p. 477],)

$$\begin{aligned} p(\theta) &= \mathrm{Di}(\theta|\mu_1, \ldots, \mu_K) \\ &= \frac{\Gamma(\sum_{k=1}^{K} \mu_k)}{\prod_{k=1}^{K} \Gamma(\mu_k)} \prod_{k=1}^{K} \theta_k^{\mu_k - 1}. \end{aligned} \tag{4.2}$$

The density (4.2) is subject to constraints $\sum_{k=1}^{K} \theta_k = 1$, $\mu_k > 0$ and $\theta_k \geq 0$. Assuming that the parameter vectors α and ϕ_{ki} are independent, the joint prior distribution of all the parameters can be expressed as a product of Dirichlet densities

$$\mathrm{Di}\,(\mu_1, \ldots, \mu_K) \prod_{k=1}^{K} \prod_{i=1}^{m} \mathrm{Di}\,(\sigma_{ki1}, \ldots, \sigma_{kin_i})\,.$$

We have pointed out earlier that the finite mixture model family is universal in the sense that it can approximate any distribution arbitrarily close. Unfortunately such generality of a model family typically implies also that certain types of prediction methods, in particular calculating the evidence predictive distribution, become computationally very costly. There is, however, a simple class of discrete finite mixtures for which HAL can compute all the predictive distributions discussed in Section 3.2 efficiently without approximations. This class follows from Equation (4.1) when we remove the latency of Y and assume that one of the variables $X_1, \ldots, X_m$ gives us the partitioning of the data (for notational simplicity we will assume that it is always X_m). This new model family $\mathcal{M}_{\mathrm{NB}}$ corresponds to a specific model class M_{n_m} of the more general case, thus the joint probability distribution for a data vector $\vec{d}$ can be expressed as

$$\begin{aligned} p(\vec{d}) &= p(X_1 = x_1, \ldots, X_m = x_m) \\ &= p(X_m = x_m) \prod_{i=1}^{m-1} p(X_i = x_i | X_m = x_m). \end{aligned} \tag{4.3}$$

The connection to the general mixture formulation of Equation (4.1) can be seen if we write (4.3) in the form

$$\begin{aligned} p(\vec{d}) &= p(X_1 = x_1, \ldots, X_{m-1} = x_{m-1}, X_m = k) \\ &= \sum_{j=1}^{n_m} \left(P(X_m = j) \prod_{i=1}^{m-1} p(X_i = x_i | X_m = k) \right). \end{aligned}$$

This model family $\mathcal{M}_{\mathrm{NB}}$ is known in the machine learning community as the Naive Bayes or Simple Bayes classifier [43, 82]. Despite of its simplicity and apparent strict conditional independence assumptions, it has shown competitive performance when compared to the results achieved by more complex model families [39, 99]. Naive Bayes model family illustrates also nicely the fact that in classification tasks for the general mixture models the partitioning of the data to clusters is not necessarily based on the class variable value assignments. In some cases the latent variable Y based clustering can coincide with the class value based clustering, but in the general case even the number of clusters differs from the number of classes n_m. Naive Bayes model family $\mathcal{M}_{\mathrm{NB}}$ results if we *assume* these clusterings to be equal.

4.2 Finite mixtures in perspective

Due to the contrast caused by the generality of the generic Bayesian approach, which is "parameterized" by using the notion of model family, one might underestimate the flexibility of the finite mixtures modeling language. It is well known that the finite mixtures of multinomials correspond to a special subclass of Bayesian networks. On the other hand such model families as Specht's probabilistic neural networks [131], kernel estimators [124]—and consequently also Radial Basis functions [100])—together with instance-based (memory-based) models [2, 4], are instantiations of finite mixture models, if viewed in the probabilistic framework (albeit some of them mixtures of Gaussians instead of mixtures of multinomials). Below we will also discuss the interesting interpretation of the lazy learning approach from the Bayesian perspective.

4.2.1 Related work

Neural approaches. Finite mixture models are an ideal model family for massively parallel hardware [103], and thus can be realized by neural architectures [98, 102]. In continuous domains mixture models have proven to be a viable family for supervised learning tasks as demonstrated by the "mixtures of experts" architecture of Jordan et al. [74, 146]. In supervised learning one assumes that the attribute to be predicted is known at model construction time, and this information is used in the model construction process. In HAL's case, instead of the supervised learning framework, we explore the unsupervised case where we do not fix in advance the predictive distributions to be estimated, and build a full probability model of the problem domain. Therefore the most related work to ours in the neural network community is the work by Bishop et al. on "mixture density networks", which, however, are mixtures of Gaussians (see [11, 13]). In addition the Bayesian approach can also be used for constructing Self-Organizing Maps [80], which brings the method close to HAL's model construction—again with the difference that mixtures of Gaussians are used [14, 142].

AutoClass. AutoClass system [23, 24] is a Bayesian classification[2] program, which uses Bayesian mixture modeling for discovering "natural" classes in data. In our terminology it performs Bayesian induction to find cluster descriptions, i.e., mixture components for explorative purposes. Consequently, AutoClass work is interested in finding the most probable mixture model given the data and the priors, and then use these descriptions in a knowledge discovery process, e.g., LandSat data clustering [24]. In HAL's design we are only interested in the predictive capability of our models, not their semantic interpretation. The work for HAL presented here combines both Bayesian deductive inference and Bayesian induction with mixtures and is unique in this respect.

Bayesian networks. Perhaps the most common model family used with Bayesian inference is the family of Bayesian networks [64, 66, 107], which

[2] Term "classification" in statistics means usually unsupervised discovery of clusterings of data, not prediction of a discrete value as it is understood in machine learning or computational intelligence communities.

describe the independence assumptions as directed acyclic graphs. In the framework of the theory of Bayesian networks, HAL's finite mixture models can be seen as "one-level Bayesian trees", where the root of the tree represents a latent (hidden) variable, corresponding to the mixing distributions, and the leaves represent the actual random variables of the problem domain (see the discussion in [104]). The theoretical framework in Chapter 2 and 3 can be developed for general Bayesian networks [64], why then does our design for HAL prefer mixture models over the more general Bayesian network family?

Although Bayesian networks are an intuitively appealing model family, in practice the Bayesian approach leads to some serious computational problems when working with general Bayesian network structures. Constructing Bayesian networks from data is computationally a very difficult problem, since the search space, i.e., the number of possible network structures, grows exponentially with respect to the number of variables. In addition, Bayesian deductive inference used for prediction in multi-connected Bayesian networks is an NP-hard problem [30, 34, 120], and hence very probably not computationally feasible. By restricting HAL to finite mixture models we gain the benefit of efficient Bayesian deduction phase, and just have to concern ourselves with the complexity of the Bayesian induction problem. Even in the induction phase HAL does not need to search over the exponentially many Bayesian network structures, since for mixture models we have a fixed model structure. In order to find good models, however, HAL has to search over the missing values of the unobserved latent variable in the dataset. In theory there is an exponential number of possible value combinations for this latent variable, but in HAL's case efficient algorithms for estimating this type of missing data exist.

One of the often advocated advantages of Bayesian networks is the fact that the independence structures can be fixed by human experts, and the Bayesian induction is not necessary. Although admittedly HAL's finite mixtures are semantically not quite as intuitive as Bayesian networks, in many cases the domain experts seem to be able to express their expert knowledge very easily by using prototypical examples or distributions. These examples could then be coded as mixing distributions in HAL's finite mixture framework. In this work we are interested in combination of both Bayesian deductive inference and Bayesian induction, thus this knowledge acquisition aspect is not of importance here. One important related observation is in order. When constructing finite mixture models from data, it should be noticed that neither in the Bayesian nor in any other framework, are finite mixture models always identifiable, i.e., there is not necessarily a unique characterization for a distribution [59, 138]. HAL's focus is in prediction, therefore this non-identifiability is not of any concern for it: if HAL wants to select a single model θ for prediction, it can pick any of these characterizations as they will perform equally well.

Single parametric and non-parametric methods. When compared to single parametric distribution methods and non-parametric methods (such as kernel density estimators [124]), HAL's finite mixture models have many appealing properties. Firstly, in natural domains the distribution to be modeled is often so complicated that probabilistic modeling through single parametric distribution (e.g., normal, multinomial, Poisson [55]) does not lead to good prediction performance. It seems that in real life domain distribution multimodality is more a rule than an exception, at least with small data sets. Finite mixtures have a natural means to model multimodality by placing a component distribution around each mode. Secondly, finite mixtures can model quite complex distributions with few parameters and a high degree of accuracy. Thirdly, many non-parametric approaches depend directly on the sample size. Consequently, if the training sample is large, deductive inference with non-parametric techniques becomes very inefficient in terms of computation time and space. Finite mixture models compress the information present in the data set at two levels: first by grouping similar data vectors to a same cluster, and second by modeling a set of clusters by a single mixing distribution.

4.2.2 Bayesian lazy learning

From Section 3.1.3 we know that the standard approach to machine learning can be viewed as a three phase modeling process. Initially, the models to be considered are restricted to some limited set of models, the *model family*. In the second phase, some specific *model class*, i.e., a skeleton or a template structure for a model, without any parameters, is selected from the chosen model family. In the third phase, the parameter values for the selected model class are estimated from the sample data. The resulting full model (model structure + parameter values) is then used for making predictions.

In contrast to the traditional approach described above, in the *instance-based* (also known as the *memory-based* or the *case-based*) approach [2, 6, 78, 101, 132], the learning algorithms base their predictions directly on the sample data, without producing any explicit (stored) models. This type of machine learning is often referred to as *lazy learning*, since the algorithms defer all the essential computation until the prediction phase [3]. In [143], the lazy learning prediction type of a process is called *transductive inference*.

For making predictions, lazy learning algorithms typically use a distance function (e.g., Euclidean distance) for determining the most relevant data items for the prediction task in question. Some simple function, such as majority voting in classification problems, is then used for determining the prediction from the most relevant data items. It has been shown in various studies (see e.g., [99] for references) that this type of an approach in some cases produces quite accurate predictions, when compared to alternative machine learning methods. The method suffers, however, from several drawbacks when applied in practice (see, e.g., the discussion in [136]). Most importantly, the performance of lazy learning algorithms seems to be highly sensitive to the selection of distance function to be used, as demonstrated in recent work reported in [7, 51].

One possible Bayesian framework for lazy learning using the finite mixture model family has been discussed in [102, 103, 135, 136]. The approach suggested in those studies can be seen as a *partially lazy approach* [3], i.e., a hybrid between the traditional machine learning and the lazy learning approach, which is based solely on the given data. In this probabilistic approach the given data vectors are transformed into local distributions, which can be seen as sample points in a distribution space. Thus the predictive distributions required for making predictions could then be computed by using the lazy learning approach in the distribution space, i.e., by introducing a probabilistic "distance metric". Somewhat similar frameworks have been suggested in [48, 49, 76].

From the design of `HAL` we can see that there is a new, improved probabilistic formalization of the *purely lazy learning approach*, which extends the earlier results by presenting a Bayesian approach for making (discrete) predictions directly from data, *without the transformation step between the original sample space and the distribution space.* Intuitively this new approach is based on the central idea in Bayesian inference: if we wish to make predictions by using only the data given, avoiding the notion of individual models, from the Bayesian point of view we can take this as a requirement for marginalizing, i.e., integrating, out the models. This means that *the predictions will be made by using all the (infinitely many) models*, corresponding to different parameter settings. From this point of view, prediction can be viewed as a missing data problem, where the criterion for filling in the missing data (for making the predictions) is the integral over all the possible models. Therefore using the evidence predictive distribution (4.8) that will be introduced in the next section will actually perform optimal (with respect to the model family $\mathcal{M}_{\mathrm{NB}}$) lazy learning. To avoid terminological confusion it should be observed that even lazy learning is with respect to some model family. The illusion of not having any model family in traditional lazy learning approaches is due to the fact that the model family is *implicitly* induced by the combination of the distance function and the domain of the data, also in the cases where the distance function is allowed to vary locally.

4.3 Predictive inference with finite mixtures

Inference with general finite mixtures

From (3.2) in Section 3.1.3 we know that the correct Bayesian procedure for `HAL` to make predictions would be to use all possible mixture models by weighting them by their posterior density. Unfortunately this is not feasible for the finite mixture model family, so `HAL` needs to resort to approximations with fewer models. In Section 3.1.2 we argued that in many cases it is natural to partition the model family $\mathcal{M}$ to a finite number of model classes, where each model class M consisted of models sharing the same parametric form. In the finite mixture case, a natural choice for determining the model classes is to distinguish different models by the number of the mixing distributions used. In practice the number of model classes, i.e., the maximal number of mixing distributions, can be assumed to be bounded by the size of the data set D. Since `HAL` cannot use all mixture models in $\mathcal{M}$ for prediction, let us now turn to the question whether it can predict with a single model class M. Since we are now in a fixed model family, we will again drop $\mathcal{M}$ from our notation.

Following the discussion in Section 3.1.3 if `HAL` were to use a single model class, it should use the class M with the maximal posterior probability, i.e.,

$$\hat{M} = \arg\max_{M} \frac{p(D|M)p(M)}{p(D)}.$$

Assuming equal priors for the model classes, the posterior probability $P(M_k|D)$ is proportional to the evidence $P(D|M_k)$ (or alternatively viewed, the likelihood of the model class M), hence `HAL` should find the model class with the highest evidence. Assuming maximum a posteriori model class $\hat{M}$ has been found, `HAL` can then use the evidence predictive distribution in Equation (3.11)

$$p_{\text{ev}}(X_m = x_{mi}|(x_1, \ldots, x_{m-1}), D) \propto \int p(\vec{d}[x_{mi}], D|\theta, \hat{M})p(\theta|\hat{M})d\theta.$$

Consequently, if we wish to select the most likely data vector from a small set of alternatives, they can be compared by using the equation (3.11). An example of such task would be a standard classification problem in machine learning.

Unfortunately we will demonstrate in Chapter 5 that this model class based prediction suffers from the problem that calculating the evidences $p(D|M)$ for the general finite mixture family requires computing a sum exponential in the size of the data set D. This exponential sum follows from HAL's incomplete information about the assignments to the clusters (components), which forces HAL to sum over all the possible assignments. Thus in practice HAL has two choices: approximate the actual evidence or use a single model θ from $\hat{M}$ for prediction. Since the notion of evidence is also central for model class selection in Bayesian induction, we defer the derivation of the exact closed form solution in the finite mixture case to Section 5.3, and discuss here the use of a single MAP model for prediction.

In many practical situations HAL may face with a more complex prediction problem, where it wishes to compute the conditional predictive distribution for a set of uninstantiated variables X_i, given that some other variables are instantiated. In the special case of the finite mixture family solving this slightly more general problem is as easy as solving the prediction problem for single X_m discussed in Section 3.2.[3]

Therefore we will discuss HAL's single model prediction in this multiple prediction framework. Let $\mathcal{I} = \{i_1, \ldots, i_t\}$ be the indices of the instantiated variables, and $\mathcal{X} = \{X_{i_s} = x_{i_s l_s}, 1 \leq s \leq t\}$ denote the corresponding assignments. Now we want HAL to determine

$$p(\vec{d}[x_{il}]|D, \mathcal{X}, \hat{M}) = \int p(\vec{d}[x_{il}]|D, \mathcal{X}, \theta, \hat{M}) p(\theta|D, \mathcal{X}, \hat{M}) d\theta, \qquad (4.4)$$

for each value x_{il} of each uninstantiated variable X_i, $i \notin \mathcal{I}$. As observed above, using the evidence predictive distribution for this purpose is infeasible as it requires HAL to sum over all the possible clusterings of the data. On the other hand, if HAL instead of integrating over all the model class parameter settings θ uses a single, maximum a posteriori model $\hat{\theta}$, we can calculate the predictive distribution of X_i directly. For notational convenience, let us first define

$$p_Y(\cdot) \equiv p(Y = y_k|\hat{\theta})$$

and

$$p_{X_{i_s}}(\cdot) \equiv p(X_{i_s} = x_{i_s l_s}|Y = y_k, \hat{\theta}).$$

Since X_i is conditionally independent of M and D, given the maximum a

[3] In fact this result is true also for the general Bayesian network model family, see [66].

posteriori model $\hat{\theta}$, we have

$$p(X_i = x_{il}|D, \hat{\theta}, \mathcal{X}, M) = p(X_i = x_{il}|\hat{\theta}, \mathcal{X}) = \frac{p(X_i = x_{il}, \mathcal{X}|\hat{\theta})}{p(\mathcal{X}|\hat{\theta})}.$$

Using the notation introduced above, we have

$$\begin{aligned}\frac{p(X_i = x_{il}, \mathcal{X}|\hat{\theta})}{p(\mathcal{X}|\hat{\theta})} &= \frac{\sum_{k=1}^{K}\left(p_Y(\cdot)p(X_i = x_{il}|Y = y_k, \hat{\theta})\prod_{s=1}^{t} p_{X_{i_s}}(\cdot)\right)}{\sum_{k=1}^{K}\left(p_Y(\cdot)\prod_{s=1}^{t} p_{X_{i_s}}(\cdot)\right)} \\ &= \frac{\sum_{k=1}^{K}\left(\hat{\alpha}_k\hat{\phi}_{kil}\prod_{s=1}^{t}\hat{\phi}_{ki_sl_s}\right)}{\sum_{k=1}^{K}\left(\hat{\alpha}_k\prod_{s=1}^{t}\hat{\phi}_{ki_sl_s}\right)}. \qquad (4.5)\end{aligned}$$

Here the model class M is assumed to be a class of finite mixtures with K components. Expressing the probabilities $\hat{\alpha}_k$ and $\hat{\phi}_{kil}$ as sufficient statistics of D will be discussed in Section 5.1, when we address the problem of Bayesian induction, i.e., constructing finite mixture models from the data D. The Equation (4.5) offers `HAL` an alternative to (3.9) for calculating the MAP predictive distribution p_{map} when `HAL` is only interested in the predictive distribution of X_m.

As opposed to general Bayesian networks, Bayesian deductive inference with finite mixtures is efficient, since the conditional predictive distribution of X_i can be calculated in time $\mathcal{O}(Ktn_i)$, where K is the number of clusters, t the number of instantiated variables and n_i the number of values of X_i. K is usually small compared to the sample size N, and thus the prediction computation can be performed very efficiently[4].

Inference with Naive Bayes model family

We have already indicated earlier that there exists a trade-off between the complexity of the model family and the degree of model averaging approximations needed. If we restrict `HAL`'s set of possible models enough, we are also able to integrate (marginalize) over models. Consequently, if we restrict `HAL`'s language to be the Naive Bayes model family $\mathcal{M}_{\text{NB}}$, in addition to MAP predictive distributions `HAL` can also compute the more sophisticated evidence (3.11) and stochastic complexity (3.15) predictive distributions exactly *without having to resort to any approximations.* Following the Principle of Indifference we will set all hyperparameters μ_k and σ_{kij} to 1, i.e., `HAL` uses uniform Dirichlet priors for both MAP and evidence prediction. Once the

[4] If massively parallel hardware is available, the computations can be made even faster as the algorithms are easily parallelizable [102, 103].

priors are fixed the predictive distributions (3.9),(3.11), and (3.16) can be written in an explicit form. For notational simplicity, in the following discussion we leave out the the normalizing constant, i.e., the denominator. In addition, to conform to the notation we have adopted for the general finite mixture model family where Y is latent, for the Naive Bayes case we will use index k instead of the actual value x_{mk} for the variable X_m that gives the partitioning of the data (for example $\vec{d}[x_{mk}]$, $\hat{\alpha}_{x_{mk}}$ and n_m will be denoted simply by $\vec{d}[k]$, $\hat{\alpha}_k$ and K, respectively).

We will express the predictive distributions using the *sufficient statistics* [8], which are functions of data which summarize all the available data sample information concerning the model $\boldsymbol{\theta}$. These sufficient statistics of the training data D are h_k and f_{kil}, where h_k is the number of data vectors in class k, and f_{kil} is the number of data vectors in class k with variable X_i having value x_{il}.

The MAP predictive distribution is proportional to the likelihood of the test vector $\vec{d}[k]$:

$$p_{\mathrm{map}}(X_m = k | (x_1, \ldots, x_{m-1}), D) \propto p(\vec{d}[k] | \hat{\theta}) = \hat{\alpha}_k \prod_{i=1}^{m-1} \hat{\phi}_{kix_i}. \qquad (4.6)$$

The mode parameters $\boldsymbol{\theta}_j$ of a Dirichlet distribution $\mathrm{Di}(\mu_1, \ldots, \mu_K)$ are given by (see e.g., [55])

$$\frac{\mu_j - 1}{\mu_0 - K}, \text{ where } \mu_0 \equiv \sum_{k=1}^{K} \mu_k.$$

Now we can express the right hand side of (4.6) with sufficient statistics as

$$\frac{h_k + \mu_k - 1}{N + \sum_{k'=1}^{K} \mu_{k'} - K} \prod_{i=1}^{m-1} \frac{f_{kix_i} + \sigma_{kix_i} - 1}{h_k + \sum_{l=1}^{n_i} \sigma_{kil} - n_i}. \qquad (4.7)$$

The evidence predictive distribution (3.11) is defined as an integral, which in our case can be solved analytically. The derivation is somewhat technical, and uses the results in [29, 66]. The predictive probabilities for p_{ev} can be expressed with sufficient statistics as follows:

$$p_{\mathrm{ev}}(X_m = k | (x_1, \ldots, x_{m-1}), D) \propto \frac{h_k + \mu_k}{N + \sum_{k'=1}^{K} \mu_{k'}} \prod_{i=1}^{m-1} \frac{f_{kix_i} + \sigma_{kix_i}}{h_k + \sum_{l=1}^{n_i} \sigma_{kil}}. \qquad (4.8)$$

The stochastic complexity predictive distribution p_{sc} is proportional to the likelihood of the combined data set $D^+ = D \cup \vec{d}[k]$. Therefore it will be expressed with the sufficient statistics of D^+, h_k^+ and f_{kil}^+, which are defined analogously to h_k and f_{kil}. The derivation is similar to the MAP predictive distribution, thus the have

$$\begin{aligned} p_{\mathrm{sc}}(X_m = k|(x_1, \ldots, x_{m-1}), D) &\propto p(\vec{d}[k], D|\tilde{\theta}(\vec{d}[k], D)) \\ &= \prod_{k'=1}^{K} \left((\tilde{\alpha}_{k'})^{h_{k'}^+} \prod_{i=1}^{m-1} \prod_{l=1}^{n_i} (\tilde{\phi}_{k'il})^{f_{k'il}^+} \right). \end{aligned} \tag{4.9}$$

where

$$\tilde{\alpha}_k = (h_k^+)/(N+1) \quad \text{and} \quad \tilde{\phi}_{kil} = f_{kil}^+/h_k^+.$$

The final expressions for calculating the MAP predictive distribution p_{map}, the evidence predictive distribution p_{ev}, and the stochastic complexity predictive distribution p_{sc} are worthy of a closer inspection. First impression is that they look very similar in form, with only slight differences in the sufficient statistics expressions $\hat{\alpha}_k$ and $\hat{\phi}_{kil}$. In fact for many model families of this type, and for the general Bayesian networks in particular,

$$p_{\mathrm{ev}}(\vec{d}[k], D) = p(\vec{d}[k]|\hat{\theta}(\vec{d}[k], D)) \tag{4.10}$$

i.e., the evidence predictive distribution is the same as the map predictive distribution calculated from the combined data set $D^+ = D \cup \vec{d}[k]$. From the small difference in the sufficient statistics one could then deduce that even for very small data sets there would not be any significant difference. In the empirical comparison of Section 4.4, however, we will see that this is not the case, and that the above reasoning has overlooked two things. First, the parameters $\hat{\alpha}_k$ and $\hat{\phi}_{kil}$ are multiplied with each other, which amplifies the effect for even small additive changes. However, more importantly, for a fixed k', $1 \leq k' \leq K$ the probability

$$p_{\mathrm{ev}}(\vec{d}[k'], D) = p(\vec{d}[k']|\hat{\theta}(\vec{d}[k'], D)),$$

i.e., the maximum a posterior model $\hat{\theta}(\vec{d}[k'], D)$ is different for each k' as opposed to p_{map}, where all the predictions $p(\vec{d}[k']|\hat{\theta}(D))$ use the same $\hat{\theta}(D)$. From the observation (4.10) and the Equation (3.16) we can also see the connection between p_{ev} and p_{sc} for uniform priors: p_{sc} is p_{ev} "corrected" by the factor $p(D|\tilde{\theta}(D^+))$, i.e., how likely the training data are assuming the ML model of the combined data D^+.

4.4 Comparing predictive inference methods

We have discussed the three alternative predictive distributions $p_{\text{map}}, p_{\text{ev}}$, and p_{sc} that HAL could use for approximating the predictive distribution $p(\cdot)$ presented in Section 3.1.1. Each of them exploits a different joint probability distribution for the variables, but they are all defined with respect to the same parametric model family $\mathcal{M}$. Prediction with p_{map} is the traditional approach based on the single most probable model. Using the evidence predictive distribution p_{ev} is the Bayes optimal approach with respect to a given model class M, and p_{sc} is optimal in the shortest possible code-length sense [117]. An interesting question then is, given a choice, which predictive distribution should HAL use, or is there any difference at all in practice?

Computing the p_{map} or p_{sc} predictive distribution, or at least a good approximation of them, is often feasible in practice, but the evidence approach requires integrating over M, which is infeasible for many commonly used model families. With the Naive Bayes model family we have an exceptional situation, where all three predictive distributions can be represented in a form allowing a computationally efficient implementation, as demonstrated in Section 4.3. We will now proceed by evaluating and comparing the predictive accuracy of the three predictive inference methods empirically by using publicly available natural classification data sets. We are not aware of any earlier comparisons of this nature, where all these prediction methods are compared together (in fact we are not aware of any studies, where the recent new version of stochastic complexity has been applied in natural domains).

Our interest here is to get information on the general behavior of the prediction methods. However, we would like to point out that, despite the simplicity of the parametric form, the family of Naive Bayes models is widely used in practice [43, 39, 82, 99], and thus the results have relevance also from a practitioner's point of view.

In our experiments five public domain classification data sets from the UCI data repository[5] were used: Australian, Hepatitis, Glass, Primary Tumor and Heart Disease (for the description of the data sets see Appendix A). These data sets were chosen to represent samples with different attribute number and size combinations. The same data sets, among others, are later on also used for the classification experiments in Chapter 6.

For each data set two separate sets of experiments were performed. In the first set of experiments we explored HAL's learning rate with the altern-

[5] The data sets can be obtained from the UCI data repository at URL "http://www.ics.uci.edu/~mlearn/".

ative approaches, i.e., how the prediction quality of the different predictive distributions depends on the size of the training set D. In the second set of experiments we make a more detailed investigation on the differences in prediction quality between the three approaches for a fixed training set size.

Prediction performance and the training set size

In this set of experiments we randomly partitioned each data set in a *training reservoir* D_r containing 70% of the data instantiations and a *test set* D_q containing the remaining 30%.

One data instantiation $\vec{d}_1$ was then randomly taken out of the training reservoir and used as HAL's training set $D_1 = \{\vec{d}_1\}$. This initial training set D_1 was used to generate the predictive distributions

$$\begin{aligned} p_{\text{map}}(X_m &= k|(x_1, \ldots, x_{m-1}), D), \\ p_{\text{ev}}(X_m &= k|(x_1, \ldots, x_{m-1}), D), \\ p_{\text{sc}}(X_m &= k|(x_1, \ldots, x_{m-1}), D) \end{aligned}$$

for all $\vec{d}_j \in D_q$, and the predictions thus obtained for each $\vec{d}$ were compared to the actual outcomes k in a manner to be described below.

Next the training set D_1 was extended by another data instantiation $\vec{d}_2$, unequal to the element already in D_1 but otherwise randomly picked from the training reservoir D_r. This new training set is denoted by D_2. For all $\vec{d}_j \in D_q$ the three predictive distributions were determined and again, the predictions thus obtained were compared to the actual outcomes k. This procedure of adding one training element to D_i to form D_{i+1}, determining all predictive distributions using D_{i+1} and predicting the value of X_m for each entry in the test set was repeated until $D_{i+1} = D_r$, i.e., contained the full training reservoir.

The 0/1-score results lose quite a lot of information about the predicted distributions, as we can only see whether or not the mode of the predicted probability distribution coincides with the correct class k, but not what the predicted probability was. In order to be able to get more detailed information about the predictive distributions each prediction was evaluated by both log-score and 0/1-score. The *log-score* of a predictive distribution $p(X_m|(x_1, \ldots, x_{m-1}), D)$ is defined as

$$-\log p(X_m = k|(x_1, \ldots, x_{m-1}), D_i),$$

where k is the actual outcome of X_m.[6] For 0/1-score we simply determine the k for which $p(X_m = k|(x_1, \ldots, x_{m-1}), D)$ is the maximum, and then

[6] As we used logarithm of base two, the log-score has also a coding interpretation: If

predict X_m to take the value k. If the actual outcome is indeed k, then the 0/1-score is defined to be 1; if it is not equal to k, the 0/1-score is defined to be 0.

This whole procedure of partitioning the data set and successively predicting using larger and larger subsets of the training reservoir as our training data was repeated 100 times. In Figure 4.1 the performance of the three predictive distributions on the Glass data set is shown for both log-score and 0/1-score. The vertical axis indicates the average score where the average is taken over the predictions of all class values in the test set and all the 100 training sets of the size indicated on the horizontal axis.

We can see that both scores rapidly increase as the size of the training set increases in all three approaches. For these experiments the final level of accuracy achieved (when full training reservoir is used) is quite similar, no meaningful differences can be seen. Let us now focus on the interesting question, what happens in HAL's learning with small training set sizes.

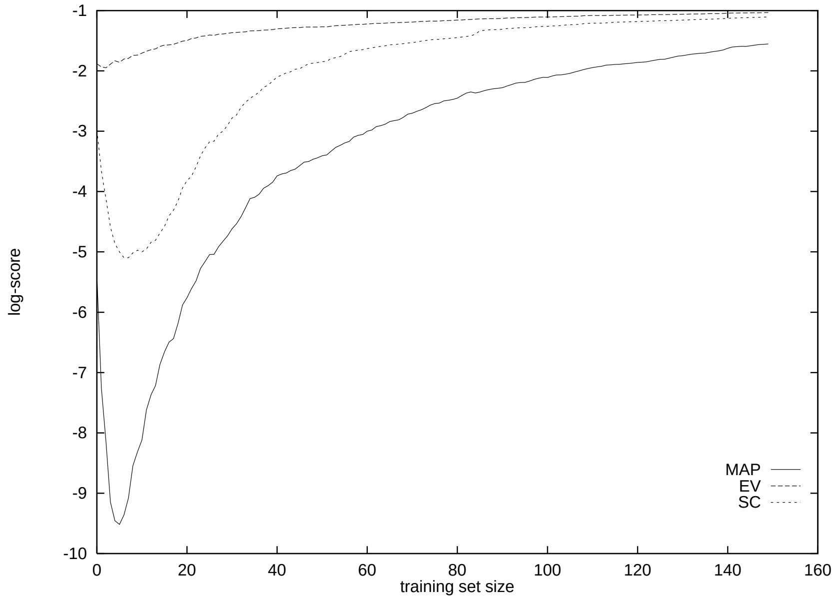
For the log-score, p_{ev} performs already well with training set sizes of 20 (i.e., 10 % of the full data set), while both the p_{map} and the p_{sc} predictions show weak performance. For the 0/1-score, the stochastic complexity predictive distribution shows the same behavior, while the p_{map} predictions tend to behave in a manner more similar to the p_{ev}. These results with the Glass data set are quite representative—analogous, though sometimes less extreme, behavior was found for all of the five data sets used. Additional examples are shown in Figure 4.2, where the graphs for the Australian and Hepatitis data sets are presented (for the log-score). Hepatitis presents an extreme case for p_{ev}, where on the average three data vectors is enough to achieve essentially as good performance as if the full training reservoir were used. Corresponding figures for all the data sets appear in Figures B.3–B.2 in Appendix B.

Figures 4.1 and 4.2 describe average behavior over many training sets; this raises the question of how well the methods perform for individual training sets. In Figure 4.3 the log-score performance averages for p_{ev} and p_{sc} are shown together with the maximum and the minimum prediction performance for the Glass data set. For each training set size N, the maximum (minimum) performance is defined to be the prediction performance of the one training set out of the 100 training sets of size N that had the best (worst) performance on the test set. We see that after HAL has seen about 20 data

one encodes the data using the code corresponding to $p(X_m|(x_1,\ldots,x_{m-1}),D)$, then

$$-\log p(X_m = k|(x_1,\ldots,x_{m-1}),D)$$

is equal to the number of bits one needs to describe the actual outcome k.



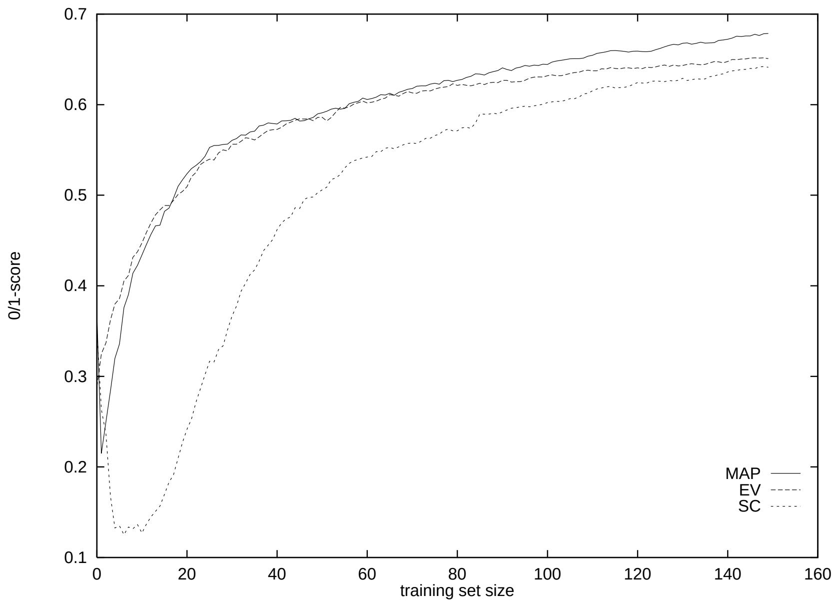
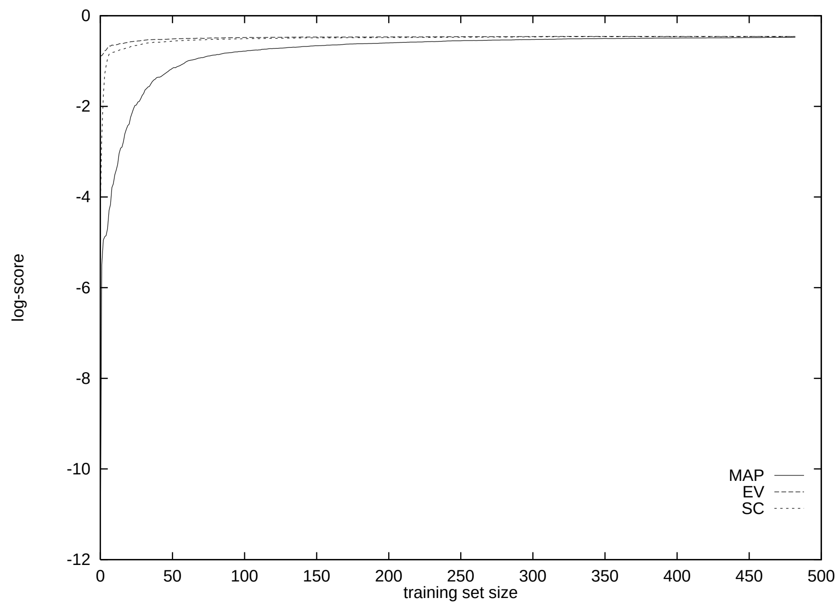


Figure 4.1: Average performance of the methods by log-score (up) and 0/1-score (below) on the Glass data set.



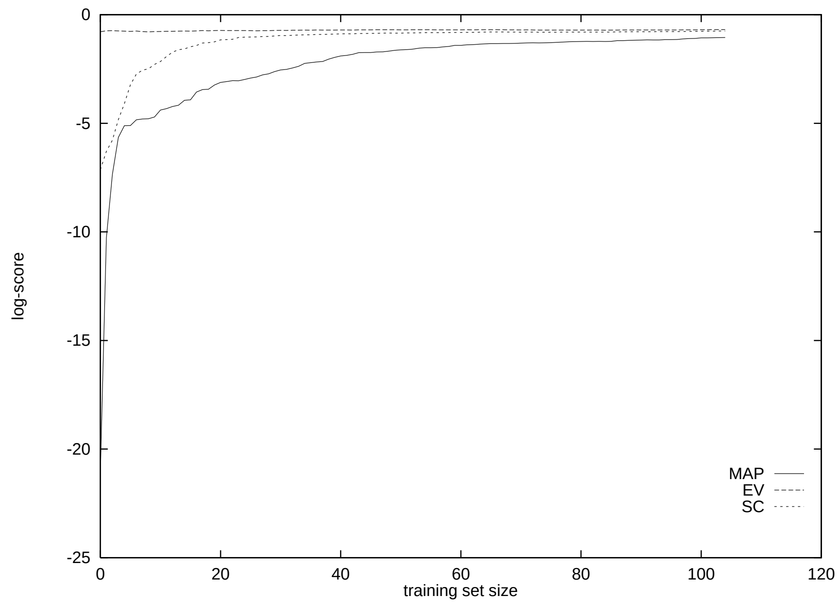


Figure 4.2: Average performance of methods by log-score on the Australian (up) and Hepatitis (below) data sets.

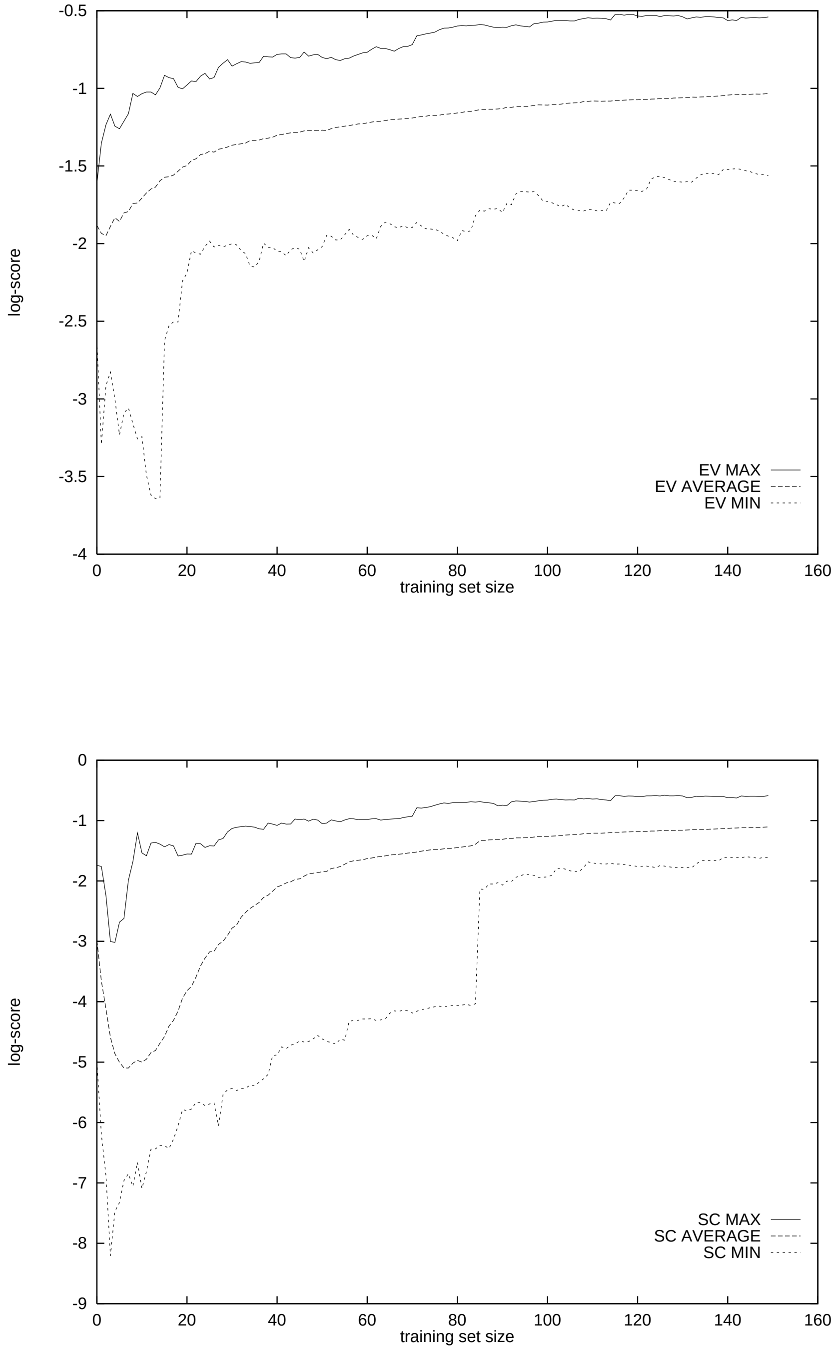


Figure 4.3: Performance of evidence (up) and stochastic complexity (below) by log-score on the Glass data set.

vectors (about 10% of the data), the worst case p_{ev} prediction suddenly goes up. This "phase transition" behavior also occurs for the stochastic complexity predictive distribution p_{sc}, but only after about 80 data items. The corresponding graphs for the other data sets used and for the 0/1-score show similar tendencies (Figures B.12–B.11 in Appendix B). This observation is, however, more a property of the particular sample than the "learning behavior"—the "phase transition" moves to larger training set sizes if the number of training sets is increased.

From this set of experiments one can conclude two things. First, some of these commonly used data sets are quite redundant, and when properly used, only a very small sample of these data sets is needed to construct good models. Second, at least for the Naive Bayes model family, p_{ev} with uniform priors can be an extremely "safe" choice for `HAL`'s predictive distribution: even for small sample sizes it predicts well in most cases.

If `HAL` uses evidence predictive distribution p_{ev}, in the extreme cases it would need only few data items to predict well, and for almost every case less than 20% of the training data would be enough to achieve good predictive performance. If `HAL` used p_{map} prediction for the same data, it would require much more data vectors to "catch up" the p_{ev} performance. The predictions with stochastic complexity predictive distribution p_{sc} for the log-score are in most cases in between these two extremes. How can this be explained?

If one looks at the actual predictions made, the p_{ev} prediction is much more "conservative" than the p_{map} prediction. The latter is in our case equal to using the maximum likelihood predictive distribution, and it is a well-known fact that, for small sample sizes, the ML predictor is too dependent on the observed data and does not take into account that future data *may* turn out to be different. A very simple example illustrates this point. Suppose our data consists of a string of ones and zeros generated by some Bernoulli-process. If we have seen an initial string consisting of just one '1', then the p_{map} prediction will determine that the probability of the second symbol being a 1 is unity. Using the p_{ev} prediction with uniform priors, however, this probability is $\frac{2}{3}$. If the next data item turns out to be a 0, then the log-score of the p_{map} prediction will be $-\infty$ while that of the evidence predictive distribution p_{ev} will be $\log 2 - \log 3$.

The behavior of the stochastic complexity predictive distribution p_{sc} lies somewhere in between that of p_{map} and that of the p_{ev}. In our example, the probability of the second symbol being a 1 will be $\frac{4}{5}$: the stochastic complexity predictive distribution is less conservative than the evidence predictive distribution, but still more conservative than the p_{map}, which explains part of the small sample size behavior for the log-score.

Data set	MAP-01	EV-01	SC-01	MAP-LS	EV-LS	SC-LS
Australian	**0.851**	0.848	0.848	**-0.456**	-0.457	-0.458
Primary Tumor	0.460	**0.490**	0.434	-3.247	**-1.930**	-2.112
Heart Disease	0.830	**0.837**	**0.837**	-0.476	**-0.439**	-0.444
Glass	**0.701**	0.668	0.636	-1.216	**-0.981**	-1.015
Hepatitis	**0.847**	0.820	0.827	-0.853	**-0.666**	-0.692

Table 4.1: Leave-one-out cross-validation results on the five data sets used.

Predictive performance with fixed training set size

In our second set of experiments, we tested HAL's three prediction methods using leave-one-out cross-validation [133], both for the log-score and the 0/1-score. As the training sets in leave-one-out cross-validation are almost as large as the full data sets, Figures 4.1-4.3 already suggest that the methods will show quite similar performance. Our results on the five data sets are summarized in Table 4.1. The middle columns show the cross-validated results for the 0/1-score, the three rightmost columns show the results for the log-score. Though the differences in performances are all quite small, we see that for the log-score, the evidence predictive distribution prediction p_{ev} performs consistently better than the stochastic complexity predictive distribution, which itself outperforms the use of p_{map} predictions. For the 0/1-score, the picture is not as clear-cut, but it seems that the p_{map} prediction performs slightly better than both the evidence and stochastic complexity distribution predictions. This is not particularly surprising since 0/1-score is very coarse, and it is not important what actual probability HAL attaches to a class value being k; all probability distributions over the class values for which k gets the maximum probability will lead to the same prediction. Thus it can very well happen that, while the p_{map} captures less well the regularities underlying the data (and hence performs worse with respect to log-score), it still captures them well enough to give maximum probability to the class value that should indeed receive maximum probability.

This comparison was performed using $\mathcal{M}_{\mathrm{NB}}$, i.e., in the complete data case. For the general finite mixture models the class variable X_m is assumed to be a latent variable, thus the three predictive distributions described here can only be solved analytically by summing over all the possible instantiations of the missing data, which are exponential in number. Analogously to the comparison performed here we could also compare the methods for general finite mixtures, but in such a case one would have to consider also the effect of the approximations (see Chapter 5). Many of these approximations are asymptotic, and thus would render the type of study about learning rate performed here meaningless.

Chapter 5

Constructing finite mixtures

"Too many machine learning programs suffer from an extreme case of this deficiency [requirement for parameter tuning], which is named the "China Syndrome" because sometimes the only person who is able to make a program run is in China."

—Buchanan, 1987 talk paraphrased in
Megainduction: machine learning on very large databases

We have now completed the construction of `HAL`'s machinery for predicting with Bayesian deductive inference in the case of finite mixture model family. In order to be able to use its prediction formulae, however, `HAL` needs either the maximum a posteriori model class $\hat{M}$, or the maximum a posterior model $\hat{\theta}$ in $\hat{M}$. Unfortunately, for the finite mixture family both calculating the maximum of the posterior $p(\theta|M)$ for models in a fixed model class, and the posterior $p(M|D)$ for the model classes involves computation exponential in the amount of data. Consequently for any realistic data set `HAL` has to use approximations to the actual maximum a posteriori model or model class.

This chapter is concerned with Bayesian inductive inference, which in `HAL`'s case reduces to finding high posterior finite mixture models or model classes, to be used for its predictions. At this point we will see the advantage of being able to separate the criterion for good models (model classes) and the search process: we can discuss the problem of computing the posterior probability of different models θ, given a model class M (Section 5.1), without being forced to fix any particular search algorithm. Section 5.2 then discusses how to use a variant of the Expectation-Maximization (EM) algorithm to search over the model class for models with high posterior probability, but as shown in [84], other alternatives exist.

It has become apparent that the model class evidence $p(D|M)$ has a very central role in HAL's prediction process. We thus discuss first the exact form of the model class evidence for finite mixtures (Section 5.3), and then study the performance of some efficient approximations to it (Section 5.4). We conclude the chapter by summarizing the Bayesian model construction procedure (Section 5.5).

5.1 Selecting finite mixture models

Let us now turn to the problem of computing the posterior of the models θ in a given model class M. Use of Bayes' theorem gives us the posterior probability of a model θ as (Equation (3.5))

$$p(\theta|D, M) \propto p(D|\theta, M)p(\theta|M). \tag{5.1}$$

From the discussion in Chapter 4 we know that a finite mixture model θ in model class M_K assumes that the data is sampled from K distributions, i.e., the finite mixture model θ imposes a partitioning of the data into K clusters. This partitioning can be modeled by introducing an unobserved latent variable Z_j for each data vector $\vec{d}_j$, the value of which gives the cluster index of the cluster vector $\vec{d}_j$ belongs to. We can now think a vector

$$Z = (z_1, \ldots, z_N)$$

consisting of the values of the latent variables $Z_1, \ldots, Z_N$, as a random sample from the distribution of Y like D is a random sample from the joint distribution of $X_1, \ldots, X_m$. However, for technical reasons it is more convenient to consider each value z_j as a vector of *cluster indicator variable* values, $z_j = (z_{j1}, \ldots, z_{jK})$, where

$$z_{jk} = \begin{cases} 1, & \text{if } \vec{d}_j \text{ is sampled from } p(\cdot|Y = y_k), \\ 0, & \text{otherwise.} \end{cases}$$

The vector Z is unknown, thus in order to calculate the *incomplete data likelihood* $p(D|\theta)$ in (5.1), we need to marginalize it out. Therefore

$$p(D|\theta) = \sum_Z p(D, Z|\theta),$$

where the summation is over all the possible K^N values of the vector Z. The joint likelihood terms $p(D, Z|\theta)$ in the sum are called *complete data likelihoods*. Let us introduce an indicator variable v_{jil} to simplify the form of the complete data likelihood definition:

$$v_{jil} = \begin{cases} 1, & \text{if } d_{ji} = x_{il}, \\ 0, & \text{otherwise.} \end{cases}$$

Here we have adopted the notation that d_{ij} is the value of the variable X_i of data vector $\vec{d}_j$.

We have assumed that the data are i.i.d., thus the complete data likelihood $p(D,Z|\theta)$ can be written as a product

$$p(D,Z|\theta) = \prod_{j=1}^{N}\prod_{k=1}^{K}\left(\alpha_k \prod_{i=1}^{m}\prod_{l=1}^{n_i}\phi_{kil}^{v_{jil}}\right)^{z_{jk}}. \tag{5.2}$$

The incomplete data likelihood can be expressed as marginalized complete data likelihood

$$p(D|\theta) = \sum_Z p(D,Z|\theta) = \sum_Z \left(\prod_{j=1}^{N}\prod_{k=1}^{K}\left(\alpha_k \prod_{i=1}^{m}\prod_{l=1}^{n_i}\phi_{kil}^{v_{jil}}\right)^{z_{jk}}\right). \tag{5.3}$$

This sum has exponential number of terms, where each term represents one of the possible clusterings of the data. This is the reason for the difficulty of finding the MAP finite mixture model: no computationally feasible solution for maximizing the incomplete data posterior (5.1) is known.

On the other hand we saw already above that maximizing the complete data likelihood does not involve any exponential sum anymore, as the cluster assignments Z are given. Since this indicates that calculating the complete data posterior is feasible, we can approach the incomplete posterior maximization indirectly as follows.

Here we again choose Dirichlet prior distribution for the parameters (see the discussion in Section 4.1), i.e.,

$$(\alpha,\phi) \sim \mathrm{Di}\left(\mu_1,\ldots,\mu_K\right)\prod_{k=1}^{K}\prod_{i=1}^{m}\mathrm{Di}\left(\sigma_{ki1},\ldots,\sigma_{kin_i}\right). \tag{5.4}$$

Consequently the joint density $p(\theta)$ can be written as

$$p(\theta) = \Gamma\left(\sum_{k=1}^{K}\mu_k\right)\prod_{k=1}^{K}\left(\frac{\alpha_k^{\mu_k-1}}{\Gamma(\mu_k)}\right)\prod_{k=1}^{K}\prod_{i=1}^{m}\left(\Gamma\left(\sum_{l=1}^{n_i}\sigma_{kil}\right)\prod_{l=1}^{n_i}\frac{\phi_{kil}^{\sigma_{kil}-1}}{\Gamma(\sigma_{kil})}\right). \tag{5.5}$$

We will use use the sufficient statistics introduced already in Chapter 4: h_k, which is the number of instantiations in cluster k, and f_{kil} denoting the number of instantiations in cluster k with variable X_i having value x_{il}. These sufficient statistics can now be expressed using the indicator variables z_{jk} and v_{jil}:

$$h_k = \sum_{j=1}^{N} z_{jk} \quad\text{and}\quad f_{kil} = \sum_{j=1}^{N} z_{jk}v_{jil}.$$

Since we have chosen a conjugate prior, it follows that the posterior distribution is also a product of Dirichlet distributions and is given by

$$\mathrm{Di}\left(\mu_1 + h_1, \ldots, \mu_K + h_K\right) \prod_{k=1}^{K} \prod_{i=1}^{m} \mathrm{Di}\left(\sigma_{ki1} + f_{ki1}, \ldots, \sigma_{kin_i} + f_{kin_i}\right).$$

Therefore the complete data posterior density $p(\theta|D,Z)$ can be expressed

$$\begin{aligned} p(\theta|D,Z) = & \, \Gamma\left(\sum_{k=1}^{K}(\mu_k + h_k)\right) \prod_{k=1}^{K} \left(\frac{\alpha_k^{\mu_k + h_k - 1}}{\Gamma(\mu_k + h_k)}\right) \\ & \cdot \prod_{k=1}^{K} \prod_{i=1}^{m} \left(\Gamma\left(\sum_{l=1}^{n_i}(\sigma_{kil} + f_{kil})\right) \prod_{l=1}^{n_i} \frac{\phi_{kil}^{\sigma_{kil} + f_{kil} - 1}}{\Gamma(\sigma_{kil} + f_{kil})}\right). \end{aligned} \tag{5.6}$$

The parameters α_k and ϕ_{kil} appear disjointly in (5.6), and hence can be maximized separately. We can again use the information about the mode of the Dirichlet density (see Section 4.3) to find the MAP values of the complete data posterior:

$$\alpha_k = \frac{h_k + \mu_k - 1}{N + \sum_{k'=1}^{K} \mu_{k'} - K}, \quad \phi_{kil} = \frac{f_{kil} + \sigma_{kil} - 1}{h_k + \sum_{l'=1}^{n_i} \sigma_{kil'} - n_i}. \tag{5.7}$$

By the solution given in (5.7) we have actually solved `HAL`'s problem of finding the MAP model for the Naive Bayes model family $\mathcal{M}_{\mathrm{NB}}$, since the cluster assignments Z are already given by the class variable X_m. Thus the MAP parameter values $\hat{\alpha}_k$ and $\hat{\phi}_{kil}$ in (4.7) are the maximums of the complete data posterior $p(\theta|D)$. Consequently, for the model family $\mathcal{M}_{\mathrm{NB}}$ `HAL` can calculate the MAP model directly from the training data *without any search process.*

Here we can make an important observation: all `HAL`'s computational difficulties in predicting with the finite mixture model family result from the uncertainty about the clustering of the data. On the other hand, the complete cases of the posterior and evidence can be solved efficiently. Many of the approximations for the incomplete cases try to make use of this fact. In particular, we will next discuss a method of finding the local maximum of the incomplete data posterior by maximizing the expectation of the complete data posterior.

5.2 Searching models with EM

We have seen that finding the MAP parameter values θ for the incomplete data posterior is not feasible in practice. The problem could, however, be solved by augmenting the observed data D with the missing cluster assignments Z. This is a typical *missing data problem* [93], which has been addressed in several different ways. One of the most commonly used generic missing data algorithms is the *Expectation-Maximization (EM)* algorithm [38]. In the following, we will apply EM in our case to find good approximations to the MAP parameters of the incomplete data posterior $p(D|\theta)$.

The EM algorithm is an iterative procedure consisting of two steps: Expectation (E)-step and Maximization (M)-step. In the E-step, the expected complete data posterior given the incomplete data and the current estimated parameter values is determined. In the M-step, the parameter values are updated in such a way that the obtained expected posterior is maximized. The underlying intuition is that we would like to maximize the incomplete data posterior but since it cannot be done, we maximize the expectation of the complete data posterior instead.

The E-step in Algorithm 5.2.1 requires the evaluation of the function Q, which is an expectation over missing data. In our case the missing data is the cluster assignment vector Z, which has K^N possible values. It follows that the exact determination of Q is again computationally infeasible. Fortunately, in our case there is a standard way to overcome this problem. We can apply the Bayes' theorem sequentially considering only one data vector at a time, and then approximating the resulting posterior in a suitable way. This procedure is called the Quasi-Bayes algorithm [138] or fractional updating [137]. In the Quasi-Bayes algorithm, instead of taking the expectation of the whole posterior, the missing data is simply replaced by its expectation. This gives us an approximation to the function Q.

In order to define the expectations of the sufficient statistics $\bar{h}_k$ and $\bar{f}_{kil}$ we need the expectations of the cluster indicators z_{jk}:

$$w_{jk} = E[z_{jk}|D, \theta^{(t)}] = p(z_{jk} = 1|\vec{d}_j, \theta^{(t)}),$$

since the data are i.i.d. From Bayes' theorem we now have

$$\begin{aligned} w_{jk} &= \frac{p(z_{jk} = 1|\theta^{(t)})p(\vec{d}_j|z_{jk} = 1, \theta^{(t)})}{p(\vec{d}_j|\theta^{(t)})} \\ &= \frac{\alpha_k^{(t)} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \left(\phi_{kil}^{(t)}\right)^{v_{jil}}}{\sum_{k'=1}^{K} \left(\alpha_{k'}^{(t)} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \left(\phi_{k'il}^{(t)}\right)^{v_{jil}}\right)}. \end{aligned} \tag{5.8}$$

Algorithm 5.2.1
General Expectation-Maximization algorithm (EM)

i. Initialize parameters randomly. Set $t = 0$.

ii. ITERATE until selected convergence criteria are satisfied:

(a) E-step: Determine $Q(\theta, \theta^{(t)}) = E[\log p(\theta|D, Z)|D, \theta^{(t)}]$.

(b) M-step: Set $\theta^{(t+1)} = \arg\max_{\theta}\{Q(\theta, \theta^{(t)}) \mid \theta \in M\}$, where $\theta^{(t)}$ are the parameter estimates in time step t.

(c) Set $t = t + 1$.

iii. Return $\theta^{(t)}$.

Using (5.8) the expectations of the sufficient statistics can be expressed as

$$\begin{aligned}\bar{h}_k &= E[h_k|D, \theta^{(t)}] = \sum_{j=1}^{N} w_{jk}, \\ \bar{f}_{kil} &= E[f_{kil}|D, \theta^{(t)}] = \sum_{j=1}^{N} w_{jk} v_{jil}.\end{aligned}$$

Now the approximation to the function Q is given by

$$\begin{aligned}\hat{Q}(\theta, \theta^{(t)}) &= \mathrm{Di}\,(\mu_1 + \bar{h}_1, \ldots, \mu_K + \bar{h}_k) \\ &\cdot \prod_{k=1}^{K} \prod_{i=1}^{m} \mathrm{Di}\,(\sigma_{ki1} + \bar{f}_{ki1}, \ldots, \sigma_{kin_i} + \bar{f}_{kin_i}) \,. \end{aligned} \tag{5.9}$$

This approximation can be intuitively explained as follows. By replacing unknown z_{jk}'s by their expectations, each data vector is divided between the clusters based on the probabilities that it originates from these clusters. Summing of these probabilities is justified by the incremental nature of the Dirichlet distribution[1]. An analysis of the convergence of the Quasi-Bayes procedure and a comparison to other similar methods is given in [138].

In the M-step, we can now make use of the fact that computing the complete data posterior is feasible. The functional form of (5.9) is the same as

[1] By incrementality we mean here the property that when a new data vector arrives, the parameters of Dirichlet distribution are updated by adding the contribution of the new data vector to the old value of each parameter.

Algorithm 5.2.2
Expectation-Maximization algorithm for incomplete data posterior

i. Initialize parameters randomly. Set $t = 0$.

ii. ITERATE until selected convergence criteria are satisfied:

(a) E-step: for each j, k compute

$$w_{jk} = E[z_{jk}|D, \theta^{(t)}] = \frac{\alpha_k^{(t)} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \left(\phi_{kil}^{(t)}\right)^{v_{jil}}}{\sum_{k'=1}^{K} \left(\alpha_{k'}^{(t)} \prod_{i=1}^{m} \prod_{l=1}^{n_i} \left(\phi_{k'il}^{(t)}\right)^{v_{jil}}\right)}.$$

(b) M-step: Update parameters as

$$\alpha_k^{(t+1)} = \frac{\sum_{j=1}^{N} w_{jk} + \mu_k - 1}{N + \sum_{k=1}^{K} \mu_k - K},$$

$$\phi_{kil}^{(t+1)} = \frac{\sum_{j=1}^{N} w_{jk} v_{jil} + \sigma_{kil} - 1}{\sum_{j=1}^{N} w_{jk} + \sum_{l=1}^{n_i} \sigma_{kil} - n_i}.$$

(c) Set $t = t + 1$.

iii. Return $\alpha_k^{(t)}$ and $\phi_{kil}^{(t)}$.

the complete data posterior (5.6), therefore the same maximization formulas (5.7) apply. Thus $\theta^{(t)}$ that maximizes the Equation (5.9) can be found efficiently. Algorithm 5.2.2 presents the (Quasi-Bayes) EM algorithm for our finite mixture case. In principle the EM algorithm can be shown to have linear convergence [38], i.e., near the mode $\theta = \hat{\theta}$ of the posterior,

$$\| \theta^{(t+1)} - \hat{\theta} \| = \lambda \| \hat{\theta}^{(t)} - \hat{\theta} \|,$$

where $\lambda \in \mathbb{R}, \lambda < 1$ is the *convergence rate* of EM. The convergence of EM is monotonic and the algorithm is assured to converge to a local optimum. Although the two main competitors of EM, the Method of Scoring and Newton-Raphson (see e.g., [138]), have a quadratic convergence rate, the methods lack the nice monotonic convergence property (and may not converge at all), and are also usually more difficult to apply than EM. In practice, at least in all the experiments related to the work here, a relatively small number of iterations is sufficient for finding good models θ in terms

	cl	weight	small	big	red	green	blue
Random	1	0.754348	0.236773	0.763227	0.203462	0.278712	0.517826
	2	0.245652	0.484881	0.515119	0.468076	0.358158	0.173766
Round 1	1	0.727074	0.181035	0.818965	0.180993	0.178833	0.640174
	2	0.272926	0.739309	0.260691	0.738858	0.134891	0.126251
Round 2	1	0.687241	0.067032	0.932968	0.067016	0.222526	0.710458
	2	0.312759	0.918675	0.081325	0.918186	0.044528	0.037286
Round 3	1	0.668115	0.006017	0.993983	0.006015	0.247537	0.746448
	2	0.331885	0.992419	0.007581	0.991921	0.004451	0.003627

Table 5.1: The values of the parameters (probabilities) initially and after each iteration of the Algorithm 5.2.2.

of the prediction error (see also the discussion in [84]). This allows `HAL` to search for good approximations of the global optimum $\hat{\theta}$ by iteratively running algorithm 5.2.2 with randomly chosen initializations of the parameters as many times as possible within a given time limit.

We conclude our discussion on model selection by illustrating the behavior of Algorithm 5.2.2 with a small example. Let us assume that `HAL`'s observable domain consists of balls with two attributes: `color = {red,green,blue}` and `size = {big,small}`. We thus have two variables $X_{\texttt{color}}$ and $X_{\texttt{size}}$. Furthermore assume that we want to find a model $\theta \in M_2$, i.e., a two-cluster model. In this case the model is

$$\theta = (\alpha_1, \alpha_2, \phi_{111}, \phi_{112}, \ldots, \phi_{222}).$$

The data set D used for model construction is

$$D = \{(\texttt{small}, \texttt{red}), (\texttt{small}, \texttt{red}), (\texttt{big}, \texttt{blue}), \\ (\texttt{big}, \texttt{blue}), (\texttt{big}, \texttt{blue}), (\texttt{big}, \texttt{green})\}.$$

The convergence criteria for the EM is set to 3 iterations. Table 5.1 gives the model parameter values, i.e., the probability estimates θ_i, in the initial random state and after each iteration of the algorithm. The same distribution information is presented graphically in Figure 5.1. As we can see, starting from a random initial assignment, EM converges towards a model θ^b with clusters "`big` and `blue`" (Cluster 1) and "`small` and `red`" (Cluster 2). `HAL` can now represent the queries "If I observe that the ball is green, what is its size?" and "If I observe that the ball is big, what is its color?" as new data vectors d_1 and d_2 with missing values, i.e.,

$$(*, \texttt{green})? \quad \text{and} \quad (\texttt{big}, *)?$$

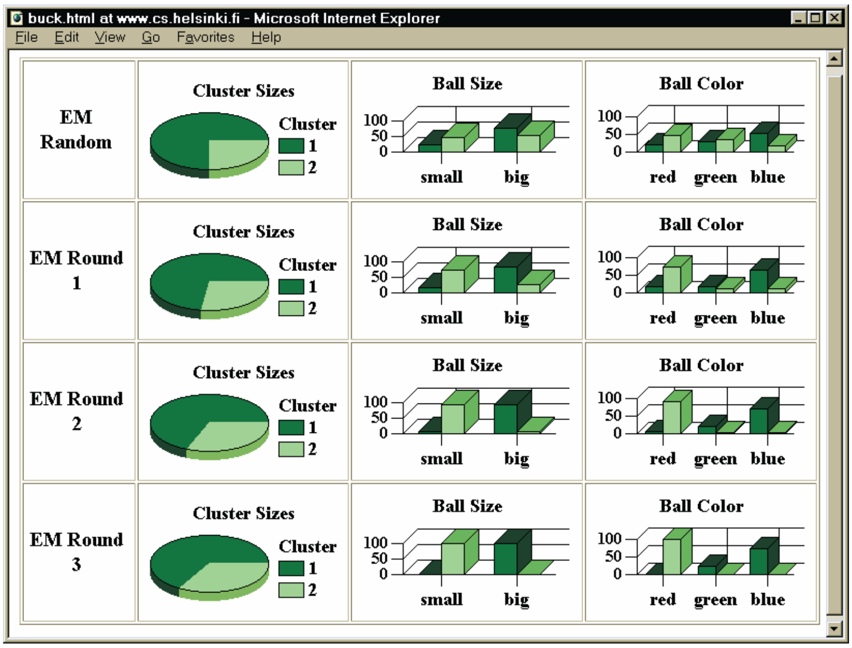


Figure 5.1: The parameter distributions initially and after each iteration of the Algorithm 5.2.2.

Then the predictions with the model θ^b using the MAP prediction are

$$\begin{aligned} p(\texttt{small}|\texttt{green}, \theta) &= 0.015, \\ p(\texttt{big}|\texttt{green}, \theta) &= 0.985, \\ p(\texttt{red}|\texttt{big}, \theta) &= 0.010, \\ p(\texttt{green}|\texttt{big}, \theta) &= 0.247, \\ p(\texttt{blue}|\texttt{big}, \theta) &= 0.743. \end{aligned} \tag{5.10}$$

The same predictive distributions are presented graphically in Figure 5.2.

5.3 Selecting finite mixture model classes

We have seen that in the Bayesian framework one of the terms in the Bayes' theorem, the evidence $p(D|M)$, has a very central role: the posterior probability for each fixed model class $p(M|D)$ is directly proportional to the

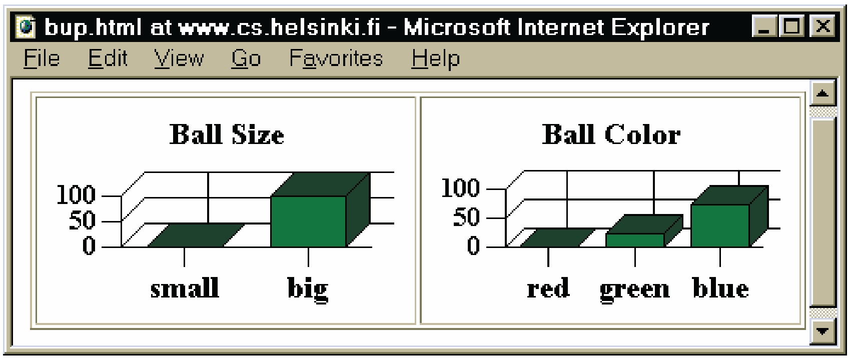


Figure 5.2: The predictive distributions to the queries ($*$,`green`) and (`big`,$*$).

evidence, with uniform priors $p(M)$ it gives `HAL` the optimal number of mixing distributions (clusters), and by using (3.11) `HAL` can use the evidence directly for predictions. We now turn to the problem of computing the evidence

$$p(D|M) = \int p(D|\theta, M)p(\theta|M)d\theta \tag{5.11}$$

for a fixed model class M, and consequently drop M from the conditioning parts of our expressions.

As discussed earlier, each instantiation of Z represents a cluster assignment of all the N data vectors to the K possible classes. The *incomplete data evidence* (5.11) can now be computed by marginalizing, i.e., summing over the K^N *complete data evidences*, i.e.,

$$p(D) = \sum_Z p(D, Z), \tag{5.12}$$

where the summing goes over all the possible clusterings.

Let us now consider the complete data evidence $P(D, Z)$,

$$p(D, Z) = \int p(D, Z|\theta)p(\theta)d\theta. \tag{5.13}$$

Let us now recall from (5.2) that

$$p(D, Z|\theta) = \prod_{j=1}^{N}\prod_{k=1}^{K}\left(\alpha_k \prod_{i=1}^{m}\prod_{l=1}^{n_i}\phi_{kil}^{v_{jil}}\right)^{z_{jk}} \tag{5.14}$$

and from (5.5) that

$$p(\theta) = \Gamma\left(\sum_{k=1}^{K}\mu_k\right)\prod_{k=1}^{K}\left(\frac{\alpha_k^{\mu_k-1}}{\Gamma(\mu_k)}\right)\prod_{k=1}^{K}\prod_{i=1}^{m}\left(\Gamma(\sum_{l=1}^{n_i}\sigma_{kil})\prod_{l=1}^{n_i}\frac{\phi_{kil}^{\sigma_{kil}-1}}{\Gamma(\sigma_{kil})}\right). \tag{5.15}$$

Then we see that the integrand $p(D,Z|\theta)p(\theta)$ can be written as

$$C\cdot\left(\prod_{k=1}^{K}\alpha_k^{h_k+\mu_k-1}\right)\prod_{k=1}^{K}\prod_{i=1}^{m}\prod_{l=1}^{n_i}\phi_{kil}^{f_{kil}+\sigma_{kil}-1}, \tag{5.16}$$

where C is the normalizing constant,

$$C = \frac{\Gamma\left(\sum_{k=1}^{K}\mu_k\right)}{\prod_{k=1}^{K}\Gamma(\mu_k)}\prod_{k=1}^{K}\prod_{i=1}^{m}\frac{\Gamma\left(\sum_{l=1}^{n_i}\sigma_{kil}\right)}{\prod_{l=1}^{n_i}\Gamma(\sigma_{kil})}. \tag{5.17}$$

Now we can make use of the fact that the parameters α and ϕ_{ki} appear disjointly in (5.16), and decompose the integral (5.14) into a product of integrals:

$$p(D,Z) = C\cdot\left(\int\prod_{k=1}^{K}\alpha_k^{h_k+\mu_k-1}d\alpha\right)\prod_{k=1}^{K}\prod_{i=1}^{m}\int\prod_{l=1}^{n_i}\phi_{kil}^{f_{kil}+\sigma_{kil}-1}d\phi_{ki}. \tag{5.18}$$

These integral forms are called *Dirichlet integrals* with a known solution (see [147, p. 178]), consequently (5.18) becomes

$$p(D,Z) = C\cdot\frac{\prod_{k=1}^{K}\Gamma(h_k+\mu_k)}{\Gamma\left(\sum_{k=1}^{K}(h_k+\mu_k)\right)}\cdot\prod_{k=1}^{K}\prod_{i=1}^{m}\frac{\prod_{l=1}^{n_i}\Gamma(f_{kil}+\sigma_{kil})}{\Gamma\left(\sum_{l=1}^{n_i}(f_{kil}+\sigma_{kil})\right)}. \tag{5.19}$$

From (5.19) and (5.17) we then have

$$\begin{aligned} p(D,Z) &= \frac{\Gamma\left(\sum_{k=1}^{K}\mu_k\right)}{\Gamma\left(N+\sum_{k=1}^{K}\mu_k\right)}\prod_{k=1}^{K}\frac{\Gamma(h_k+\mu_k)}{\Gamma(\mu_k)} \\ &\cdot\prod_{k=1}^{K}\prod_{i=1}^{m}\left(\frac{\Gamma\left(\sum_{l=1}^{n_i}\sigma_{kil}\right)}{\Gamma\left(h_k+\sum_{l=1}^{n_i}\sigma_{kil}\right)}\prod_{l=1}^{n_i}\frac{\Gamma(f_{kil}+\sigma_{kil})}{\Gamma(\sigma_{kil})}\right). \end{aligned} \tag{5.20}$$

The complete data evidence formula (5.20) for the discrete finite mixture family can be seen as a generalization of the work in [60], where a similar result was derived for finite mixtures with only binary variables and uniform prior distributions for the parameters (we used discrete variables with Dirichlet priors). The result was derived independently of [66], where a similar result is given for the general family of Bayesian networks. Consequently, as finite mixture models can be regarded as a special case of Bayesian networks, the formula for computing the evidence can also be obtained by adapting the corresponding general result for Bayesian networks.

From this derivation it now follows that we can in principle compute the incomplete data evidence (5.11) by using Equation (5.12) together with (5.20). Analogously to computing the incomplete data posterior, the incomplete data evidence also requires summing over all the possible clusterings, i.e., computing an exponential sum. Thus in practical situations `HAL` has two choices:

i. **Use Naive Bayes model family $\mathcal{M}_{\mathbf{NB}}$**
`HAL` can restrict the model family from the general discrete finite mixtures to the Naive Bayes model family, for which the complete data evidence (5.20) can be computed efficiently. With $\mathcal{M}_{\mathrm{NB}}$ `HAL` can predict using p_{ev} with no approximations.

ii. **Use an approximation to $p(D)$**
`HAL` can use a computationally feasible approximation of the evidence.

The choice between these two alternatives is dependent on the fact, whether the stronger independence assumptions of $\mathcal{M}_{\mathrm{NB}}$ can be justified (even approximately) in the problem in question. The experimental work Chapter 6 indicates that in general the alternatives produce comparable predictions.

Most approximations to $p(D)$ are based on Laplace's method of integration (see the discussion in e.g., [9, 77]), where the logarithm of the integrand of the evidence (5.11) is expanded at the posterior mode $\hat{\theta}$. Laplace's approximation is based on the assumption that when the amount of data N grows, the posterior

$$p(\theta|D) \propto p(D|\theta)p(\theta)$$

can be approximated as a multivariate Gaussian distribution. In order to approximate $\log(p(D|\theta)p(\theta))$ one can use a second degree Taylor polynomial about $\hat{\theta}$, and thus obtain

$$\log(p(D|\theta)p(\theta)) \approx \log(p(D|\hat{\theta})p(\hat{\theta})) - \frac{1}{2}(\theta - \hat{\theta})^T \tilde{\Sigma}(\theta - \hat{\theta}), \tag{5.21}$$

where $\tilde{\Sigma}$ is the negative Hessian of $\log(p(D|\theta)p(\theta))$ evaluated at $\hat{\theta}$. Substituting (5.21) to (5.11), integrating and taking the logarithm we now have Laplace's approximation:

$$\log p(D) \approx \log p(D|\hat{\theta}) + \log p(\hat{\theta}) + \frac{dim}{2}\log(2\pi) - \frac{1}{2}\log|\tilde{\Sigma}|, \tag{5.22}$$

where *dim* is the dimension of the model, i.e., the number of parameters.

In practice for computing efficiency Laplace's approximation is applied in alternative forms by retaining only those terms in (5.22) that depend on N: $\log p(D|\hat{\theta})$ and $\log|\tilde{\Sigma}|$. An example of this type of an approximation is the *Bayesian Information Criterion (BIC)* [123], also known as the Schwarz criterion, which is based on the observation that for large N the determinant

$$|\tilde{\Sigma}| \propto N^{dim}$$

and that $\hat{\theta}$ can be approximated by $\tilde{\theta}$, thus leading to an approximation without any need for prior terms given by

$$\log p(D) \approx \log p(D|\tilde{\theta}) - \frac{1}{2} dim \log N, \tag{5.23}$$

where *dim* is the number of parameters. This approximation can also be given an information theoretic interpretation in the Minimum Description Length (MDL) setting, as demonstrated in [116].

The BIC approximation can also be used as a motivation for another approximation method. From (5.23) we know that with increasing N the evidence $p(D) \approx p(D|\tilde{\theta}) \cdot C$, where C is a term depending only on N, and on the dimensionality of θ. Similarly, we get

$$p(D, Z) \approx p(D, Z|\tilde{\theta}) \cdot C.$$

Now solving C in both cases gives

$$p(D) \approx p(D, Z) \cdot \frac{p(D|\hat{\theta})}{p(D, Z|\hat{\theta})}. \tag{5.24}$$

If we now replace $\tilde{\theta}$ by $\hat{\theta}$, take the logarithm of the right-side of (5.24) and replace the unobserved data Z with its conditional expectation given the data D and the parameters $\hat{\theta}$, we get the *Cheeseman-Stutz* (C-S) approximation, which is used in the AutoClass system [24][2].

5.4 Approximating the evidence

In the literature several methods for computing the evidence $P(D)$ approximately, including the ones discussed (BIC and Cheeseman-Stutz), have been suggested (see e.g., [5, 15, 24, 77, 116, 123, 134]). The quality of most of

[2] An alternative way of deriving the C-S approximation is given by Chickering and Heckerman in [25].

these approximations is not well known, except for some asymptotic results. As can be seen from the experiments in Chapter 6, typical data sets that HAL encounters in real life are small in the asymptotic sense. Therefore we will now investigate empirically the performance of the most common incomplete evidence $p(D)$ approximations in an attempt to understand their small sample behavior in the finite mixture framework. In many earlier similar studies (see e.g., [112, 119]), the model family used has either been too restricted for extending the results to real-world domains, or too general to allow an exact solution to be used for the comparisons.

The evaluation of the quality of the approximations faces the obvious problem that for any reasonable sized data set calculating the exact incomplete data evidence (5.11) is not feasible. In a recent continuation of the work discussed here [85], some of the incomplete evidence approximations are compared to $p(D)$, which is computed by the "brute force method," i.e., by actually computing complete data evidence for all the clusterings of the data vectors. The study supported our empirical observation of the viability of the Cheeseman-Stutz approximation, but also revealed the sensitivity of the approximation to the selection of approximate $\theta' \approx \hat{\theta}$, if the exact $\hat{\theta}$ cannot be found.

In [25], the problem of not knowing the actual incomplete evidence is circumvented by using synthetic data, in which case the correct value is "implicitly known," and the number of mixing distributions used for generating data can be controlled. Unfortunately such an empirical study can face a serious validity problem, as one does not know whether the results could be generalized to real-world problem domains, or whether they are simply caused by some anomaly in the artificial data generating method. It should be pointed out that when validating approximative evidence measures against generated data, one should be extremely careful in providing samples that are representative to the intended mixing distributions. Negative results, i.e., approximations suggesting model classes differing from the "true number" of mixture components M_k can also be caused by the fact that the data in the sample can indeed be described best with a different model class $M_{k'}$, since no finite sample can capture all the information of the generating process. The amounts of data needed to represent the underlying distribution are substantial (thousands of data vectors for parameter spaces of only moderate dimensionality), which defies the whole purpose of finding out the approximation quality for small sample sizes encountered in real life. The results reported in [25] clearly reflect this difficulty.

In order to investigate this issue of incomplete evidence approximation we performed two sets of experiments with three approximations from the

Laplace's approximation family. We first study the behavior of Akaike Information Criteria AIC [5] and BIC [123] with respect to complete evidence $p(D, Z)$. As Cheeseman-Stutz [24] offers an approximation for the incomplete evidence using the complete evidence, we could not include it in this set of experiments. In the second set of experiments, however, we compare Cheeseman-Stutz against cross-validation [133] for a typical use of the incomplete evidence: model class selection. The first set of experiments used real data sets and the second both synthetic (generated) and real data sets. The data sets used here are among those discussed in Chapter 6.

AIC and BIC approximations vs. complete evidence

From Section 5.3 (Equations (5.11) and (5.13)) we know that each term in the exponential sum required for computing the incomplete evidence is itself an integral of the same form. This in fact is one of the intuitions behind the Cheeseman-Stutz approximation. Therefore it might be reasonable to assume that any method capable of approximating such integrals well in general is also a good approximator for computing the incomplete evidence. Hence in the first set of experiments we use the complete data integral which is closest to the incomplete data integral, corresponding to the Z assignment with the highest probability. We compute the complete data evidence for different model classes by using (5.20), and compare the results to the results obtained by Bayesian Information Criterion (BIC), and Akaike Information Criterion (AIC), to be defined below. It should be observed that we do not suggest here that one complete data evidence term should be used for practical model selection tasks—it is used only as a tool for evaluating the quality of the BIC and AIC approximations.

Let us recall that the Bayesian Information Criterion [123] approximates the incomplete data evidence by

$$\log p(D) \approx \log p(D|\tilde{\theta}) - \frac{1}{2} dim \log N, \tag{5.25}$$

where dim is the number of parameters ($dim = K(1 + \sum_{i=1}^{m} n_i) - (Km + 1)$). The *Akaike Information Criteria* approximation [5] is a Laplace's approximation which is even simpler:

$$\log p(D) \approx \log p(D|\tilde{\theta}) - dim.$$

Both of these approximations are interesting in the sense that they are quite intuitive and do not require assessing any prior. Namely, they contain a likelihood term measuring how well the Maximum Likelihood (or MAP if uniform priors are used) model in the model class predicts the data, and a

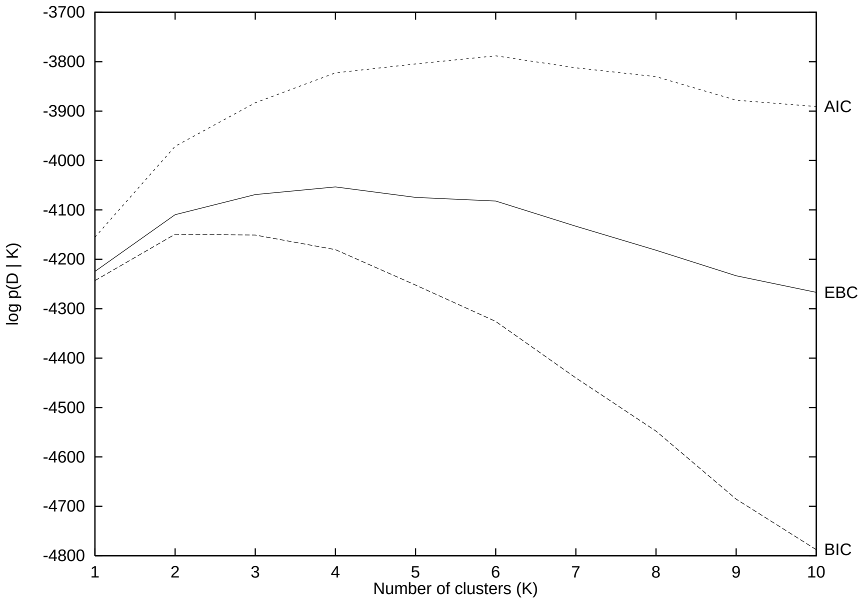


Figure 5.3: Results with the Primary Tumor dataset.

"regularization term" that penalizes the complexity of the model. For more technical justifications of these approximations, see e.g., [5, 77, 123].

In order to be able to compute the complete data evidence, we need the cluster assignment Z. Here we have chosen to use the clustering corresponding to the approximate MAP parameters, and use the EM-algorithm described earlier to find a good approximation of $\tilde{\theta}$. Finding the cluster assignments Z for a fixed model θ is trivial: each data vector is assigned to the most probable cluster by setting

$$z_{jk} = \begin{cases} 1, & \text{for } k = \arg\max_{k'=1,\ldots,K}\{\alpha_{k'} \prod_{i=1}^{m} \phi_{k'id_{ji}}\} \\ 0, & \text{otherwise.} \end{cases} \tag{5.26}$$

In our experiments, we used five public domain classification datasets from the same collection as before. The datasets were of varying size, and contained natural data from various problem domains. We chose these five data sets from the larger set to reflect various different features in the domains `HAL` could encounter in practice: very small training data set (Iris), large data set with many attributes (DNA), a binary classification case (Diabetes), a multi-class classification (Primary Tumor) and a "difficult" case where the results with various alternative prediction methods have a high variance (Glass).

For each dataset, and for each model class $M_K, K = 1, 2, \ldots$, we estimated the optimal clustering Z by the EM-algorithm. In each case, the algorithm was repeated 500 (with the DNA dataset 50) times by starting with randomly chosen initial states. In each trial, the algorithm was run until converged, which took typically 10–20 iterations. Like for all the empirical tests in this work, uniform priors were used (i.e., we set $\mu_k = 1, \sigma_{kil} = 1$, for all k, i, and l). Consequently the clusterings could be ranked according to their likelihoods $P(D|\hat{\theta}_Z)$, where $\hat{\theta}_Z$ denotes the MAP parameter values corresponding to a clustering Z. The best of the 500 locally optimal clusterings found was then chosen to be the clustering used in our experiments, and the complete data evidence was then computed by using (5.20). The result was then compared to the approximations given by the BIC and the AIC methods.

The results for each of the five datasets are presented in Figures C.1–C.5 given in the Appendix C. Here we show only one illustrative example in Figure 5.3, which depicts the results from the experiments with the Primary Tumor data set (EBC denotes here the exact evidence formula given by (5.20)). In the Appendix, where the complete results are given, the graphs are sorted in ascending order by the size of the corresponding dataset. From the results we can see that the approximations give estimates which are consistently too large (AIC) or too small (BIC), when compared to the exact solution. This is not important, however, if we are interested in using the criterion for comparing different model classes, in which case the relative shape of the curves, and the location of the maxima are more interesting. As we can see, the approximation curves follow quite accurately the shape of the exact solution curve, especially with the larger datasets. An interesting observation is that the approximative model class estimators perform very well already with a sample size of about 300 data vectors for an 18-dimensional problem (i.e., 18 domain attributes), and extremely well for a 181-dimensional problem with a sample size of 3000.

Cheeseman-Stutz approximation vs. cross-validation

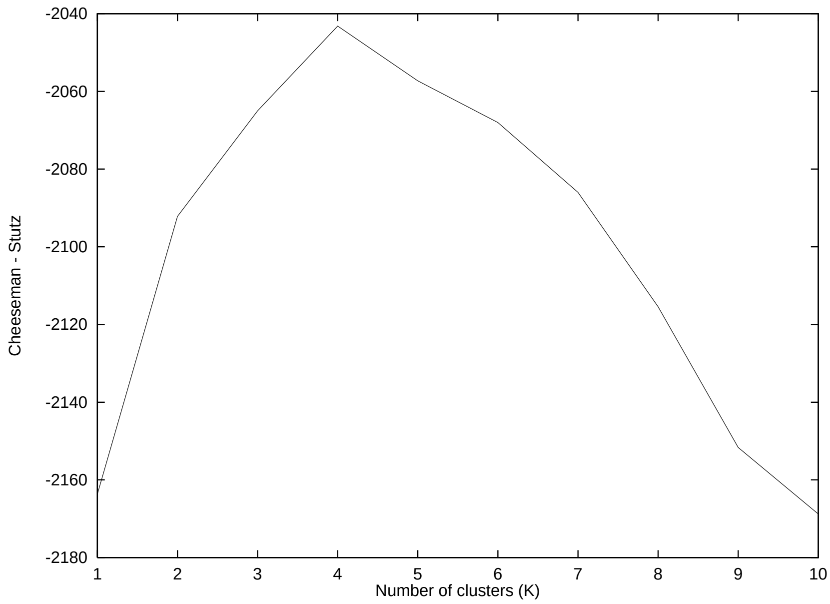
In Chickering and Heckerman's study [25] the Cheeseman-Stutz approximation was found to outperform AIC and BIC for synthetic data, which seems to indicate that C-S is a good approximation of $p(D)$ for the specific purpose of selecting the "true" model class M. Their approach corresponds to an exploratory data mining setting were one wants to examine whether the Bayesian model class selection scheme can be used for determining the "correct" model class for the problem domain in question. As pointed out earlier, in this exploratory type of an analysis one has to be careful in generating the sample to be representative enough, otherwise the evidence measure tends to suggest overly simple structures. We will illustrate this by evaluating C-S in an experiment with also synthetic data sets, which clearly demonstrate this problem when the dimensionality of the model is increased.

Even putting the philosophical issues aside, since we know that in the general case the non-identifiability of finite mixtures [138] prevents `HAL` from uniquely determining the components from data, this exploratory view point is only of marginal interest to us. For `HAL` the "correct" model class of the data is M which produces the best predictions. Hence an interesting question is, what is the quality of the Cheeseman-Stutz approximation when evaluated by the predictive performance of the model class it selects. To evaluate the feasibility of C-S approximation for predictive modeling, i.e., for determining the optimal model class for predictive purposes, we use natural data sets and compare the results to those suggested by cross-validation.

The data sets used in these experiments are again from the same collection of data sets used also in Chapter 6. The Australian, Heart Disease and Lymphography domains are illustrative of the behavior of the C-S approximation for all the data sets. DG10 and DG20 are synthetic datasets with 2000 data vectors generated by sampling random mixture models with a variable number of clusters, the number of domain attributes being 10 and 20, respectively. The number of possible attribute values varied between 2 and 4.

The Cheeseman-Stutz measure requires the calculation of the MAP estimate $\hat{\theta}$. Since we know that the EM algorithm produces only locally optimal approximations of θ, for each test case 50 repetitions of EM with random initialization were performed, of which the highest posterior θ^b was chosen to represent the maximum a posteriori model $\hat{\theta}$. Convergence of the EM algorithm proved to be very fast, in the order of 100 or less iterations for each individual run.

Figures D.1, D.2 and D.3 in Appendix D illustrate the behavior of the Cheeseman-Stutz approximation and the cross-validation results for model classes with $K = 1, \ldots, 10$. We will show here only the results for the Lymphography data set (Figure 5.4). For all three datasets the Cheeseman-Stutz model class suggestions coincide with the cross-validation results with a relatively high accuracy. It should be pointed out, however, that the cross-validated results themselves are an approximation, and therefore not decisive for both theoretical and pragmatic reasons. For efficiency reasons only 10-fold cross-validation was used, which can have high variance depending on the partitioning to folders (see the discussion in Chapter 6). This factor could be removed if computationally more demanding leave-one-out cross-



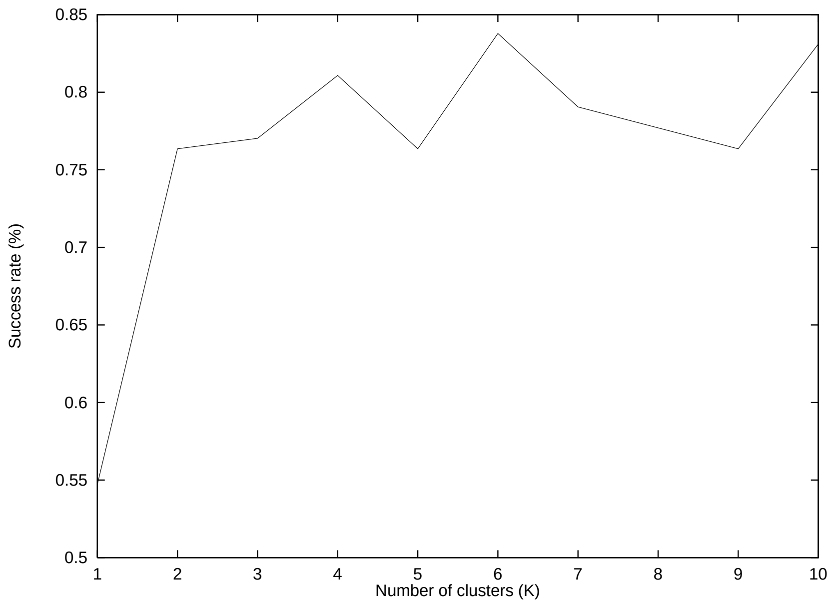


Figure 5.4: The C-S measure (up) and the cross-validation results (below) with the Lymphography dataset. The property of interest here is the model class selected, i.e., the number of clusters for which the measures give the maximum value.

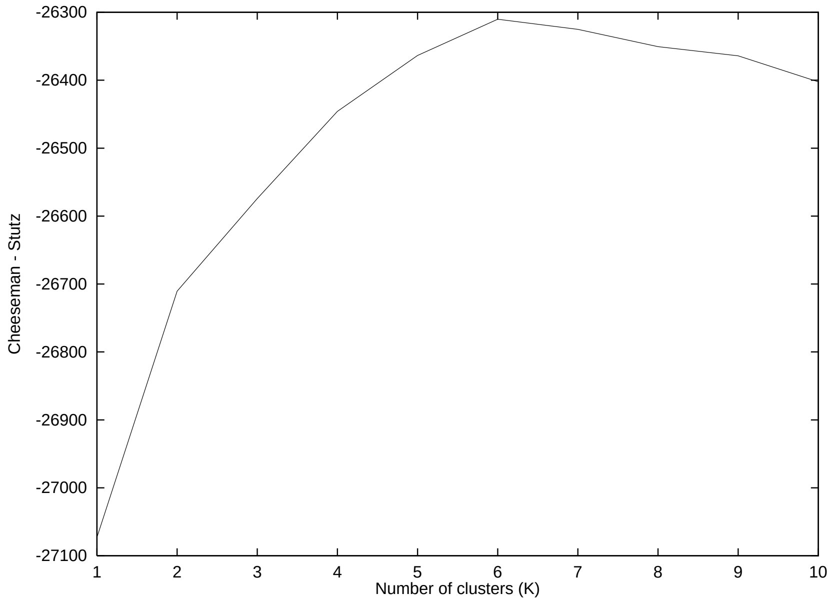


Figure 5.5: The C-S measure for cluster counts 8 with the DG20 dataset.

validation were used. Even if we had used leave-one-out cross-validation, we cannot expect a perfect match for two theoretical reasons, which again illustrate the problem of validating evidence approximations.

First of all, one should keep in mind that the cross-validation error is computed by measuring the accuracy of one single classification variable, whereas the evidence criterion is a measure of the overall accuracy of the underlying probability distribution *for all the variables.* Second, we know that the cross-validation measure is in fact an average value of one of the terms in the sequential decomposition of the evidence [26]. That is the log evidence can be decomposed as a sum of "online" predictive performances:

$$\begin{aligned}\log p(D|M) = &\log p(\vec{d}_1|M) + \log p(\vec{d}_2|\vec{d}_1, M) + \\ &\log p(\vec{d}_3|\vec{d}_1, \vec{d}_2, M) + \log p(\vec{d}_N|\vec{d}_1, \ldots, \vec{d}_{N-1}, M).\end{aligned}$$

Cross-validation examines the average value of just the last term

$$\log p(\vec{d}_N|\vec{d}_1, \ldots, \vec{d}_{N-1}, M)$$

under random re-orderings of the data.

Thus we are in fact judging the quality of an approximation against another approximation, albeit a pragmatic one. The relationship between the evidence (called "scientific criterion") and cross-validation measure (called

Algorithm 5.4.1
Bayesian model construction for finite mixtures with Cheeseman-Stutz approximation

Input:	D,	*data set*
	K_{limit},	*maximum number of components in the model class*
	EM_{limit},	*number of EM runs per model class*
	EM_{conv},	*EM convergence criteria*

Output: *locally optimal MAP approximation* θ' *from* M' *that approximates* $\hat{M}$.

i. ITERATE K from 1 until $\mathrm{K_{limit}}$:
   (a) ITERATE R from 1 until $\mathrm{EM_{limit}}$:
      i. Run Algorithm 5.2.2 until $\mathrm{EM_{conv}}$.
   (b) Let $\theta'_K = \arg\max_{\theta^R}\{p(\theta^R|D, M_K)\}$.

ii. Let $K' = \arg\max_K\{\text{Cheeseman-Stutz}(K)|K = 1, \ldots, \mathrm{K_{limit}}\}$.

iii. Return the model $\theta'_{K'}$ from model class $M_{K'}$.

"engineering criterion") together with some additional experimental results in model class selection tasks are discussed in [65].

As with the natural data, in the synthetic data experiments the Cheeseman-Stutz approximation was determined for model classes $K = 1, \ldots, 10$. The synthetic data was generated by varying the number of generating clusters (G) from 2 to 8, and the Cheeseman-Stutz behavior was compared to the so called "gold standard", i.e., the number of components used in the data generation. As seen from the Figures D.4–D.7 in Appendix D, the Cheeseman-Stutz approximation coincides well with the underlying cluster structure with both datasets when the number of clusters is less than 8. For the 8 cluster case shown here (Figure 5.5) the datasets are clearly too small to be able to represent the problem domain probability distribution well enough, and we see that the data "justifies" a simpler structure than we have used in the data generation process.

5.5 Model construction in a "nutshell"

We conclude the discussion on HAL's model construction by summarizing the discussion of this chapter as Algorithm 5.4.1. Naturally HAL could replace the Cheeseman-Stutz incomplete evidence approximation by other Laplace's approximations (see Section 5.3). From pragmatic point of view it should be observed that HAL needs to decide only on very few input parameters: the maximum size of the model class K_{limit}, the number of EM runs performed for each model class EM_{limit}, and the convergence criteria EM_{conv} for an individual EM run.

Chapter 6

Bayesian classification with finite mixtures

"The word "valid" would be better dropped from the statistical vocabulary. The only real validation of statistical analysis, or of any scientific enquiry, is confirmation by independent observations."

—Anscombe in Topics in the investigation of linear relations fitted by the method of least squares

In the preceding chapters we have discussed `HAL`'s predictive framework, starting from the general principles of using probabilities as plausibilities, and ending in Bayesian methods for using a specific model family of finite mixtures of multinomial components for plausible predictions in the presence of uncertainty. It is now time to put these ideas to test, and see how the developed machinery for predictive inference can be applied. `HAL`'s design was motivated by the need to infer predictive models from data, i.e., finding data constraints (expressed with the language of models) that help it to predict properties of unknown quantities. The advantage of predictive models is that they can be validated at least to a degree by inspecting the actual predictive performance, either with benchmark data sets or with real applications. The Bayesian predictive framework described here has been tested in both respects. In this chapter we report empirical results with common benchmarks, and evaluate the Bayesian predictions with finite mixtures against the results achieved with other popular model families such as neural networks and decision trees. Some of the results described in this chapter have appeared in preliminary form in [136].

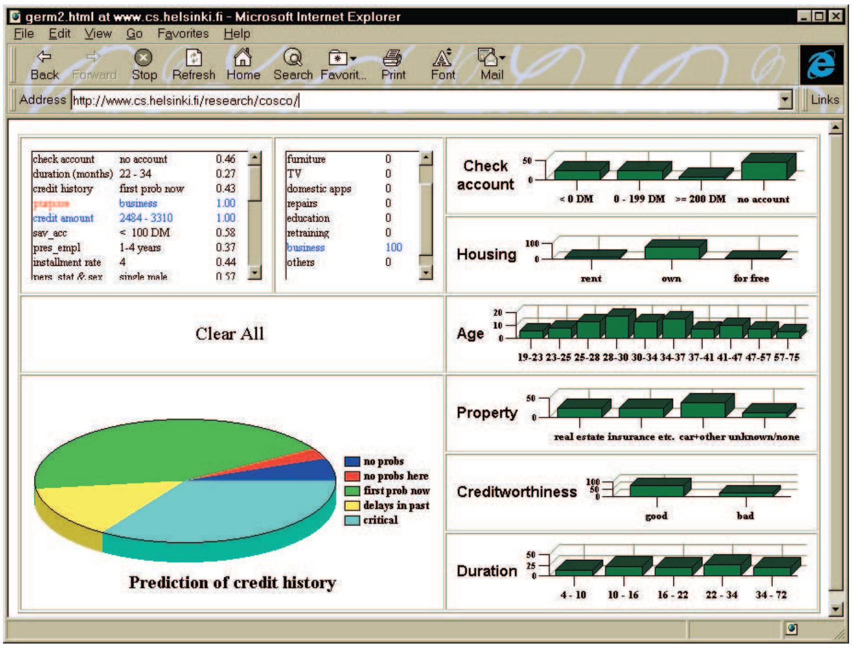


Figure 6.1: A snapshot of the interface of the D-SIDE prediction engine.

6.1 Data sets and experimental setting

The general Bayesian approach for performing plausible predictions offers a theoretically justifiable and consistent framework for HAL to predict unknown properties of new arriving data. We have "instantiated" the Bayesian framework in the case where HAL's models are finite mixtures of multinomial distributions. The approach described forms the kernel of the D-SIDE/C-SIDE software programmed in Java/C. The Bayesian prediction engine D-SIDE also provides a flexible graphical user interface for displaying predictive distributions and the mixture structure (see Figure 6.1), and can be used with any Java compatible World Wide Web-browser[1].

From the discussion in Chapters 4 and 5 we know that for a restricted sub-family of finite mixtures, the Naive Bayes family $\mathcal{M}_{\mathrm{NB}}$, HAL can be "fully Bayesian" and integrate out all the models, but for the general finite mixture case it has to use approximations in its predictions. These approximations are theoretically justified, but the quality of the approximations is

[1]A running demonstration of the D-SIDE software can be accessed through the CoSCo homepage at URL "http://www.cs.Helsinki.FI/research/cosco/".

only known asymptotically, i.e., when the data set D is large. Unfortunately in many application domains, especially industrial, we know that this is not the case. We have already pointed out that using synthetic data for empirical tests requires substantial amounts of data to allow any reliable conclusions (see the discussion in Section 5.4 and below), which defies the whole purpose of finding out the performance behavior with small data sets. Therefore the only true test of the approaches we have presented is to perform empirical testing with real data sets. In addition such tests have to be comparative in nature, otherwise we would not have any idea how well `HAL` is doing. This latter requirement makes it very difficult to use real industrial application data sets, as usually for such data sets no previous results with alternative methods exist.

A thorough comparison would use synthetic data or real application data sets, and compare several different methods by an extensive set of experiments. This is a very time-consuming task, but would allow us also to calculate for example the standard statistical significance values for the differences found. In such a comparative study, however, a serious validity problem would exist. Almost all of the learning/prediction algorithms proposed in the literature are parameterized (model size, parameter estimation convergence speed, post-processing parameters etc.). This is natural, since any model construction procedure has to somehow decide the set of models explored and the search algorithm that it uses, both of which require fixing some parameters. Consequently, the results of a particular method are also sensitive to the parameter selection. Setting the proper parameters usually requires good understanding of the method itself, and affects drastically the results achieved (see e.g., the improvements achieved by an automated parameter selection process reported in [79]). Since it is not feasible to assume that we are experts in all the possible different types of algorithms, the comparison results would be severely hampered by our (in)competence in the method used[2]. Therefore, instead of using artificial or real industrial data we decided to test `HAL`'s prediction performance for publicly available real data sets.

We are well aware of the inherent problems in using such data sets, especially for comparison purposes: it is very difficult to identify underlying causes for performance differences, there is a high variance in the observed performance differences etc. The main advantage of using natural datasets is that they are produced without any knowledge of the particular procedures

[2]See the discussion on an attempt to a fair comparison of several machine learning and computational intelligence approaches in the StatLog-project [99], and the difficulties encountered.

that they are tested on. In addition, although there is no way of telling how the results for a real data set generalizes to other problems, we at least know that there are some domains where the results have practical relevance.

Below we report results from extensive experimentation with the Bayesian finite mixture framework using publicly available data sets for classification problems, which conform to the general prediction problem as defined in Section 3.2. To be able to compare the predictive performance, we have collected from the literature performance results for alternative methods using the same data sets. While we make no claims that the list of the results of alternative algorithms is exhaustive, for each data set we have included the best results we have found in the literature. The selection of data sets in the comparison was done on the basis of their reported use, i.e., we have preferred data sets that have been used for testing many different methods over data sets with only isolated results. Therefore many of the results are from the StatLog project [99], but we have also included more recent results. The descriptions of the data sets and the testing procedures used for each data set are given in Appendix A. A more detailed description of these data sets can be found in [99] and in the documentation in the UCI data repository.

It should be observed that with the exception of the DNA dataset, all our results are cross-validated. In fact, to evaluate `HAL`'s predictive performance there is no technical reason not to cross-validate DNA results also, the choice of a "train and test" result was purely for comparative reasons. When possible (for the StatLog datasets) we have used the same cross-validation scheme as described in [99]. It should be observed that the same does not hold for many of the results for the other methods. In many cases the testing procedure either was not reported, or the best result with a single test set was given. In the comparisons we have adopted the conservative viewpoint that all the results are comparable to our cross-validated results. Even more, as demonstrated below, any cross-validation procedure departing from the leave-one-out scheme used in Section 4.4 suffers from the variance caused by the selection of the folds.

Final remark concerns the preprocessing of the data sets. Most of the data sets were already discrete from the beginning. Those that were not could have been discretized by using a discretization scheme based on the Bayesian approach itself [83]. Here we have used a simpler method, where the attributes were discretized by using k-means clustering (a special case of the EM algorithm with "hard" clusters) starting from a uniform initial state. As demonstrated in the experiments reported in [45], in general the prediction results are not very sensitive to the discretization scheme, assuming that a reasonable standard scheme such as ours is used.

6.2 Empirical results

Two sets of experiments were performed with these data sets. The first set of experiments used the Naive Bayes model family $\mathcal{M}_{\rm NB}$ with all the three predictive distributions $p_{\rm map}$ (4.7), $p_{\rm ev}$ (4.8) and $p_{\rm sc}$ (4.9). The second set of experiments used the general mixture family with MAP prediction (4.5). Both sets of experiments used 0/1 score as the performance measure, i.e., the percentage of correct classifications.

6.2.1 Naive Bayes results

For the Naive Bayes model family we tested all the three prediction methods with the ten data sets. We partitioned each data set (except DNA) to random cross-validation folds 100 times using the number of folds reported in the literature, and evaluated the results for each partitioning. The minimum, mean, maximum and the variance of the results for each data set for each method are summarized in Table 6.1.

Before entering the discussion on the results themselves, we want to point out the problem of the comparative methodology when k-fold cross validated results from different sources are compared, a practice common in the machine learning and computational intelligence communities. As the Naive Bayes results demonstrate, there is a considerable difference in the results depending on which partitioning to folds was used. Obviously, this problem ceases to exist if one moves to leave-one-out cross-validation, but unfortunately it is very seldom used in the literature. As the results reported in the literature are not cross-validation means (and in some cases not even cross-validated results), in the bar-charts in Figures 6.2–6.6 we have used the maximums. We want to stress that this is done *for comparative purposes only* in order to put our results in perspective with alternative approaches. `HAL`'s true prediction performance is closer to the means of the results.

The results are consistent with the leave-one-out cross-validated experiments in Chapter 4 when the differences between the predictive distributions were considered. Both the $p_{\rm map}$ and $p_{\rm ev}$ show consistent good performance, and we can see that the larger the data set, the smaller is the performance difference. This is of course only an empirical confirmation of the observation already mentioned in Section 3.1.3; if the training set size D is increased the posterior becomes more peaked and therefore the single MAP model tends to represent the posterior well. Similarly, as expected, the variance of $p_{\rm map}$

Data set		NB-MAP	NB-EV	NB-SC
DNA	train & test	**0.945**	0.944	0.885
Diabetes	minimum	0.746	0.746	0.745
	mean	0.756	**0.757**	0.755
	maximum	0.766	**0.767**	0.765
	variance	0.000074	0.000037	0.000067
Australian	minimum	0.843	0.843	0.843
	mean	**0.850**	**0.850**	0.849
	maximum	**0.857**	**0.857**	0.855
	variance	0.000007	0.000006	0.000006
Primary Tumor	minimum	0.428	0.467	0.280
	mean	0.456	**0.487**	0.339
	maximum	0.478	**0.507**	0.395
	variance	0.000106	0.000080	0.000533
Breast Cancer	minimum	0.700	0.710	0.703
	mean	0.720	**0.722**	**0.722**
	maximum	0.745	0.741	**0.749**
	variance	0.000074	0.000037	0.000067
Heart Disease	minimum	0.819	0.826	0.826
	mean	0.835	**0.840**	**0.840**
	maximum	0.856	0.852	**0.856**
	variance	0.000039	0.000031	0.000037
Glass	minimum	0.621	0.631	0.612
	mean	**0.687**	0.666	0.649
	maximum	**0.729**	0.696	0.673
	variance	0.000204	0.000146	0.000176
Hepatitis	minimum	0.827	0.793	0.800
	mean	**0.845**	0.817	0.822
	maximum	**0.873**	0.847	0.847
	variance	0.000095	0.000101	0.000073
Iris	minimum	0.927	0.933	0.933
	mean	0.939	0.944	**0.947**
	maximum	0.953	0.953	**0.960**
	variance	0.000259	0.000122	0.000034
Lymphography	minimum	0.777	0.811	0.635
	mean	0.811	**0.844**	0.772
	maximum	0.845	**0.872**	0.838
	variance	0.000259	0.001214	0.002865

Table 6.1: Naive Bayes model family cross-validation results on the ten data sets and for MAP (NB-MAP), evidence (NB-EV) and stochastic complexity (NB-SC) predictive distributions. The results are for 0/1-score. For each data set (except DNA) the minimum, mean, maximum and variance of the 100 partitionings to folds is reported.

performance is higher than that of p_{ev}, for which we know the small sample behavior to be the most robust from the earlier results.

The performance of p_{sc} is comparable, but slightly worse than p_{map} or p_{ev} on the average. One has to remember, however, that p_{sc} was designed to be a good predictor in the coding sense, i.e., it is designed to minimize the log-score, not the 0/1 score. And that it indeed does. Although not reported here, for the log-score results p_{sc} shows as good or only slightly worse performance than p_{ev}, which itself outperforms clearly the p_{map} predictions in all the ten data sets. This warns us to be careful not to draw too general conclusions about the predictive methods based on results for a single loss function, especially as coarse one as 0/1. The log-score results indicate that for other types of loss functions, where a more precise estimate of the predictive distribution than the mode is needed, both the p_{sc} and p_{ev} will very likely outperform p_{map}.

Finally, when compared to the results in the literature, the Naive Bayes results are very competitive and actually in some cases outperform the results with finite mixtures (Figures 6.2–6.6). In the bar-charts the results with the Naive Bayes family have been denoted by `D-SIDE NB (MAP)`, `D-SIDE NB (EV)`, and `D-SIDE NB (SC)` corresponding to p_{map}, p_{ev}, and p_{sc}, respectively. It should be observed that with the exception of the Heart Disease data set, when the Naive Bayes result outperforms the finite mixture result, either the evidence or stochastic complexity predictive distribution has been used for predictions. In general `HAL` is able to find a better single model in the finite mixture family than from the Naive Bayes family.

6.2.2 Finite mixture results

Using the Naive Bayes family has the advantage that no approximations are needed, thus the prediction performance is mainly restricted by the simplicity of the language to express the constraints in the data. For finite mixtures the situation is completely different. Choosing the optimal model class M with incomplete evidence requires an approximation like Cheeseman-Stutz or BIC, and even for finding the MAP parameters within a model class `HAL` has to submit to the local maxima found by the EM algorithm. Therefore let us now explore the trade-off between a more expressive model language and the need to approximate the Bayesian ideal in the prediction.

For the finite mixture model family we repeated the procedure described above, but for comparison we report only the achieved maximums. The selection of the model class was performed by using the Cheeseman-Stutz approximation, and from the chosen model class M among the 50 models produced by EM the model θ with the highest posterior was used for pre-

Data set	Model class	FM-MAP	NB-MAP	NB-EV	NB-SC
DNA	13	**0.970**	0.945	0.944	0.885
Diabetes	20	**0.773**	0.766	0.767	0.765
Australian	17	**0.872**	0.857	0.857	0.855
Primary Tumor	21	0.504	0.478	**0.507**	0.395
Breast Cancer	21	**0.766**	0.745	0.741	0.749
Heart Disease	8	0.848	0.856	0.852	**0.856**
Glass	30	**0.874**	0.729	0.696	0.673
Hepatitis	9	**0.880**	0.873	0.847	0.847
Iris	4	**0.980**	0.953	0.953	0.960
Lymphography	19	0.866	0.845	**0.872**	0.838

Table 6.2: Cross-validation results on the ten data sets for finite mixture MAP predictive distribution (FM-MAP) and Naive Bayes model family MAP (NB-MAP), evidence (NB-EV) and stochastic complexity (NB-SC) predictive distributions. The results are for 0/1-score, and we have reported also the best model class M.

diction.

The results for the finite mixture family (denoted by `D-SIDE`) for the individual datasets together with the corresponding Naive Bayes results are presented in Table 6.2. The comparison of both the Naive Bayes and finite mixture results to the ones in the literature are presented as bar-charts in Figures 6.2–6.6. As can be expected, from the results we see that for 7 out of the 10 data sets the finite mixture MAP prediction outperforms predictions with the Naive Bayes family. However, in most cases the performance differences are small—in particular considering the variance caused by the fold selection.

6.2.3 The results in perspective

If we compare the results to the results of the alternative approaches reported in the literature, the Bayesian predictive approach offers consistently competitive performance over the various data sets. The observation that the finite mixture based prediction outperforms the memory based methods such as K-NN, K^*, IB3, or ALLOC80 is not especially surprising in the light of the given discussion about the probabilistic interpretation of the instance-based methods (Section 4.2, see also [136]). A more interesting observation is that our Bayesian approach with the finite mixture model family outperforms also all other Bayesian approaches including CASTLE and Bayes tree in the StatLog comparison, recent variants of the Naive Bayes algorithm such as Tree Augmented Naive-Bayes (TAN) and Bayesian networks with MDL score functions [40, 53][3].

We would like to point out that in the comparison the important issue is not so much to focus on the actual performance percentages for individual data sets, since in most cases the differences between the well-performing methods are very small. A more interesting aspect is the consistent high rankings of the Bayesian finite mixture approaches regardless of the data set—a property which is not shared by the alternative approaches (see the discussion in [99]). The results above can also be interpreted as support for the common hypothesis that many real data distributions can be naturally modeled as mixtures of multinomial distributions.

6.2.4 On implementation performance

Our focus has been `HAL`'s predictive performance, and we have not yet discussed any aspects of the implementation performance. It is well-known that for many model families the "training time," i.e., model construction time in our terminology, can be substantial both in theory and practice. On the other hand for many model families the actual prediction can be performed fast, typically in linear time in the number of inputs. An illustrative example for this type of an implementation performance behavior is the family of feed-forward neural networks accompanied with the back-propagation algorithm [19], which tends to be notoriously slow in finding locally optimal models, but once a model is found, the prediction can be performed efficiently. At the opposite end of spectrum are the lazy approaches discussed in Section 4.2, where almost all computation is deferred until the the prediction phase. The commonly used model family of decision trees falls in between these two extremes—at least for the sub-families of bounded rank [44]. Where then do the Bayesian finite mixture approaches stand?

For the Naive Bayes model family $\mathcal{M}_{\mathrm{NB}}$ the situation is very straightforward: the model construction does not require any search, only calculation of the sufficient statistics, and the prediction is based on calculating a simple product in the number of variables. For the finite mixtures the situation is more complex as `HAL` has to search for the optimal model class by an evidence approximation (e.g., Cheeseman-Stutz), and after fixing the model class, `HAL` has to search for a good model θ. Since the evidence approximations also need a good model, most of the time in the model construction phase is

[3]It should be noticed that the Naive (and Successive) Bayes results for the Breast Cancer and Primary Tumor data set by Kononenko are averages over 10 random 70/30% splits of the data, where the variance of the results is even higher than for 10-fold cross-validation.

consumed by the EM algorithm. We have pointed out earlier in Section 5.2 that in practice already a relatively small number of EM iterations (< 100) is sufficient for finding good models θ. This makes it possible for `HAL` to approximate the global optimum $\hat{\theta}$ by several repeated trials with randomly chosen initializations of the parameters. Therefore the model construction time is adjustable depending on the application requirements—in principle one should use all the available time for searching better and better (in the posterior probability) models. This type of stochastic search parallelizes extremely well without any need to special hardware, and allows very efficient use of the available computing resources. The prediction phase using finite mixture models is similar to that of Naive Bayes models except for a summation over the probabilities in each mixture. Since the number of mixture components K is typically much smaller than the size of the training data set N, the prediction is also very efficient.

Sufficient efficiency is a relative concept: an embedded real-time application for telecommunications sets very different performance criteria than a medical diagnostic expert system. To give an idea of the implementation performance of our `C-SIDE`/`D-SIDE` software for finite mixture based prediction, let us consider an experiment we performed with a data set of 2 million data vectors, each consisting of 17 discrete attributes having 2-4 values and a class variable with 21 different values—the largest class containing 24.8% of the data.

The construction of a 100 cluster finite mixture model by running the EM-algorithm for 10 iterations took 9 hours 27 minutes and 34 seconds elapsed time on a 200MHz Pentium machine running Linux/`C-SIDE`. Testing the model with 0/1 p_{map} prediction for the same 2 million data vectors took 26 minutes 33.71 seconds i.e., about 1255 classifications per second, or 0.8 milliseconds per classification. The resulting model achieved 70.5% prediction accuracy.

Naive Bayes model construction with the same data set took 17.02 seconds. The "learning" handled 117509 data vectors per second, i.e., 8.5 microseconds per one data vector. Testing with 0/1 p_{ev} prediction the model with same 2 million training vectors took 29 minutes 13.57 seconds. That is 1141 vectors were classified in one second, i.e., classification of one data vector took 0.9 milliseconds. The prediction accuracy was 61.7%. The size of this data set serves also as an example of the scalability of our approach.

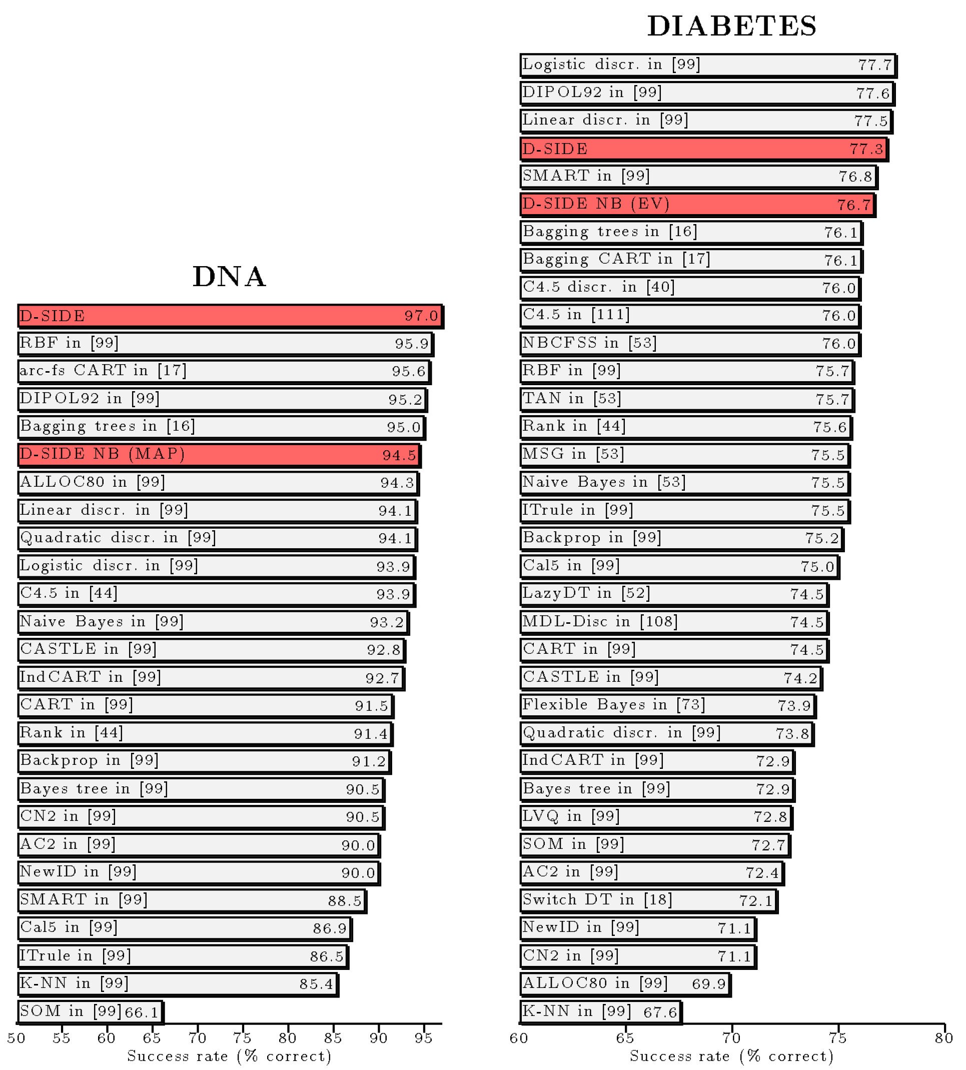


Figure 6.2: Experimental results on the DNA and the Diabetes data sets.

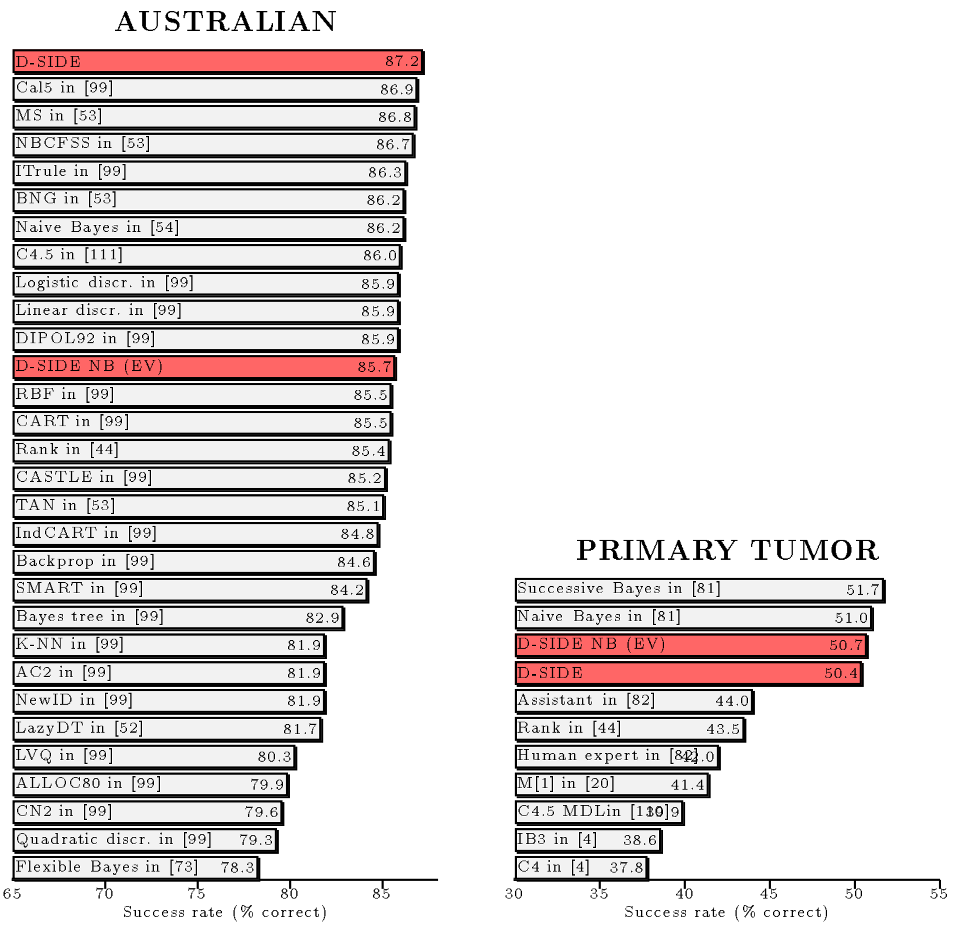


Figure 6.3: Experimental results on the Australian and the Primary Tumor data sets.

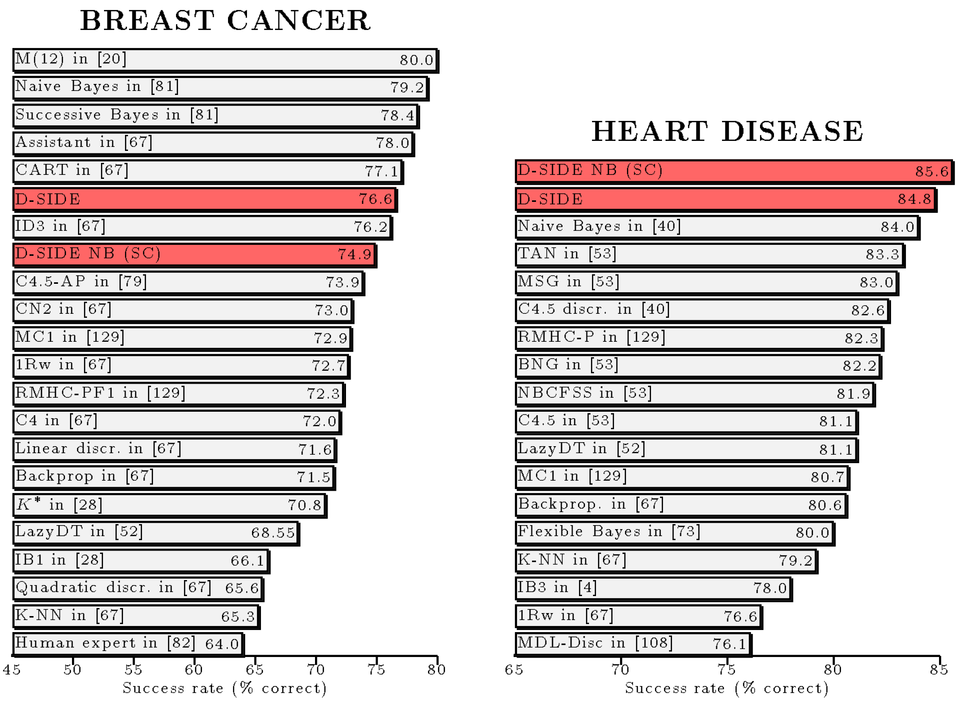


Figure 6.4: Experimental results on the Breast cancer and Heart Disease data sets.

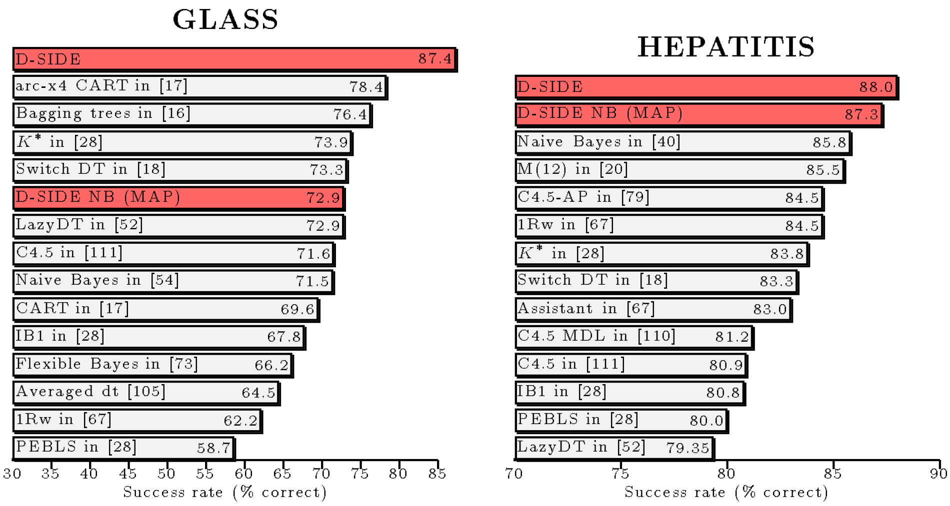


Figure 6.5: Experimental results on the Glass and Hepatitis data sets.

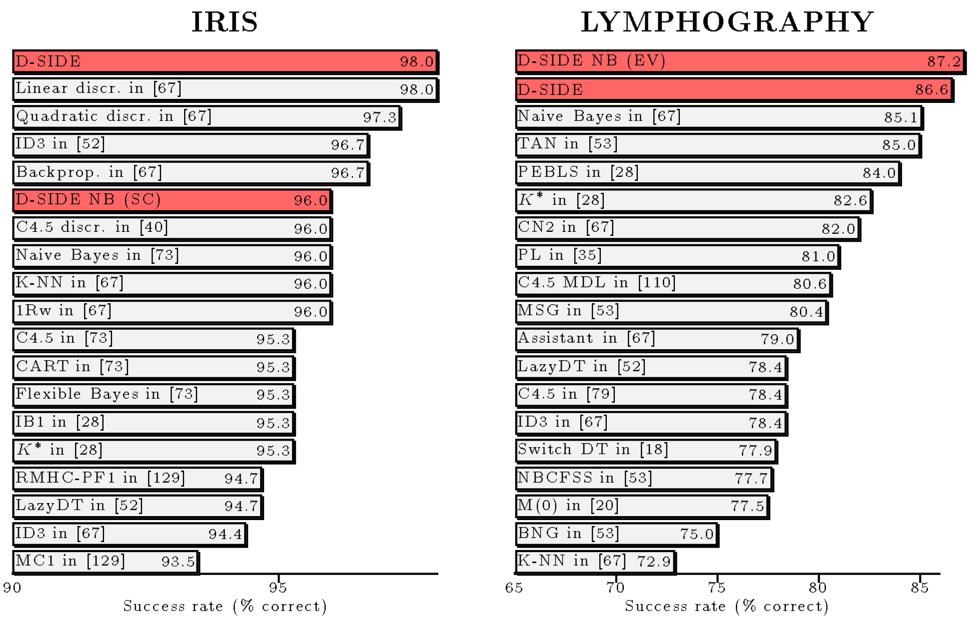


Figure 6.6: Experimental results on the Iris and Lymphography data sets.

Chapter 7

Conclusion

"Good ... morning ... Doctor ... Chandra ... This ... is ... HAL ... I ... am ... ready ... for ... my ... first lesson ... today ... "

—Arthur C. Clarke and Stanley Kubrick in 2001: A Space Odyssey

We have now completed `HAL`'s design, and demonstrated its performance with empirical experiments. It's time to reflect on what we have achieved, and what issues remain open.

`HAL` in perspective. The first interesting question is, would `HAL`'s Bayesian plausible predictions be considered "intelligent" in the sense of the two definitions given in Chapter 1? Let us look at the descriptive definition of computational intelligence [10].

A more careful inspection at Bezdek's definition reveals that it has actually two different types of requirements. The definition begins with very general statements and then gives a list of qualitative and quantitative requirements `HAL` should satisfy in a nontrivial task. It is quite debatable, whether `HAL`, or any similar predictive system, really can conform to the general requirements given. Does `HAL` "deal only with numerical (low-level) data" when it manipulates plausibilities represented as real numbers (Desideratum 2.1), or does it have "a pattern recognition component" if it is capable of classifying a given data vector with respect to a set of predefined classes? In fact our choice for `HAL`'s model family allowed `HAL` to perform also clustering, i.e., constructing predictive concepts for the problem domain from the data available. In this sense `HAL` can be said to recognize patterns. The meaning of the final point "does not use knowledge in the AI sense" is even more unclear. `HAL`'s design requires fixing the model family, which is structural information about the variables of interest and their independence relations represented e.g., as a graph structure. In this sense `HAL` uses knowledge typical to symbolic AI programs and thus violates the requirement. Interestingly this same argumentation holds for any neural network or fuzzy system.

Addressing `HAL`'s compatibility with the more explicit list of functional requirements for computationally intelligent system is easier:

i. **Computational adaptivity**. As discussed in Chapters 3 and 5 Bayes' theorem allows `HAL` to learn, i.e., update its model when new data arrives.

ii. **Computational fault tolerance**. This is an issue we have not explicitly addressed in this work. Related to fault tolerance, two things should be observed. `HAL`'s mixture models are inherently robust in the sense that if one or more of the mixture components in the model are not available, the prediction performance degrades gracefully, since the predictions are based on a weighted average of the components. But we should point out an even more general observation: the model averaging philosophy (Equation (3.2)) inherent in Bayesian inference for any model family is also robust in this sense. From the implementation point of view (which is probably the fault tolerance meant in the definition), we know that `HAL`'s deductive and inductive inference can be implemented as a neural network at least for some model families. Such families include mixtures of multinomial distributions and Gaussians, and layered feed-forward networks with sigmoidal units [12, 74, 96, 102].

iii. **Speed approaching human-like turnaround**. For the discrete prediction tasks `HAL` is programmed for, both deductive (prediction) and inductive inference (learning) surpass human performance. Notice that in some cases with the Naive Bayes model family we saw `HAL` demonstrating extremely fast learning rate reaching good prediction performance level with 2-3 examples from the data.

iv. **Error rates that approximate human performance**. In the StatLog project [99] some of the data set results included also human experts, which usually did not perform well when compared to the top ranking algorithms. Human performance given only the raw data can usually be outperformed easily by most approaches. This type of comparison is in most cases quite meaningless, as from `HAL`'s point of view the important issue is whether or not we can provide `HAL` with the same information that is available to a human expert.

In Chapter 1 we discussed also a second, normative definition of intelligence, which is concerned mainly with rational action. At this point it should be quite evident that the motivation for HAL's Bayesian approach to plausible reasoning HAL coincides perfectly with the definition of Russel and Norvig. In HAL's case the performance measure to be maximized is the prediction performance (with respect to 0/1-score), and HAL's principled design forces it to choose the best prediction given "the percept sequence" (data) and "built-in knowledge" (the model family and priors).

Extending HAL's design. Let us now turn to different ways of improving HAL's design. Extending HAL can be discussed at various levels of abstraction from minor "version improvements" to total "redesign".

Starting from the minor technical improvements one should observe that HAL's design is given for discrete data sets (but continuous model spaces). The Bayesian predictive framework with finite mixture model family extends naturally to continuous attributes also (see e.g., the work by Bishop [11, 12]). However, moving from discrete to continuous values does not always improve HAL's prediction performance, as additional assumptions of the form of the density have to be made (e.g., independent Gaussian distributions with appropriate conjugate priors), which can cancel out the possible improvements. Although the extension to mixtures of Gaussians is in principle straightforward, the technical details together with implementation issues are quite involved. Another technical improvement would be to allow HAL to use more sophisticated non-informative priors, such as Jeffrey's prior. This type of an extension to HAL's inference is given in [87], where prediction with p_{ev} using Jeffrey's prior is discussed in the context of the more general model family of Bayesian networks.

We have discussed HAL's Bayesian prediction in the standard classification context, where HAL's model used for classifying is first constructed by using the training data available, and each classification problem is then solved independently by using the model produced. The framework formulated in Section 3.2 could be extended by allowing multiple predictions to be made at the same time. In this *batch classification* case, all the classification problems are given simultaneously, and instead of dealing with a single query vector to be classified, HAL's task is to find a correct classification for a set of query vectors.

The batch classification problem can be viewed as a missing data problem, where the missing data consists of the correct classifications of query vectors. One could expect that HAL, when using batch classification mode, would produce better results than when it classifies the queries independently. This assumption follows from the fact that in the batch case the data available for making predictions consists not only of the original training data, but also of the set of all the query vectors, and HAL would have more data for constructing its model. There is a downside, however. In batch classification case the amount of missing data will also increase, making the missing data estimation problem more difficult than in the traditional case. Therefore it would be interesting to investigate the trade-off between the advantage of using the increased information available in the query batch, and the disadvantage of increased complexity in the search process.

One of the elegant features of the Bayesian language for inference is that it is "parameterized" by the model family—all the general principles and formulas in Chapters 2 and 3 apply when HAL's in-built mixture model family assumptions are replaced with some other assumptions. As discussed before, the general Bayesian network based design of HAL is a topic of much current research. One important motivation for this research is the need to build decision support systems, where Bayesian inference is not enough, since the systems are required to suggest actions rather than output just predictive distributions. All the work presented here with HAL is concerned with Bayesian inference, but for this decision support framework one needs to address also Bayesian decision-theoretical issues: the effect of different loss functions, the selection of proper estimators from the posterior etc. HAL's classification with 0/1-loss function is a very elementary example of this general framework.

Finally, there is still lot to be said about the intriguing relationship between the information theoretic and Bayesian data modeling approaches, which was only briefly discussed in Section 3.3. The conversion from the minimum encoding inference formulation to Bayesian inference formulation or vice versa is a nontrivial task. In contrast to the old definition that corresponded to the evidence term $p(D|M)$, the new version of stochastic complexity does not have a simple "Bayesian interpretation". A straightforward reverse transformation from the Bayesian inference to minimum encoding approach is also problematic, since the continuous parameter values have to be carefully quantized, otherwise the code-lengths obtained will lead to misleading results (see the discussion e.g., in [145]). On the other hand, the use of discrete hypothesis spaces in MML/MDL inference has it advantages in cases, where the Bayesian inference encounters infinite probability densities, circumventing which in the Bayesian framework requires multi-step decision procedures.

Regardless of any individual work (including the work at hand) in the machine learning, computational intelligence or artificial intelligence communities, the debate on the proper foundations for building intelligent systems will continue, as it should. Whether a truly "intelligent" HAL will be built on Bayesian, minimum encoding or some other principles, or built at all, remains to be seen—meanwhile we feel privileged to pursue the Bayesian perspective.

References

[1] J. Aczel. *Lectures on Functional Equations and their Applications.* Academic Press, New York, 1966.

[2] D. Aha. *A Study of Instance-Based Algorithms for Supervised Learning Tasks: Mathematical, Empirical, an Psychological Observations.* PhD thesis, University of California, Irvine, 1990.

[3] D. Aha. Editorial for a special issue on lazy learning. *Artificial Intelligence Review (to appear)*, 1997.

[4] D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. *Machine Learning*, 6:37–66, 1991.

[5] H. Akaike. Information theory and an extension of the maximum likelihood principle. In B.N. Petrox and F. Caski, editors, *Proceedings of the Second International Symposium on Information Theory*, pages 267–281, Budapest, 1973. Akademiai Kiado.

[6] C. Atkeson. Memory based approaches to approximating continuous functions. In M. Casdagli and S. Eubank, editors, *Nonlinear Modeling and Forecasting. Proceedings Volume XII in the Santa Fe Institute Studies in the Sciences of Complexity.* Addison Wesley, New York, NY, 1992.

[7] C. Atkeson, A. Moore, and S. Schaal. Locally weighted learning. *AI Review (to appear)*, 1995.

[8] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York, 1985.

[9] J.M. Bernardo and A.F.M Smith. *Bayesian Theory.* John Wiley, 1994.

[10] J. Bezdek. What is computational intelligence? In J. Zurada, R. Marks II, and C. Robinson, editors, *Computational Intelligence - Imitating Life.* IEEE Press, 1994.

[11] C. M. Bishop. Mixture density networks. Technical Report NCGR/4288, Neural Computing Research Group, Department of Computer Science, Aston University, 1994.

[12] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[13] C.M. Bishop, M. Svensén, and C.K.I. Williams. EM optimization of latent-variable density models. In D.S. Touretzky, M.C.Mozer, and M.E.Hasselmo, editors, *Advances in Neural Information Processing Systems 8.* MIT Press, 1996.

[14] C.M. Bishop, M. Svensén, and C.K.I. Williams. GTM: A principled alternative to the self-organizing map. In *Advances in Neural Information Processing Systems 9 (to appear).* MIT Press, 1997.

[15] H. Bozdogan. On the information-based measure of covariance complexity and its applications to the evaluation of multivariate linear models. *Communications in Statistics - Theory and Methods*, 19(1):221–278, 1990.

[16] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.

[17] L. Breiman. Bias, variance, and arcing classifiers. Technical Report TR 460, University of California, Statistics Department, April 1996.

[18] C.E. Brodley. Automatic selection of split criterion during tree growing based on node location. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 73–80. Morgan Kaufmann Publishers, 1995.

[19] A.E. Bryson and Y.-C. Ho. *Applied Optimal Control.* Blaisdell, New York, 1969.

[20] B. Cestnik and I. Bratko. On estimating probabilities in tree pruning. In Y. Kodratoff, editor, *Machine Learning EWSL-91*, pages 138–150. Springer-Verlag, 1991.

[21] P. Cheeseman. Probabilistic versus fuzzy reasoning. In L.N. Kanal and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 1*, pages 85–102. Elsevier Science Publishers B.V. (North-Holland), Amsterdam, 1986.

[22] P. Cheeseman. Inquiry into computer understanding + discussions and rebuttal. *Computational Intelligence*, 4:57–142, 1988.

[23] P. Cheeseman, J. Kelly, M. Self, J. Stutz, W. Taylor, and D. Freeman. AutoClass: A Bayesian classification system. In *Proceedings of the Fifth International Conference on Machine Learning*, pages 54–64, Ann Arbor, June 1988.

[24] P. Cheeseman and J. Stutz. Bayesian classification (AutoClass): Theory and results. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, chapter 6. AAAI Press, Menlo Park, 1996.

[25] D.M. Chickering and D. Heckerman. Efficient approximations for the marginal likelihood of incomplete data given a Bayesian network. In E. Horvitz and F. Jensen, editors, *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*, pages 158–168, Portland, Oregon, August 1996. Morgan Kaufmann Publishers.

[26] G. Chow. A comparison of the information and posterior probability criteria for model selection. *Journal of Econometrics*, 16:21–33, 1981.

[27] W. Clancey, S.W. Smoliar, and M.J. Stefik, editors. *Contemplating Minds*. MIT Press, Cambridge, Massachusetts, 1994.

[28] J.G. Cleary and L.E. Trigg. k^*: An instance-based learner using and entropic distance measure. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 108–114. Morgan Kaufmann Publishers, 1995.

[29] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[30] G.F. Cooper. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, March 1990.

[31] T. Cover and J. Thomas. *Elements of Information Theory*. John Wiley & Sons, New York, NY, 1991.

[32] R. Cox. Probability, frequency, and reasonable expectation. *Am. Jour. Phys.*, 14:1–13, 1946.

[33] R. Cox. *The Algebra of Probable Inference*. Johns Hopkins University Press, Baltimore, MD, 1961.

[34] P. Dagum and M. Luby. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60:141–153, 1993.

[35] P. Datta and D. Kibler. Learning prototypical concept descriptions. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 158–166. Morgan Kaufmann Publishers, 1995.

[36] B. de Finetti. *Theory of Probability.* John Wiley & Sons, New York, 1970.

[37] M.H. DeGroot. *Optimal statistical decisions.* McGraw-Hill, 1970.

[38] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[39] P. Domingos and M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 105–112. Morgan Kaufmann Publishers, 1996.

[40] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 194–202. Morgan Kaufmann Publishers, 1995.

[41] H.L. Dreyfus. *What Computers Can't Do: The Limits of Artificial Intelligence.* Harper and Row, New York, revised edition, 1979.

[42] H.L. Dreyfus. *What Computers Still Can't Do: A Critique of Artificial Reason.* MIT Press, Cambridge, Massachusetts, 1992.

[43] R.O. Duda and P.E. Hart. *Pattern classification and scene analysis.* John Wiley, 1973.

[44] T. Elomaa. *Tools and Techniques for Decision Tree Learning.* PhD thesis, Report A-1996-2, Department of Computer Science, University of Helsinki, May 1996.

[45] T. Elomaa and J. Rousu. General and efficient multisplitting of numerical attributes. Technical Report C-1996-82, University of Helsinki, Department of Computer Science, 1996.

[46] B.S. Everitt and D.J. Hand. *Finite Mixture Distributions.* Chapman and Hall, London, 1981.

[47] D. Fischer. Knowledge acquisition via incremental conceptual clustering. *Machine Learning*, 2:139–172, 1987.

[48] D. Fischer. Noise-tolerant conceptual clustering. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 825–830, Detroit, Michigan, 1989.

[49] D. Fischer and D. Talbert. Inference using probabilistic concept trees. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 191–202, Ft. Lauderdale, Florida, January 1997.

[50] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In P. Vitányi, editor, *Computational Learning Theory. Second European Conference (EuroCOLT'95)*, pages 23–37. Springer-Verlag, 1995.

[51] J.H. Friedman. Flexible metric nearest neighbor classification. Unpublished manuscript. Available by anonymous ftp from Stanford Research Institute (Menlo Park, CA) at playfair.stanford.edu, 1994.

[52] J.H. Friedman, R. Kohavi, and Y. Yun. Lazy decision trees. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 717–724, August, 1996. AAAI Press/MIT Press.

[53] N. Friedman and M. Goldszmidt. Building classifiers using Bayesian networks. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1277–1284, August, 1996. AAAI Press/MIT Press.

[54] N. Friedman and M. Goldszmidt. Discretizing continuous attributes while learning Bayesian networks. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 157–165. Morgan Kaufmann Publishers, 1996.

[55] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis.* Chapman & Hall, 1995.

[56] W. R. Gilks, S. Richardson, and Spiegelhalter D. J. *Markov chain Monte Carlo in practice.* Chapman & Hall, London, GB, 1996.

[57] K. Göedel. Über formal unentscheidbare sätze der Principia Mathematica und verwandter systeme, I. *Monats. Math. Phys.*, 38:173–198, 1931.

[58] I. Good, editor. *Probability and the Weighting of Evidence.* Hafners, New York, 1950.

[59] M. Gyllenberg, H.G. Gyllenberg, T. Koski, and J. Schindler. Non-uniqueness of numerical taxonomic structures. *Binary*, 5:138–144, 1993.

[60] M. Gyllenberg, T. Koski, and M. Verlaan. Classification of binary vectors by stochastic complexity. Technical Report A5, University of Turku, Institute for Applied Mathematics, November 1994.

[61] J. Halpern. A counterexample to Theorems of Cox and Fine. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1313–1319, August, 1996. AAAI Press/MIT Press.

[62] J. Haugeland, editor. *Mind Design.* MIT Press, Cambridge, Massachusetts, 1981.

[63] S. Haykin. *Neural Networks: A Comprehensive Foundation.* IEEE Press/Macmillan College Publishing Company, New York, 1994.

[64] D. Heckerman. A tutorial on learning with Bayesian networks. Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, One Microsoft Way, Redmond, WA 98052, 1996.

[65] D. Heckerman and D. Chickering. A comparison of scientific and engineering criteria for Bayesian model selection. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 275–281, Ft. Lauderdale, Florida, January 1997.

[66] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, September 1995.

[67] R.C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–91, 1993.

[68] E. Horvitz, D. Heckerman, and C. Langlotz. A framework for comparing alternative formalisms for plausible reasoning. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pages 210–214, Philadelphia, PA, August 1986. Morgan Kaufmann Publishers.

[69] E. Jaynes. Probability theory as logic. In P.Fougère, editor, *Maximum Entropy and Bayesian Methods*, pages 369–374. Kluwer Academic Publishers, 1990.

[70] E. Jaynes. Probability theory: The logic of science. 1996.

[71] H. Jeffreys. *Theory of Probability.* Clarendon Press, Oxford, 1939.

[72] H. Jeffreys. An invariant form for the prior probability in estimation problems. *Proc. Roy. Soc. A*, 186:453–461, 1946.

[73] G.H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In P. Besnard and S. Hanks, editors, *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann Publishers, 1995.

[74] M.I. Jordan and R.A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6:181–214, 1994.

[75] G. Kalkanis and G. Conroy. Principles of induction and approaches to attribute based induction. *The Knowl. Eng. Rev.*, 6:307–333, 1991.

[76] S. Kasif, S. Salzberg, D. Waltz, J. Rachlin, and D. Aha. Towards a framework for memory-based reasoning. Manuscript, in review, 1995.

[77] R.E. Kass and A.E. Raftery. Bayes factors. Technical Report 254, Department of Statistics, University of Washington, 1994.

[78] H. Kitano. Challenges of massive parallelism. In *Proc. of IJCAI-93, the Thirteenth International Joint Conference on Artificial Intelligence*, pages 813–834, Chambéry, France, August 1993. Morgan Kaufmann Publishers.

[79] R. Kohavi and G.H. John. Automatic parameter selection by minimizing estimated error. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 304–312. Morgan Kaufmann Publishers, 1995.

[80] T. Kohonen. *Self-Organizing Maps.* Springer-Verlag, Berlin, 1995.

[81] I. Kononenko. Successive naive Bayesian classifier. *Informatica*, 17:167–174, 1993.

[82] I. Kononenko and I. Bratko. Information-based evaluation criterion for classifier's performance. *Machine Learning*, 6:67–80, 1991.

[83] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Bayesian discretization by clustering. In *Proceedings of the European Symposium on Intelligent Techniques*, pages 265–268, Bari, Italy, March 1997.

[84] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. Comparing stochastic complexity minimization algorithms in estimating missing data. In *Proceedings of WUPES'97, the 4th Workshop on Uncertainty Processing*, pages 81–90, Prague, Czech Republic, January 1997.

[85] P. Kontkanen, P. Myllymäki, T. Silander, and H. Tirri. On the accuracy of stochastic complexity approximations. In *Proceedings of the Causal Models and Statistical Learning Seminar*, pages 103–117, London, UK, March 1997.

[86] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. Comparing predictive inference methods for discrete domains. In *Proceedings of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 311–318, Ft. Lauderdale, Florida, January 1997.

[87] P. Kontkanen, P. Myllymäki, T. Silander, H. Tirri, and P. Grünwald. On predictive distributions and Bayesian networks. 1997.

[88] P. Kontkanen, P. Myllymäki, and H. Tirri. Comparing Bayesian model class selection criteria by discrete finite mixtures. In D. Dowe, K. Korb, and J. Oliver, editors, *Information, Statistics and Induction in Science*, pages 364–374, Proceedings of the ISIS'96 Conference, Melbourne, Australia, August 1996. World Scientific, Singapore.

[89] P. Kontkanen, P. Myllymäki, and H. Tirri. Constructing Bayesian finite mixture models by the EM algorithm. Technical Report NC-TR-97-003, ESPRIT Working Group 8556: Neural and Computational Learning (NeuroCOLT), 1996.

[90] P. Kontkanen, P. Myllymäki, and H. Tirri. Some experimental results with finite mixture models. In *Book of Abstracts, First European Conference on Highly Structured Stochastic Systems*, pages 112–115, Rebild, Denmark, May 1996.

[91] P. Kontkanen, P. Myllymäki, and H. Tirri. Experimenting with the Cheeseman-Stutz evidence approximation for predictive modeling and data mining. In *Proceedings of the Tenth International FLAIRS Conference (to appear)*, Daytona Beach, Florida, May 1997.

[92] B. Kosko. *Neural Networks and Fuzzy Systems — A Dynamical Systems Approach To Machine Intelligence.* Prentice-Hall, New Jersey, USA, 1992.

[93] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data.* Wiley, 1987.

[94] T.J.. Loredo. From Laplace to supernova SN 1987A: Bayesian inference in astrophysics. In P.Fougère, editor, *Maximum Entropy and Bayesian Methods*, pages 81–142. Kluwer Academic Publishers, 1990.

[95] D. Mackay. *Bayesian Methods for Adaptive Models.* PhD thesis, California Institute of Technology, 1992.

[96] D. Mackay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.

[97] D. McDermott and J. Doyle. Non-monotonic logic I. *Artificial intelligence*, 13(1,2):41–72, 1980.

[98] Jordan M.I. and C.M. Bishop. Neural networks. A.I.Memo 1562, MIT, Artificial Intelligence Laboratory, 1996.

[99] D. Michie, D.J. Spiegelhalter, and C.C. Taylor, editors. *Machine Learning, Neural and Statistical Classification.* Ellis Horwood, London, 1994.

[100] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[101] A. Moore. Acquisition of dynamic control knowledge for a robotic manipulator. In *Seventh International Machine Learning Workshop.* Morgan Kaufmann, 1990.

[102] P. Myllymäki and H. Tirri. Bayesian case-based reasoning with neural networks. In *Proceedings of the IEEE International Conference on Neural Networks*, volume 1, pages 422–427, San Francisco, March 1993. IEEE, Piscataway, NJ.

[103] P. Myllymäki and H. Tirri. Massively parallel case-based reasoning with probabilistic similarity metrics. In S. Wess, K.-D. Althoff, and M. Richter, editors, *Topics in Case-Based Reasoning*, volume 837 of *Lecture Notes in Artificial Intelligence*, pages 144–154. Springer-Verlag, 1994.

[104] P. Myllymäki and H. Tirri. Constructing computationally efficient Bayesian models via unsupervised clustering. In A.Gammerman, editor, *Probabilistic Reasoning and Bayesian Belief Networks*, pages 237–248. Alfred Waller Publishers, Suffolk, 1995.

[105] J.O. Oliver and D.J. Hand. On pruning and averaging decision trees. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 430–437. Morgan Kaufmann Publishers, 1995.

[106] J. Paris. *The Uncertain Reasoner's Companion: A Mathematical Perspective*, volume 39 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1994.

[107] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, San Mateo, CA, 1988.

[108] B. Pfahringer. Compression-based discretization of continuous attributes. In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 456–463. Morgan Kaufmann Publishers, 1995.

[109] G. Polya. *Mathematics and plausible reasoning*. 2 vols., Princeton University Press, New Jersey, 1954.

[110] J.R. Quinlan. MDL and categorical theories (continued). In A. Prieditis, editor, *Machine Learning: Proceedings of the Twelfth International Conference*, pages 464–470. Morgan Kaufmann Publishers, 1995.

[111] J.R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.

[112] A. Raftery. Approximate Bayes factors and accounting for model uncertainty in generalized linear models. Technical Report 255, Department of Statistics, University of Washington, 1993.

[113] F. Ramsey, editor. *The Foundations of Mathematics and Other Logical Essays*. Routledge and Kegan Paul, London, 1931.

[114] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:445–471, 1978.

[115] J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society*, 49(3):223–239 and 252–265, 1987.

[116] J. Rissanen. *Stochastic Complexity in Statistical Inquiry.* World Scientific Publishing Company, New Jersey, 1989.

[117] J. Rissanen. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, 42(1):40–47, January 1996.

[118] R.D. Rosenkranz, editor. *E.T.Jaynes: papers on Probability, Statistics, and Statistical Physics.* D. Reidel Publishing Company, Dordrecht, Holland, 1983.

[119] S. Rosenkranz. *The Bayes factors for model evaluation in hierarchical Poisson model for area counts.* PhD thesis, Department of Biostatistics, University of Washington, 1992.

[120] D. Roth. On the hardness of approximate reasoning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, volume 1, pages 613–618, 1993.

[121] S.J. Russel and P. Norvig. *Artificial Intelligence: A Modern Approach.* Prentice Hall, New Jersey, 1995.

[122] L. Savage. *Foundations of Statistics.* Dover Publications, New York, 1954.

[123] G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

[124] D.W. Scott. *Multivariate Density Estimation. Theory, Practice, and Visualization.* John Wiley & Sons, New York, 1992.

[125] J.R. Searle. *Minds, Brains and Science.* Harvard University Press, Cambridge, Massachusetts, 1984.

[126] J.R. Searle. *The Rediscovery of the Mind.* MIT Press, Cambridge, Massachusetts, 1992.

[127] G Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.

[128] J. Shore and R. Johnson. Axiomatic derivation of the principle of maximum entropy and the principle of minimum cross-entropy. *IEEE Trans. Information Theory*, IT-26:26–37, 1980.

[129] D. Skalak. Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Machine Learning: Proceedings of the Eleventh International Conference*, pages 293–301, 1994.

[130] R. Solomonoff. A formal theory of inductive inference. *Information and Control*, 7:1–22,224–254, 1964.

[131] D. Specht. Probabilistic neural networks. *Neural Networks*, 3:109–118, 1990.

[132] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, 1986.

[133] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society (Series B)*, 36:111–147, 1974.

[134] L. Tierney and J. Kadane. Accurate approximations for posterior moments and marginal densities. *J. Amer. Statist. Ass.*, 81:82–86, 1986.

[135] H. Tirri, P. Kontkanen, and P. Myllymäki. A Bayesian framework for case-based reasoning. In I. Smith and B. Faltings, editors, *Advances in Case-Based Reasoning*, volume 1168 of *Lecture Notes in Artificial Intelligence*, pages 413–427, Proceedings of the 3rd European Workshop, Lausanne, Switzerland, November 1996. Springer-Verlag, Berlin Heidelberg.

[136] H. Tirri, P. Kontkanen, and P. Myllymäki. Probabilistic instance-based learning. In L. Saitta, editor, *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 507–515. Morgan Kaufmann Publishers, 1996.

[137] D. Titterington. Updating a diagnostic system using unconfirmed cases. *Applied Statistics*, 25:238–247, 1976.

[138] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical Analysis of Finite Mixture Distributions.* John Wiley & Sons, New York, 1985.

[139] M. Tribus. *Rational Descriptions, Decisions and Designs.* Pergamon Press, New York, 1969.

[140] A.M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, Ser.2-42:230–265, 1936.

[141] A.M. Turing. Computing machinery and intelligence. *Mind*, 59:433–460, 1950.

[142] A. Utsugi. Hyperparameter selection for self-organizing maps. *Neural Computation (to appear)*, 9, 1997.

[143] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[144] C.S. Wallace and D.M Boulton. An information measure for classifiation. *Computer Journal*, 11:185–194, 1968.

[145] C.S. Wallace and P.R. Freeman. Estimation and inference by compact coding. *Journal of the Royal Statistical Society*, 49(3):240–265, 1987.

[146] S. Waterhouse, D. MacKay, and T. Robinson. Bayesian methods for mixtures of experts. In D.S. Touretzky, M.C.Mozer, and M.E.Hasselmo, editors, *Advances in Neural Information Processing Systems 8*. MIT Press, 1996.

[147] S.S Wilks. *Mathematical Statistics*. John Wiley, 1962.

[148] L.A. Zadeh. Fuzzy logic and approximate reasoning. *Synthese*, 30:407–425, 1975.

[149] L.A. Zadeh. Possibility theory and soft data analysis. In L. Cobb and R. M. Thrall, editors, *Mathematical frontiers of the social and policy sciences*, pages 69–129. Westview Press, Boulder, CO, 1981.

Appendix A

Data sets used in the experiments

This Appendix presents the description of all the public domain data sets used in the experiments. The selection of datasets in the comparison was done on the basis of their reported use, i.e., we have preferred datasets that have been used for testing many different methods over datasets with only isolated results. Consequently many of the data sets have been used in the StatLog project [99], but we have also included other frequently used data sets. The descriptions of the data sets and the testing procedures used for each data set are given in Table A.1[1]. The last column, the default value, denotes the success rate of a simple classifier, which classifies all the instances to the most common class. A more detailed description of these data sets can be found in [99] and in the documentation in the UCI data repository.

[1]The data sets can be obtained from the UCI data repository at URL "http://www.ics.uci.edu/~mlearn/".

Data set	Size	#Attrs	#Classes	Test method	Default
DNA	3186	181	3	train&test	50.8
Diabetes	768	9	2	12-fold CV	65.0
Australian	690	15	2	10-fold CV	56.0
Primary Tumor	339	18	21	10-fold CV	24.8
Breast Cancer	286	10	2	11-fold CV	70.3
Heart Disease	270	14	2	9-fold CV	79.4
Glass	214	10	6	7-fold CV	40.7
Hepatitis	150	20	2	5-fold CV	55.6
Iris	150	5	3	5-fold CV	33.3
Lymphography	148	19	4	5-fold CV	54.7

Table A.1: The public domain data sets and the number of cross-validation folds used in our experiments.

Appendix B

Performance of the different predictive distributions

This Appendix presents the results of the comparison of `HAL`'s three alternative prediction methods discussed in Section 4.4. The results for all the five data sets (Australian, Hepatitis, Glass, Primary Tumor and Heart disease) are presented in Figures B.3–B.11. For a brief description of the data sets see Appendix A.

Figures B.3–B.2 give the mean prediction performance of $p_{\text{map}}, p_{\text{ev}}$, and p_{sc} in the test set calculated by averaging the results from 100 partitions of the data (maximum training set size 70 % and test set size 30 %). Each figure shows the performance of the methods for both log-score and 0/1-score.

Figures B.12–B.11 give the maximum and the minimum prediction performance of $p_{\text{map}}, p_{\text{ev}}$, and p_{sc} in the 100 partitions described above. Each figure shows both the log-score and 0/1-score performance.

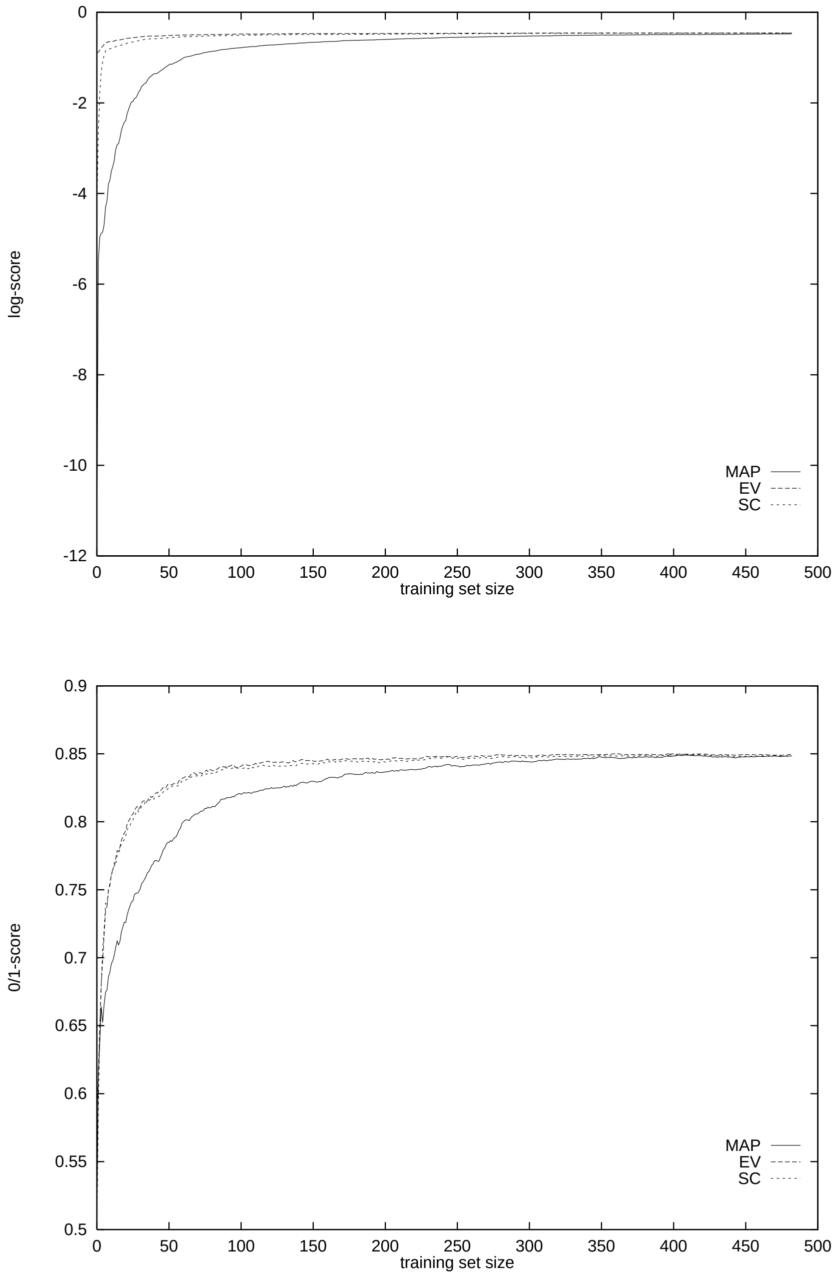
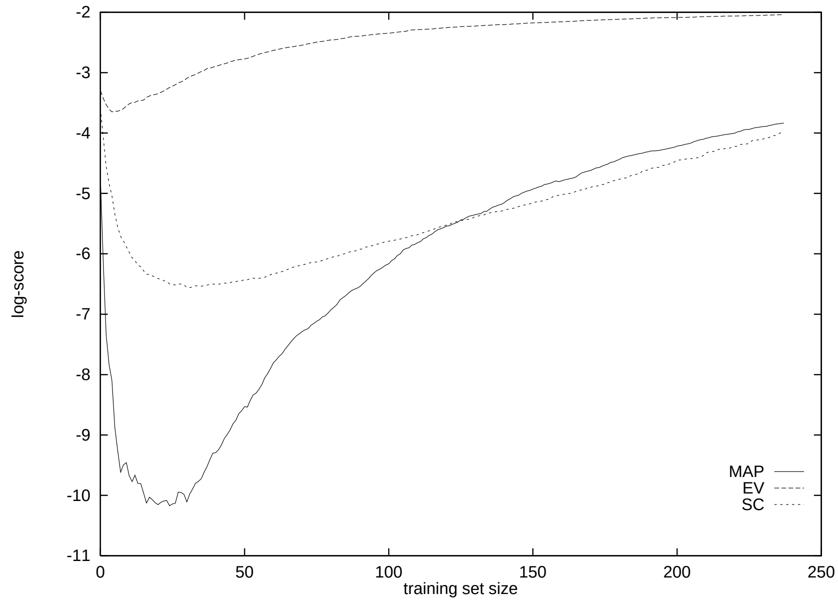


Figure B.1: Average performance of methods by log-score (up) and 0/1-score (below) for the Australian data set.



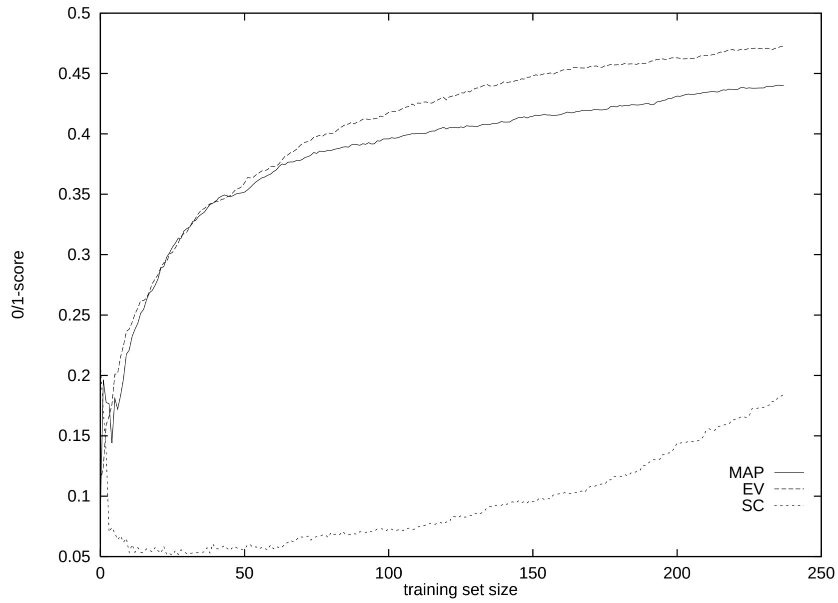
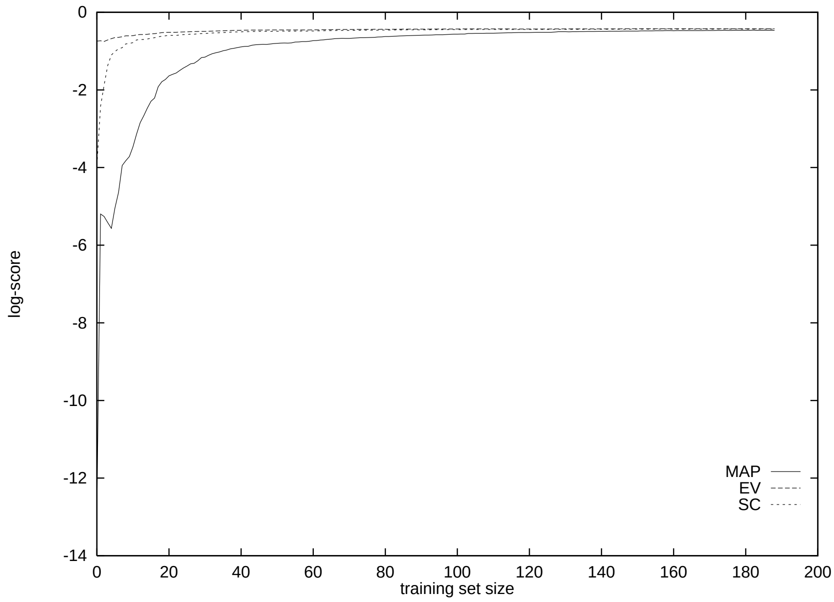


Figure B.2: Average performance of methods by log-score (up) and 0/1-score (below) for the Primary Tumor data set.



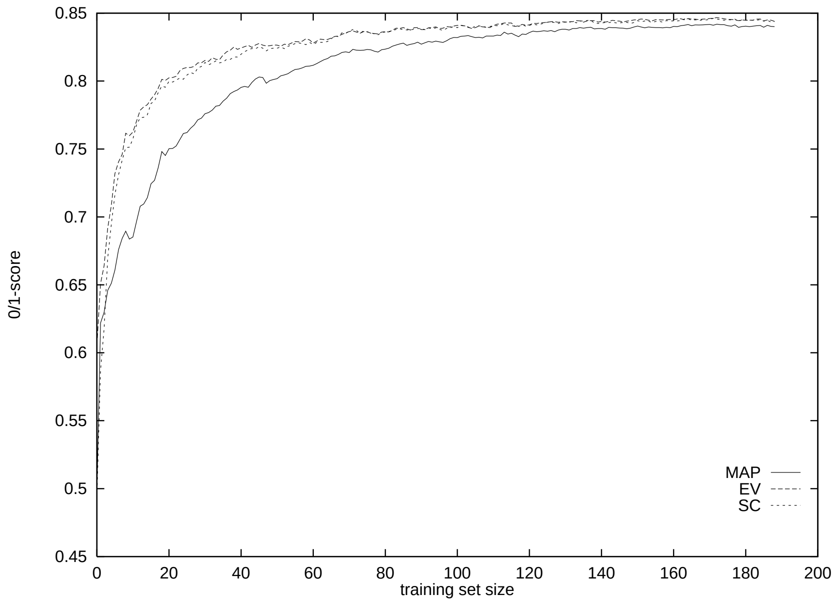


Figure B.3: Average performance of methods by log-score (up) and 0/1-score (below) for the Heart Disease data set.

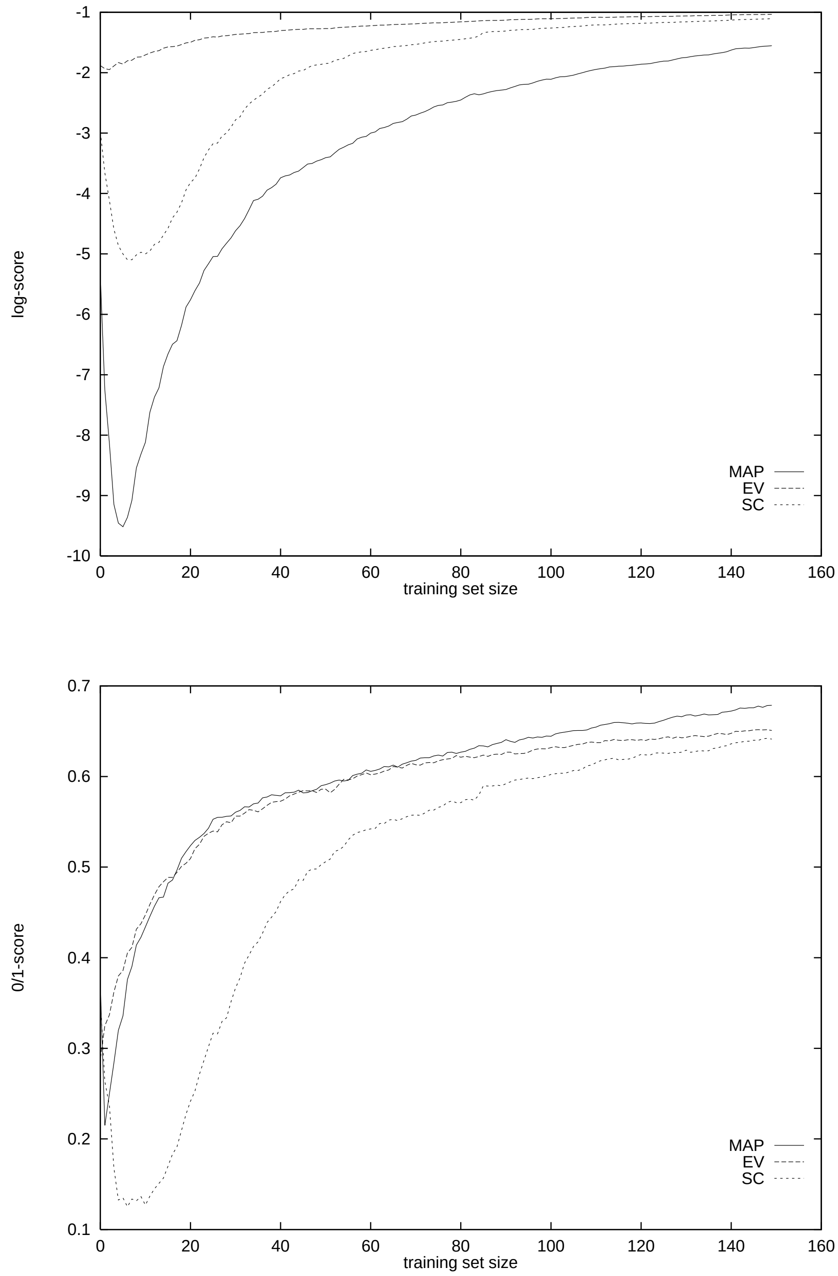
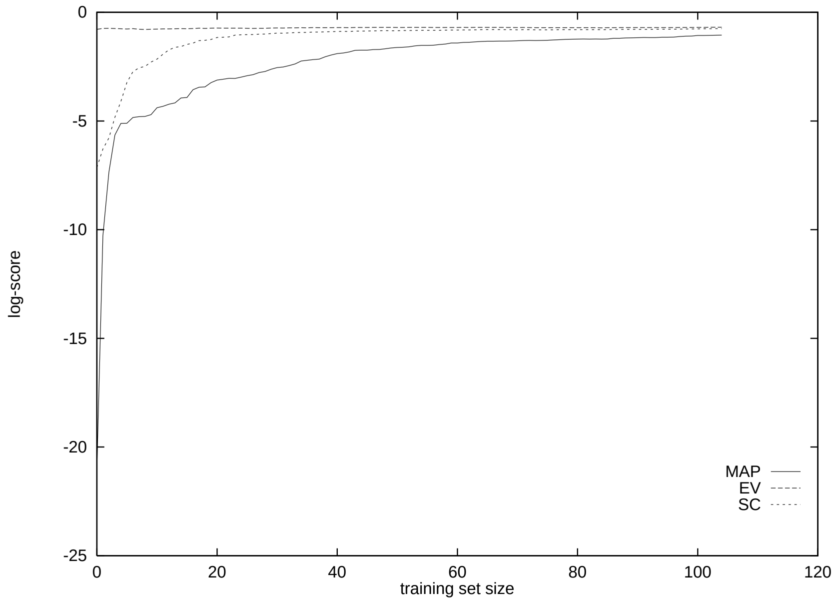


Figure B.4: Average performance of methods by log-score (up) and 0/1-score (below) for the Glass data set.



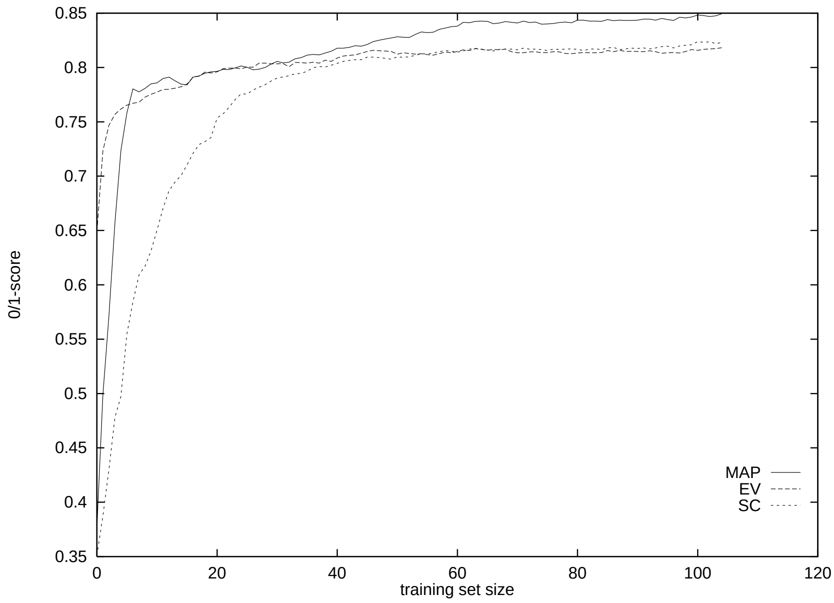


Figure B.5: Average performance of methods by log-score (up) and 0/1-score (below) for the Hepatitis data set.

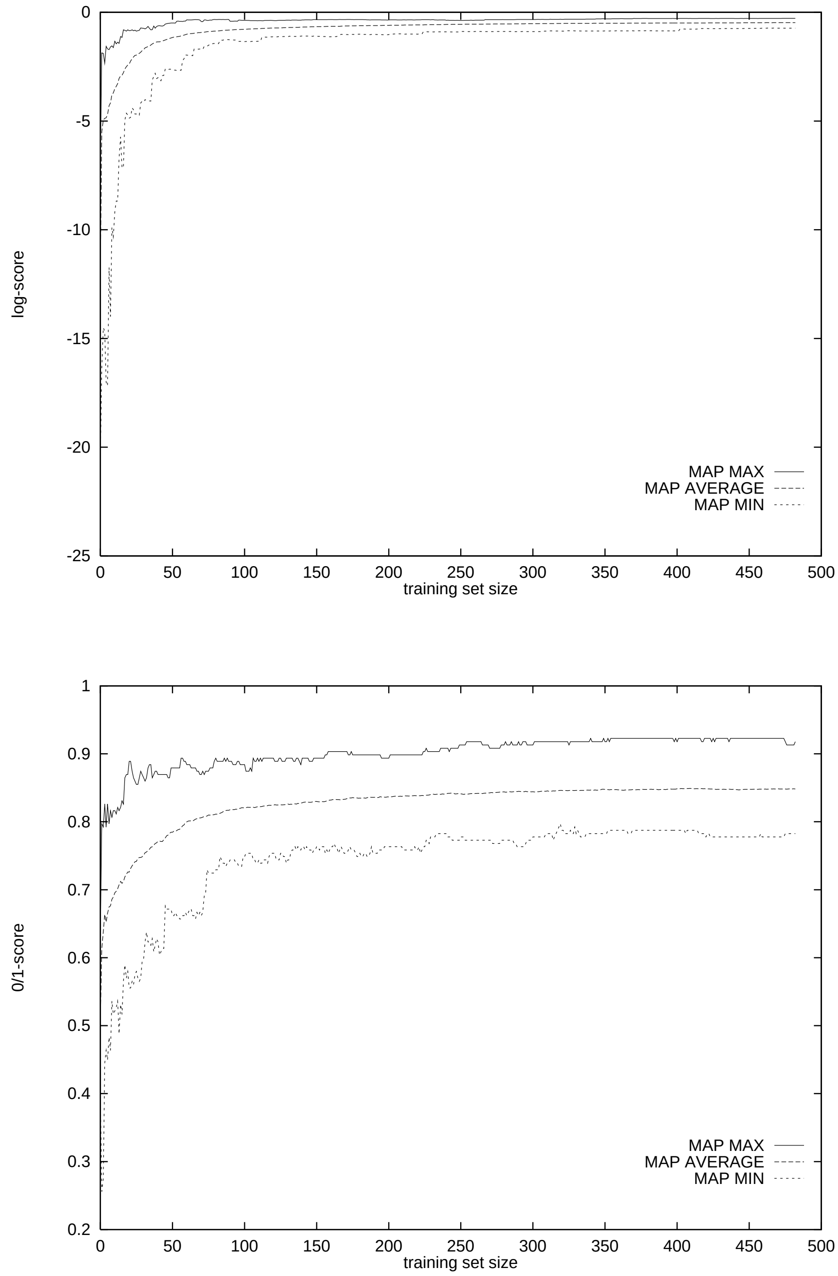
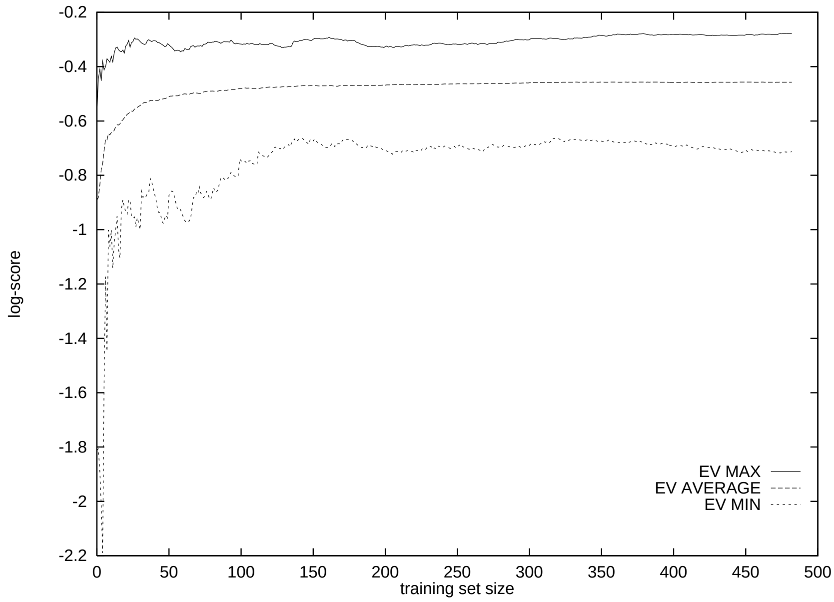


Figure B.6: The maximum and minimum performance of the MAP predictive distribution method by log-score (up) and 0/1-score (below) for the Australian data set.



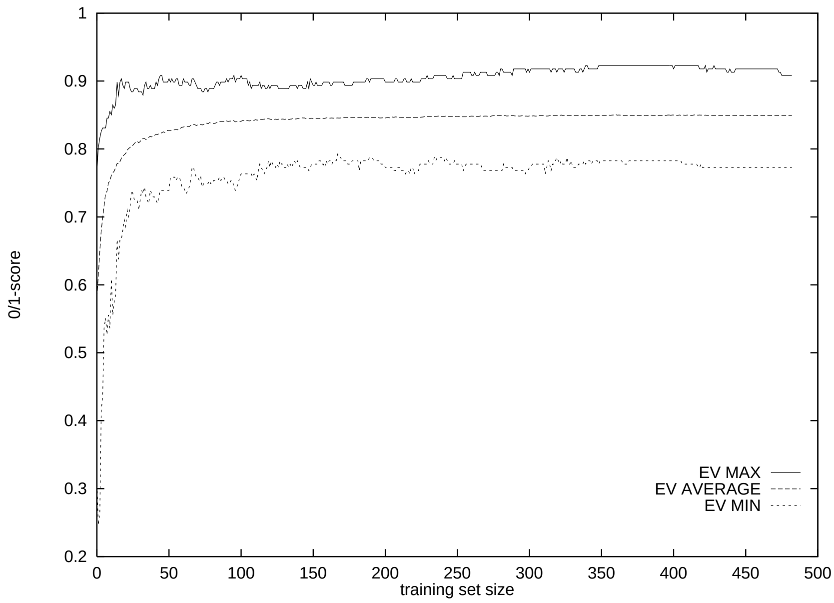
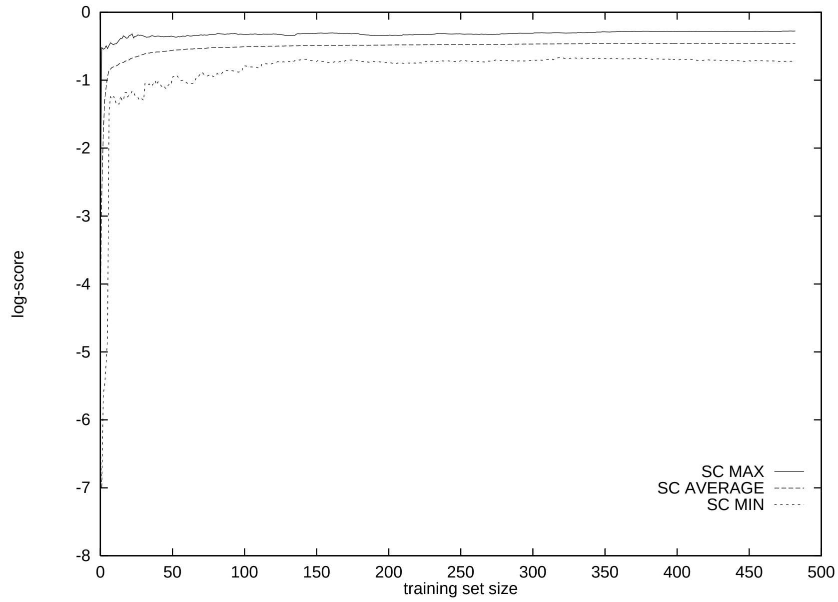


Figure B.7: The maximum and minimum performance of the evidence predictive distribution method by log-score (up) and 0/1-score (below) for the Australian data set.



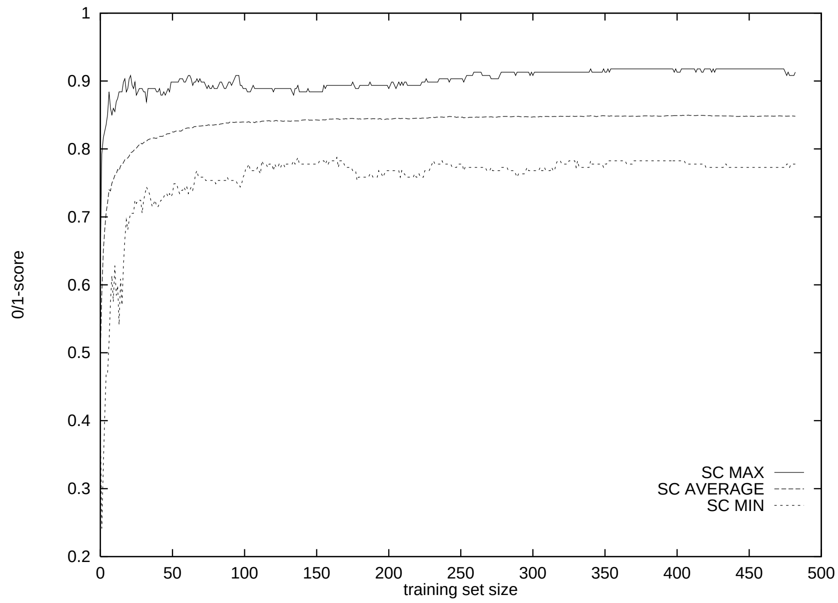
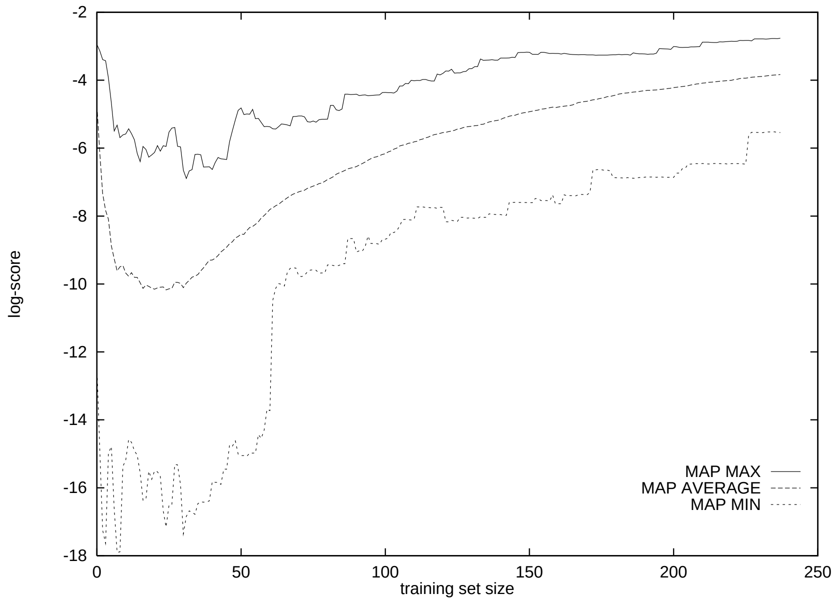


Figure B.8: The maximum and minimum performance of the stochastic complexity predictive distribution method by log-score (up) and 0/1-score (below) for the Australian data set.



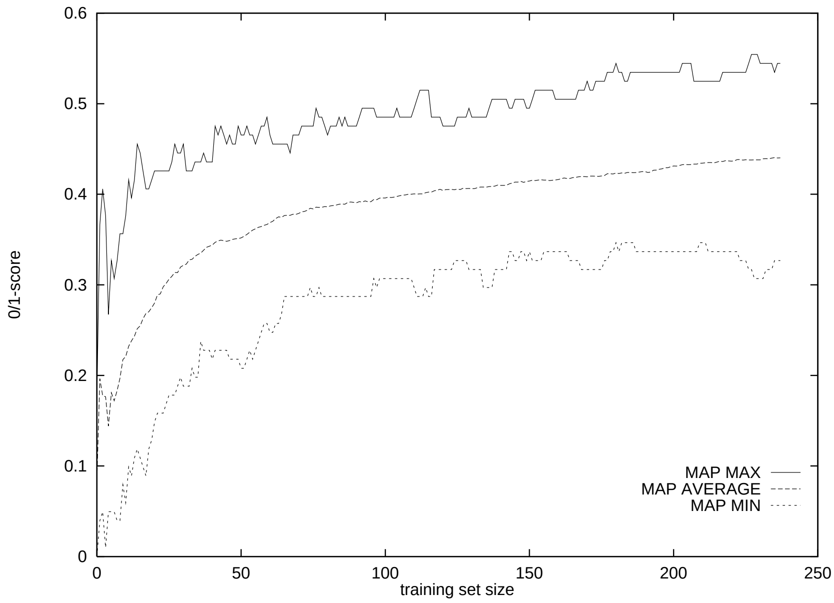
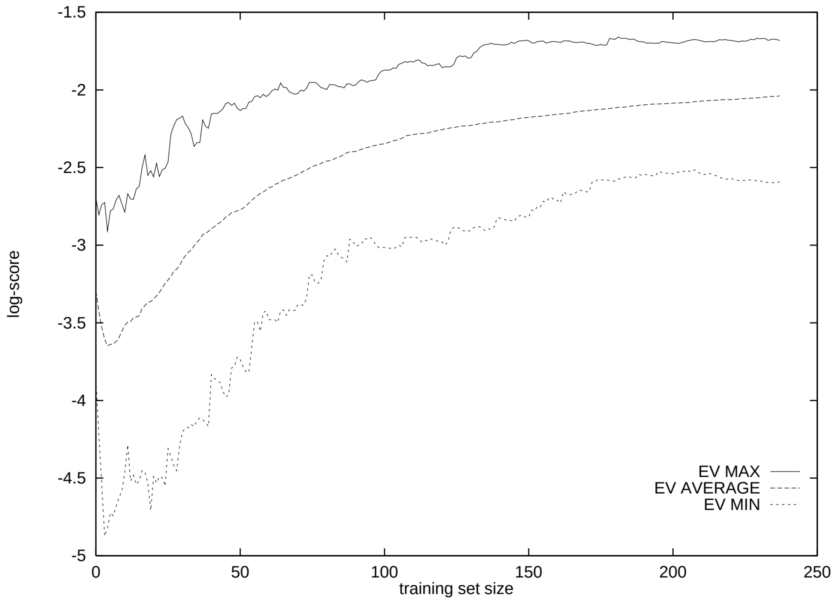


Figure B.9: The maximum and minimum performance of the MAP predictive distribution method by log-score (up) and 0/1-score (below) for the Primary Tumor data set.



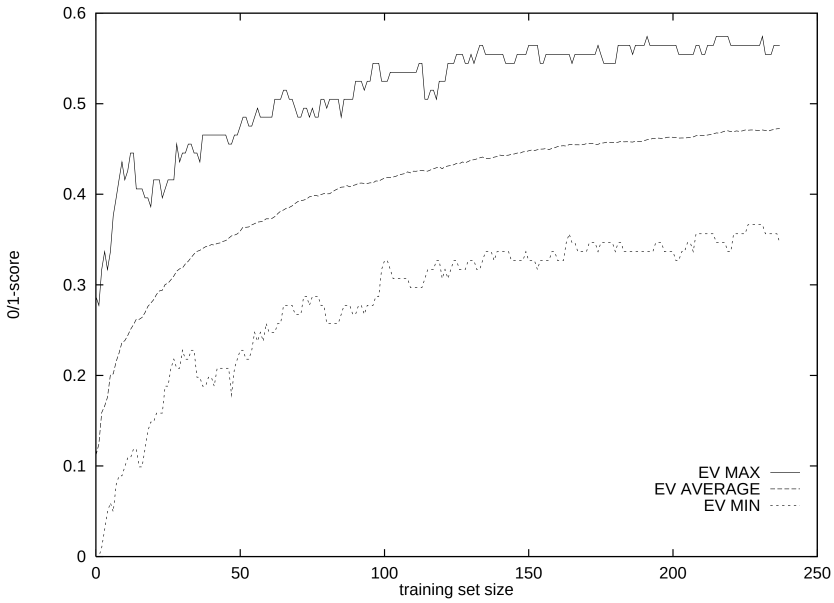
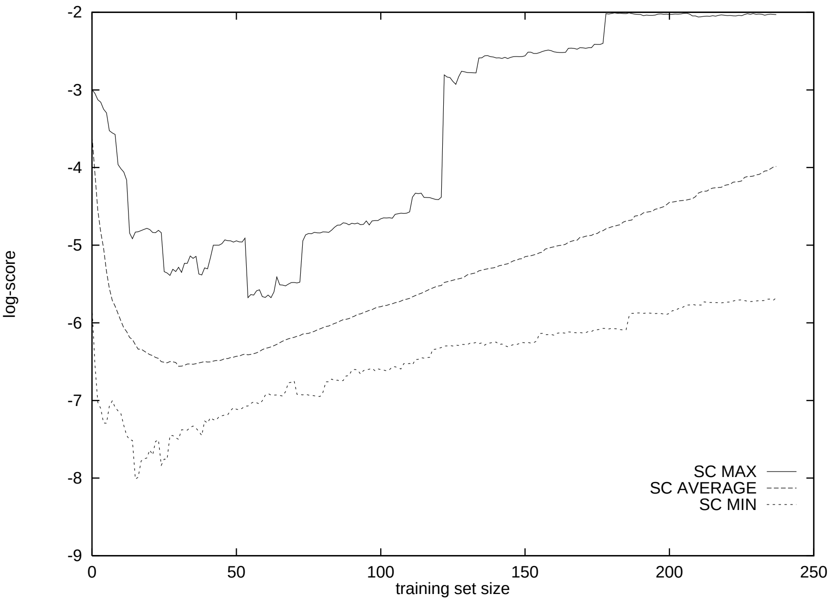


Figure B.10: The maximum and minimum performance of the evidence predictive distribution method by log-score (up) and 0/1-score (below) for the Primary Tumor data set.



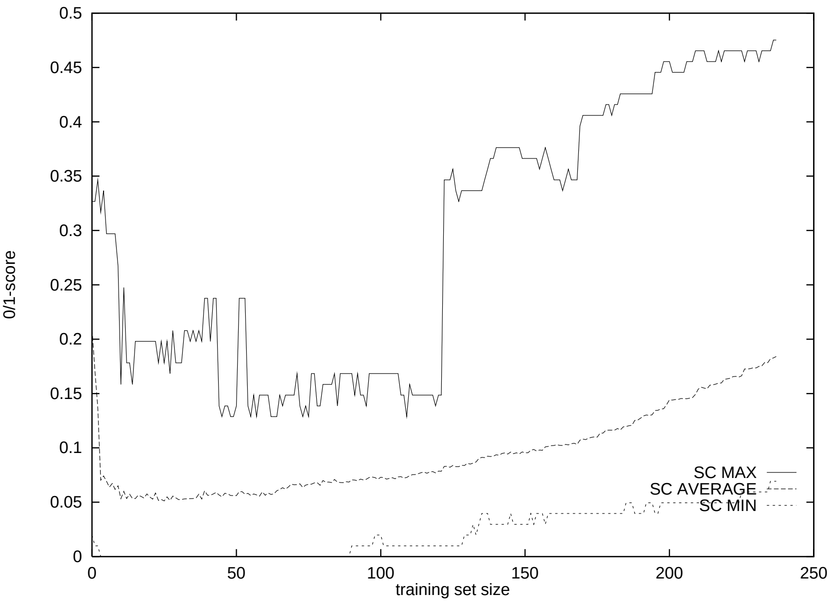


Figure B.11: The maximum and minimum performance of the stochastic complexity predictive distribution method by log-score (up) and 0/1-score (below) for the Primary Tumor data set.

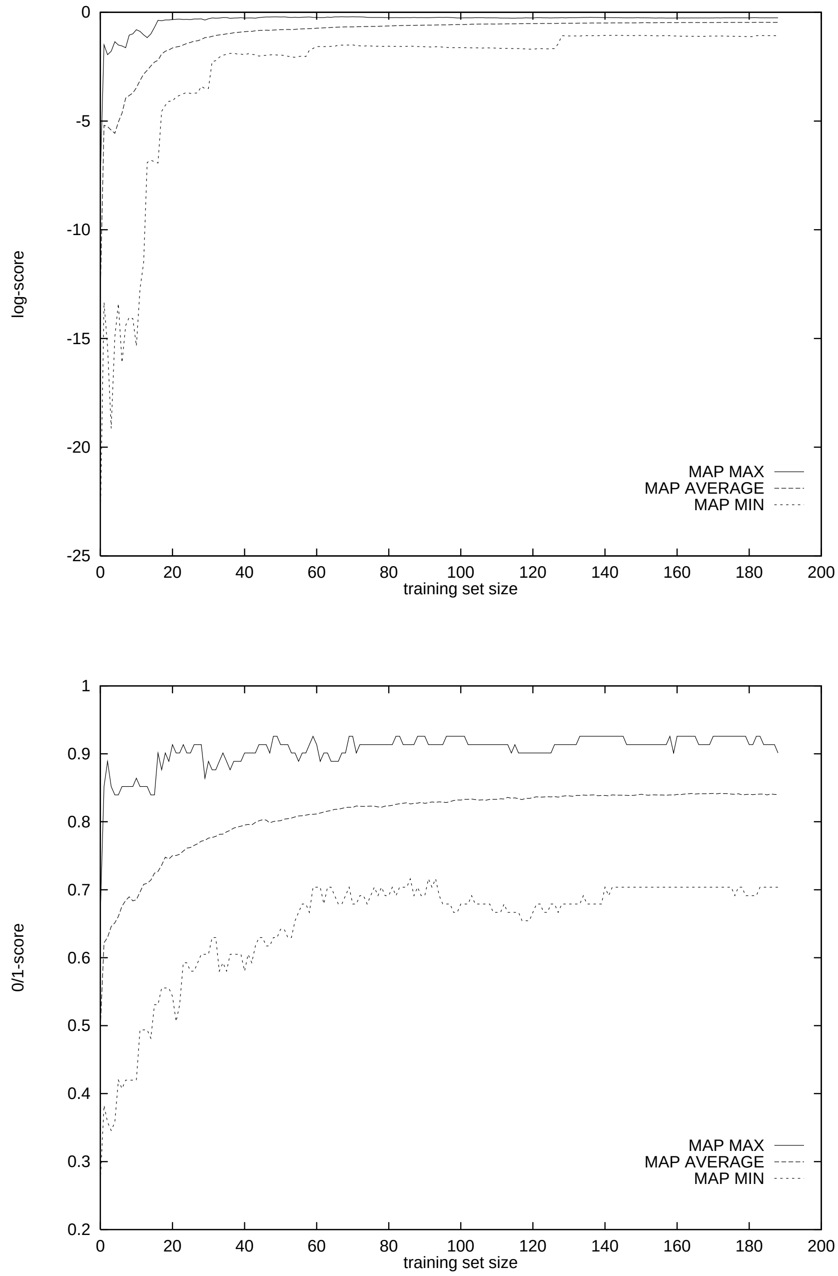
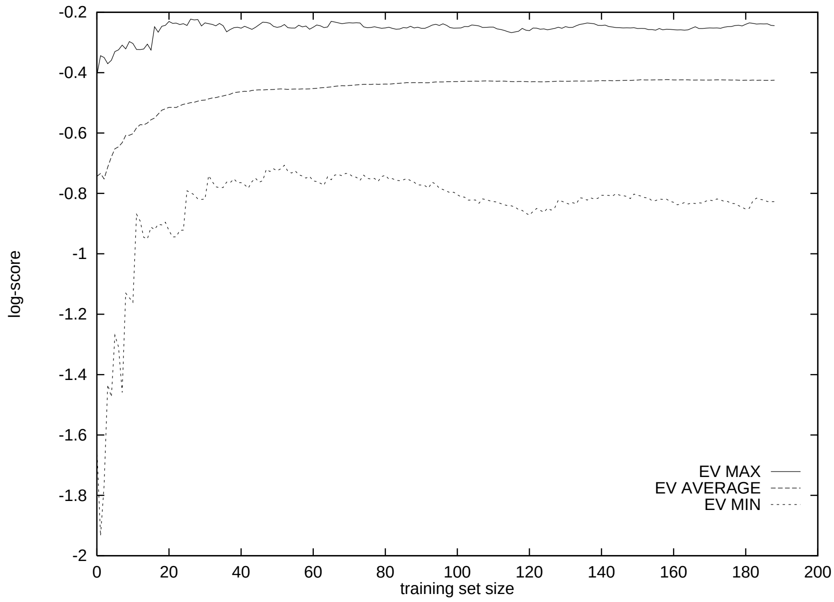


Figure B.12: The maximum and minimum performance of the MAP predictive distribution method by log-score (up) and 0/1-score (below) for the Heart Disease data set.



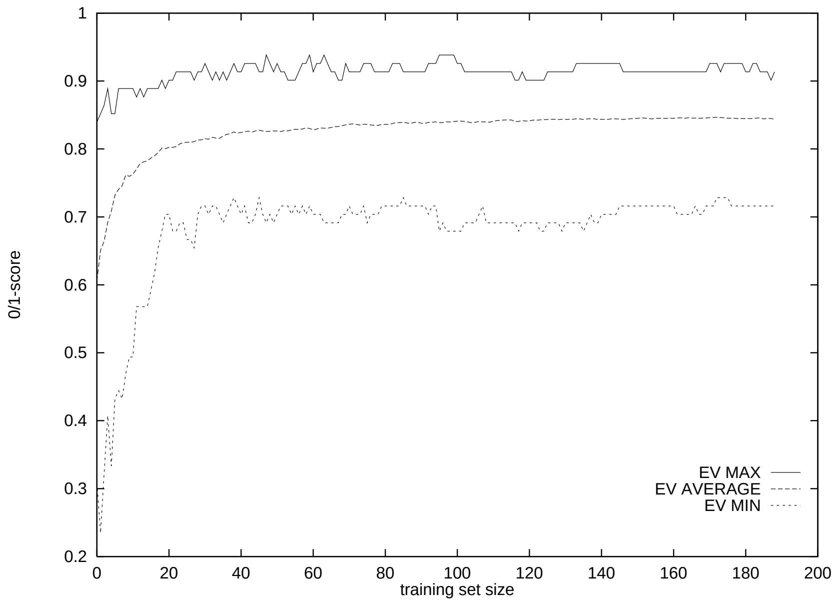
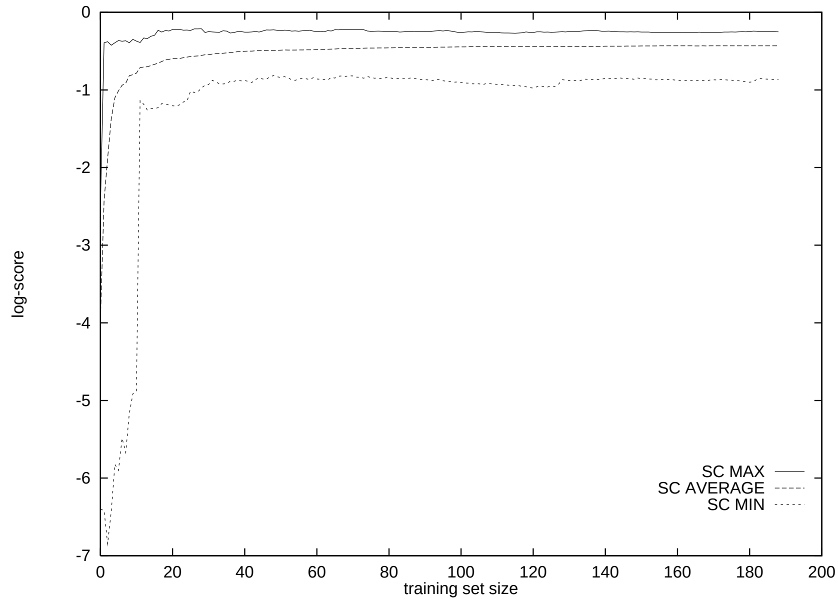


Figure B.13: The maximum and minimum performance of the evidence predictive distribution method by log-score (up) and 0/1-score (below) for the Heart Disease data set.



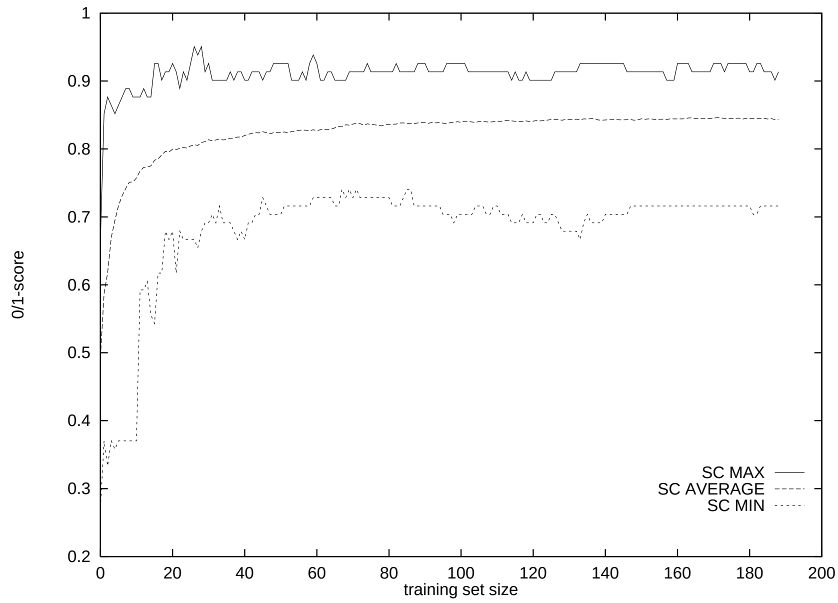
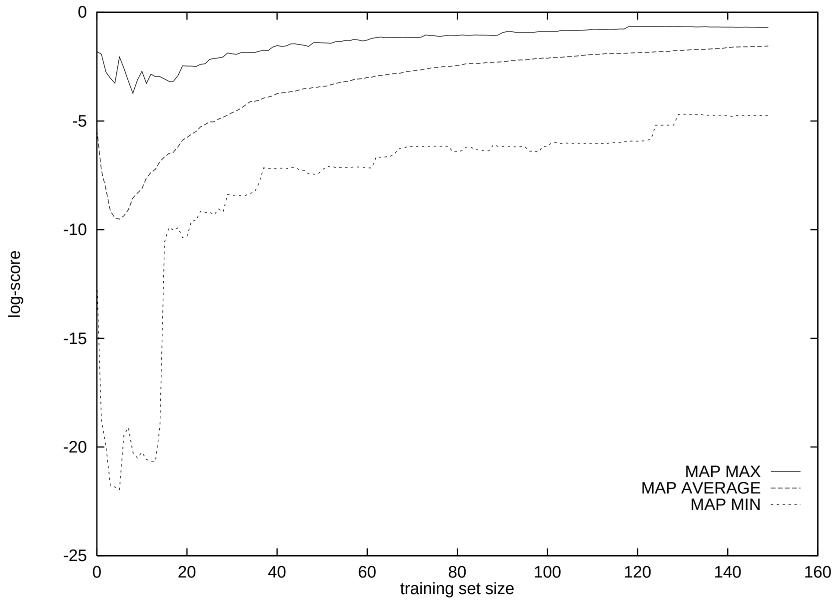


Figure B.14: The maximum and minimum performance of the stochastic complexity predictive distribution method by log-score (up) and 0/1-score (below) for the Heart Disease data set.



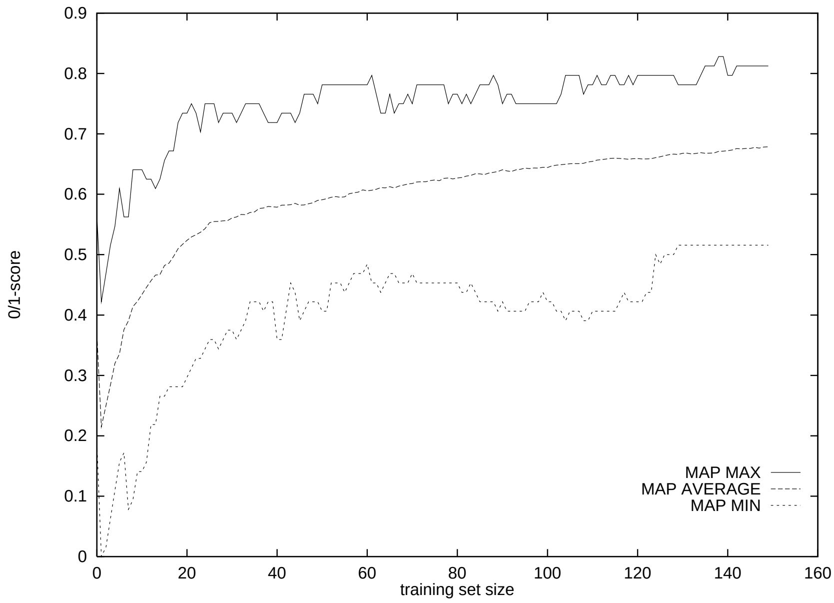


Figure B.15: The maximum and minimum performance of the MAP predictive distribution method by log-score (up) and 0/1-score (below) for the Glass data set.

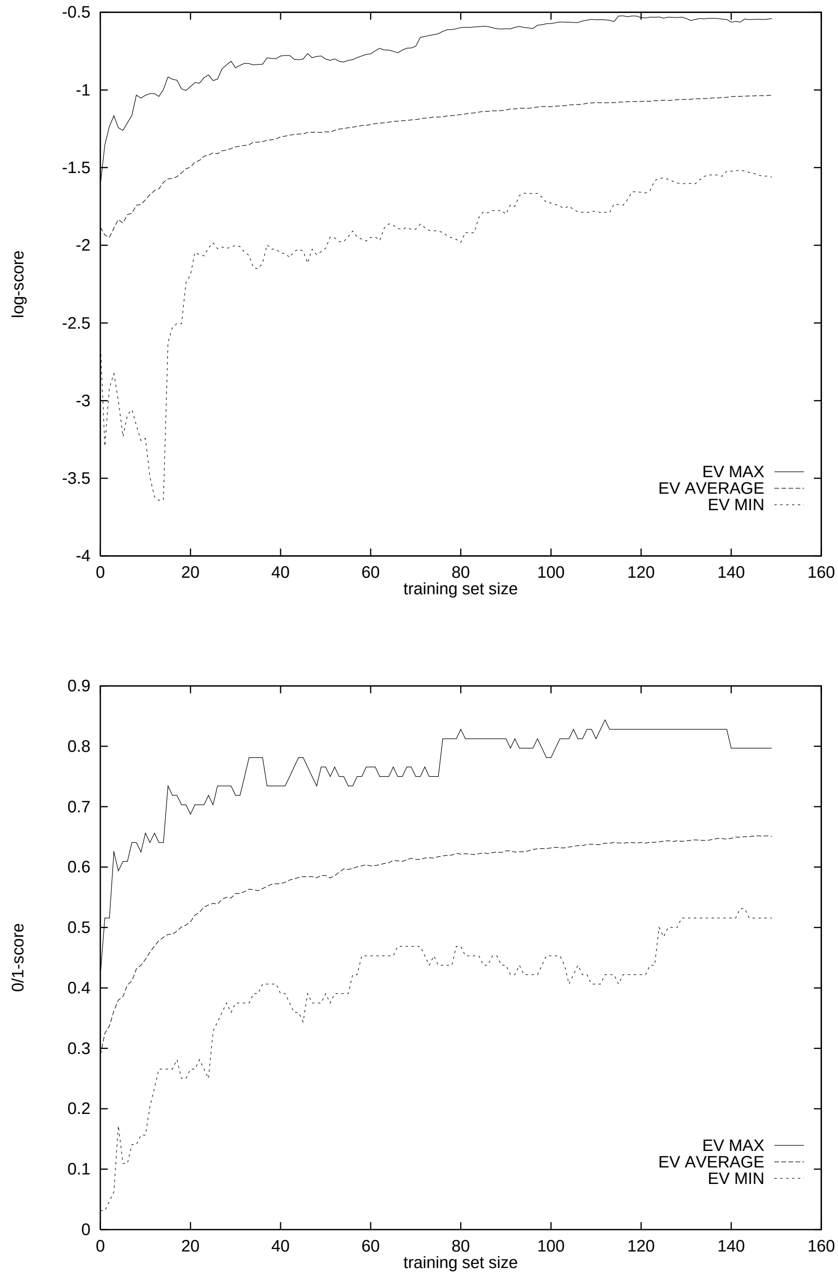
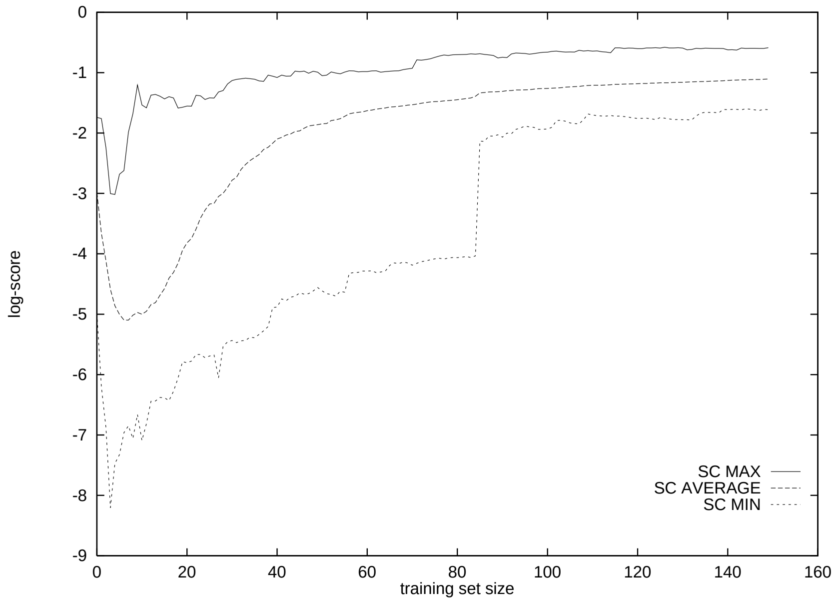


Figure B.16: The maximum and minimum performance of the evidence predictive distribution method by log-score (up) and 0/1-score (below) for the Glass data set.



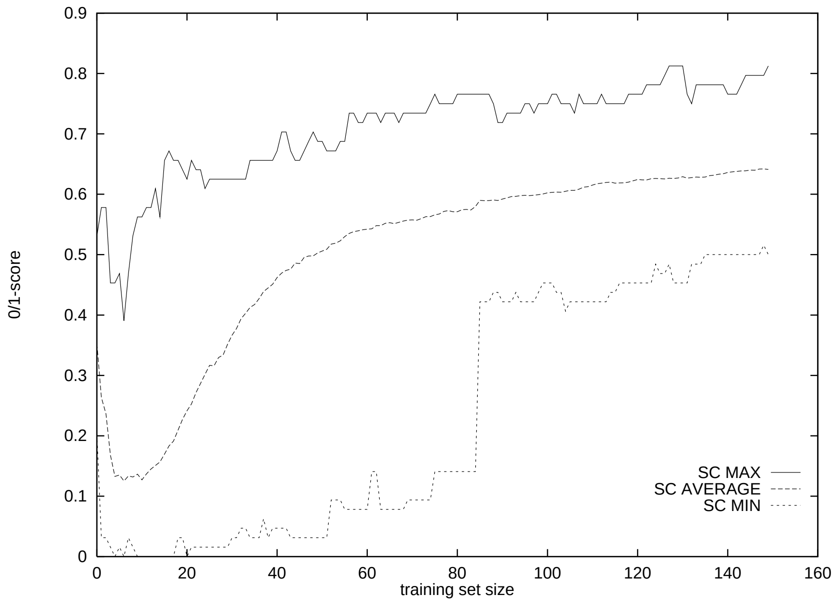
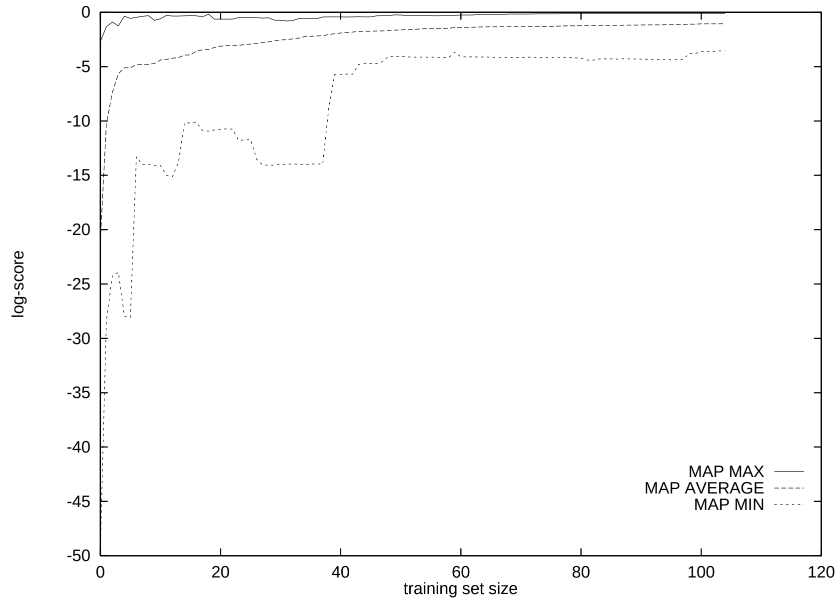


Figure B.17: The maximum and minimum performance of the stochastic complexity predictive distribution method by log-score (up) and 0/1-score (below) for the Glass data set.



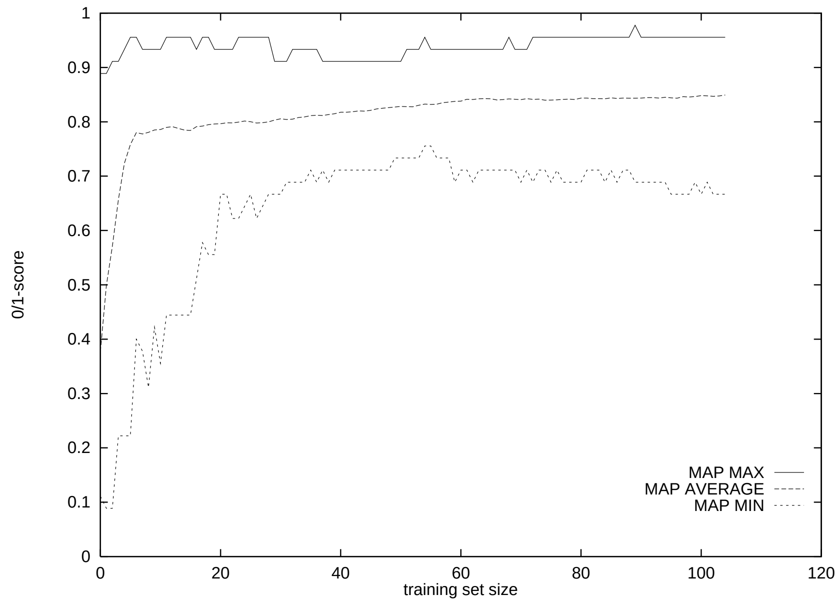
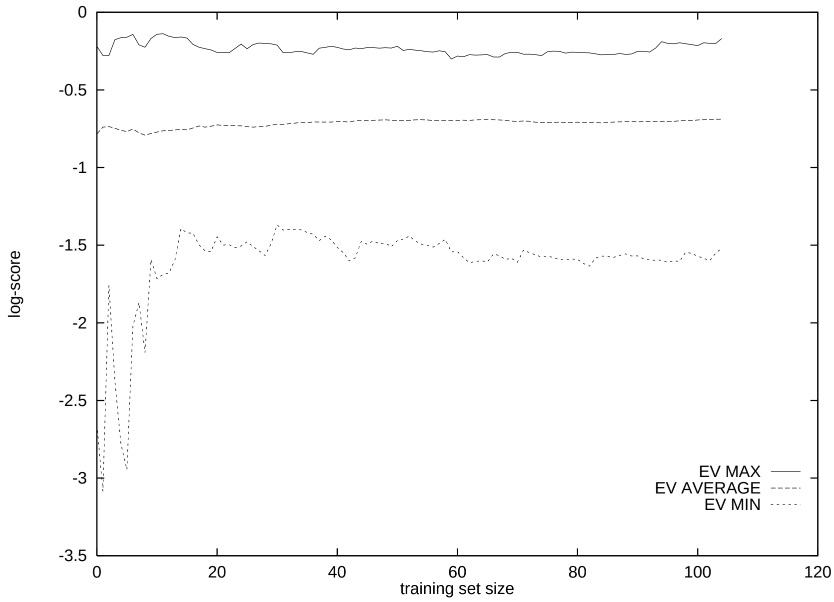


Figure B.18: The maximum and minimum performance of the MAP predictive distribution method by log-score (up) and 0/1-score (below) for the Hepatitis data set.



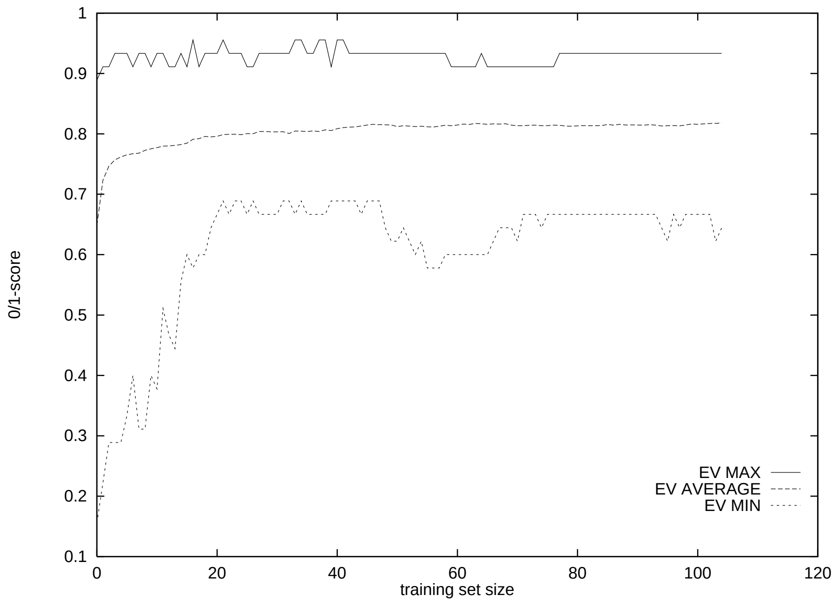


Figure B.19: The maximum and minimum performance of the evidence predictive distribution method by log-score (up) and 0/1-score (below) for the Hepatitis data set.

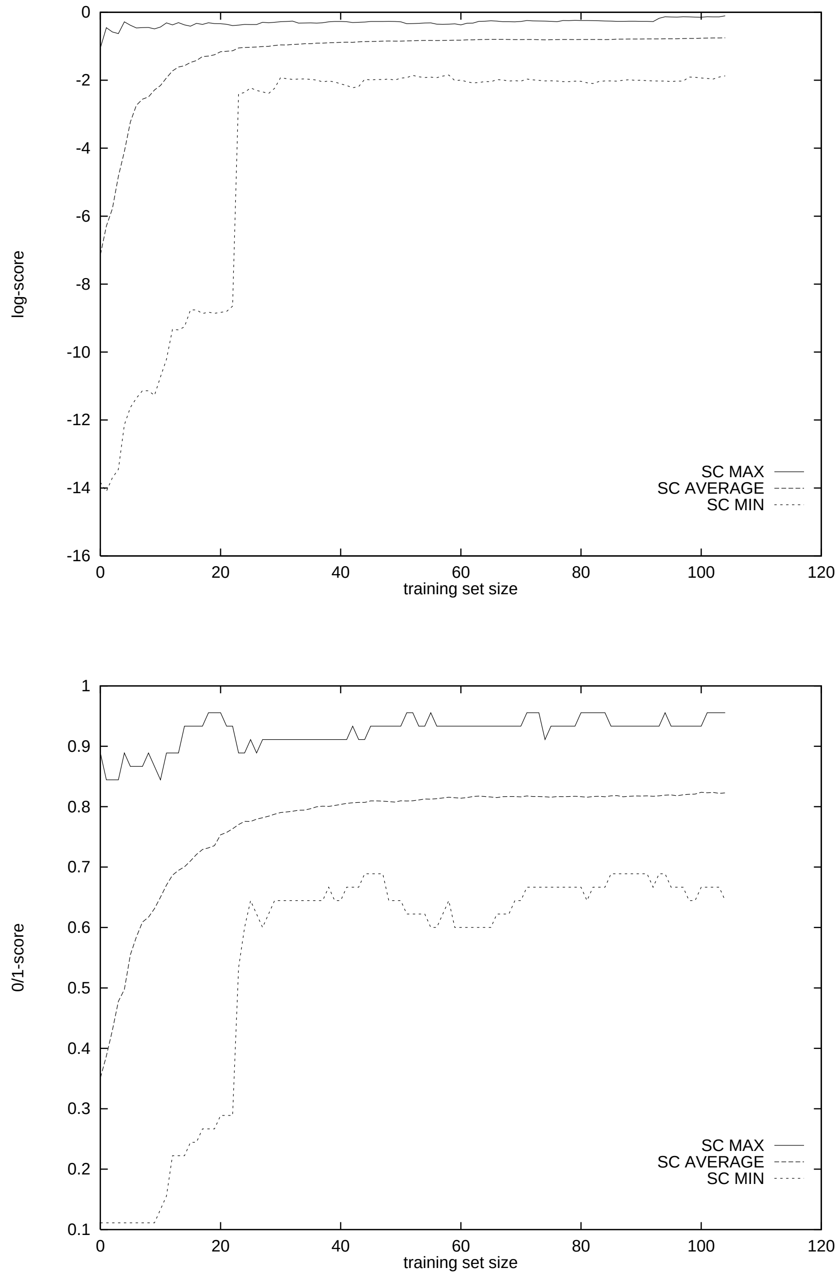


Figure B.20: The maximum and minimum performance of the stochastic complexity predictive distribution method by log-score (up) and 0/1-score (below) for the Hepatitis data set.

Appendix C

AIC and BIC approximations vs. complete evidence

This Appendix presents the results of the comparison of the evidence approximations in the complete data case, as discussed in Section 5.4. The results for each of the five datasets (Iris, Glass, Primary Tumor, Diabetes, DNA) are presented in Figures C.1–C.5. Here "EBC" denotes the exact evidence formula given by (5.20). It should be observed that the graphs are sorted in ascending order by the size of the corresponding dataset.

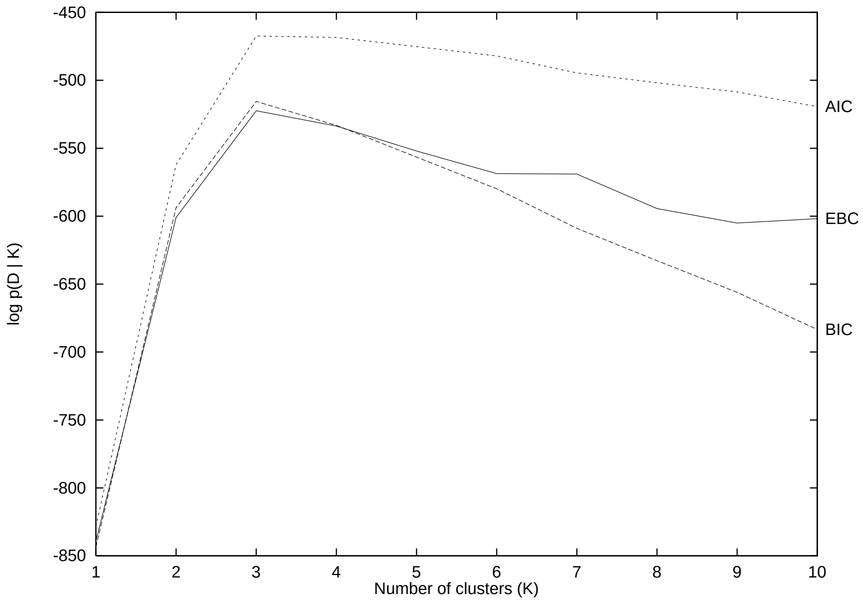


Figure C.1: Results with the Iris dataset.

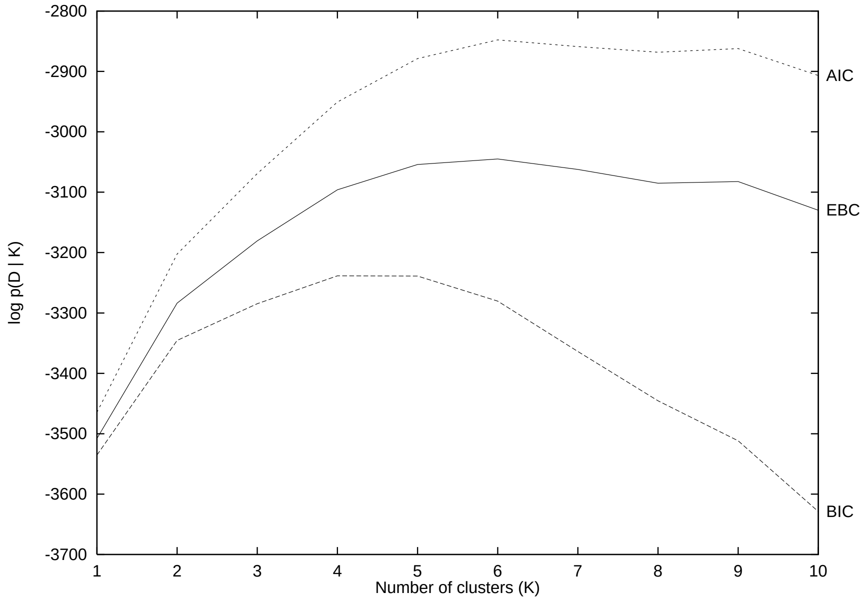


Figure C.2: Results with the Glass dataset.

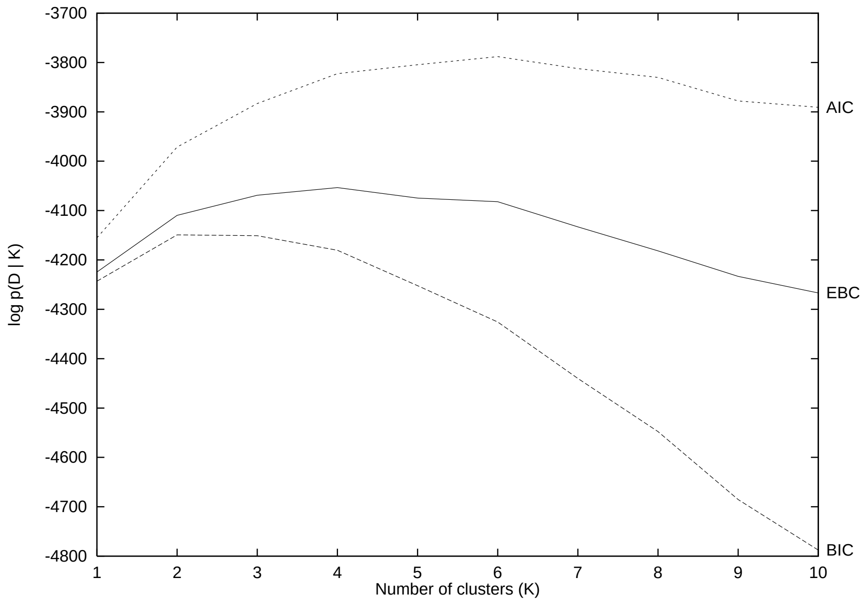


Figure C.3: Results with the Primary Tumor dataset.

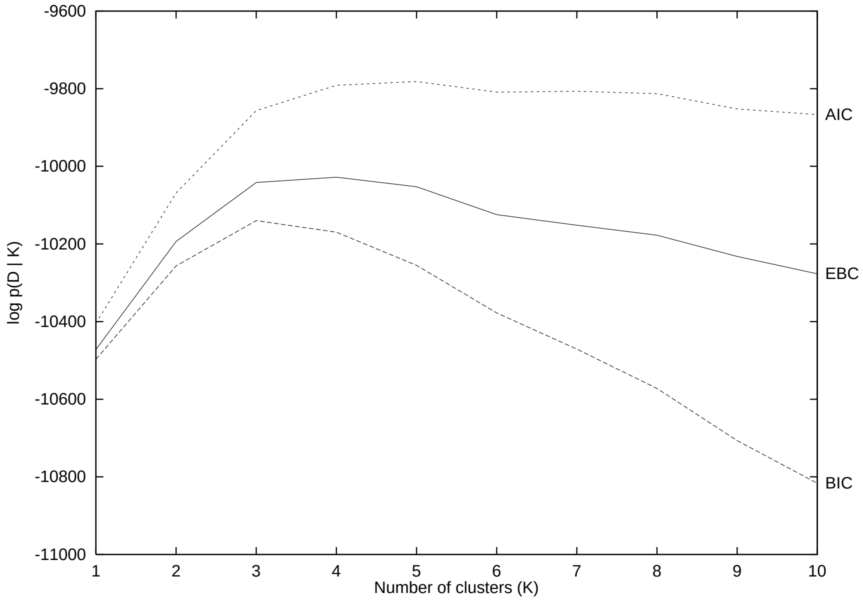


Figure C.4: Results with the Diabetes dataset.

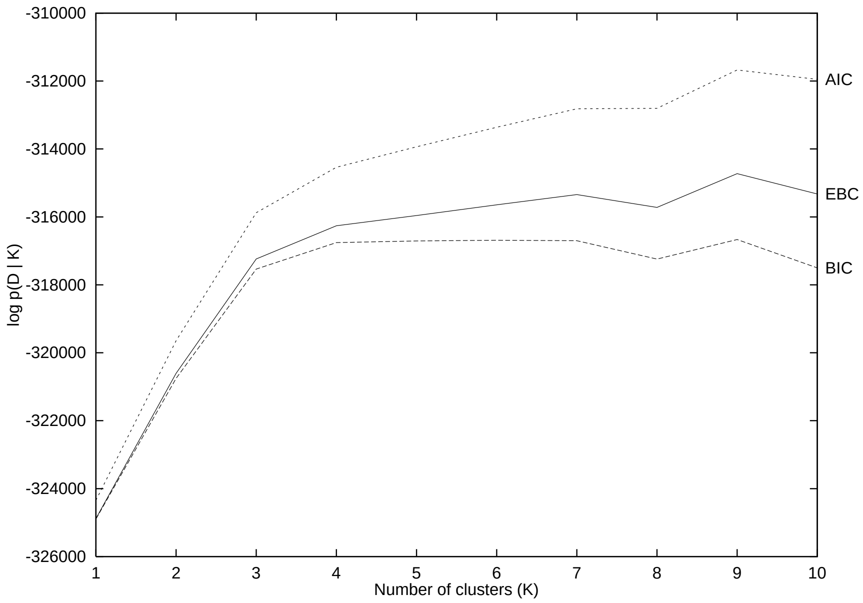


Figure C.5: Results with the DNA dataset.

Appendix D

Cheeseman-Stutz approximation vs. cross-validation

This Appendix presents both the results of comparing the Cheeseman-Stutz evidence approximation and cross-validation in the incomplete data case with real data sets, and the performance of C-S approximation with synthetic data (see the discussion in Section 5.4). The Cheeseman-Stutz measure requires the calculation of the MAP estimate $\hat{\theta}$. In these experiments for each test case 50 repetitions of EM with random initialization were performed, of which the highest posterior θ^b was chosen to represent $\hat{\theta}$. EM convergence proved to be very fast, in the order of 100 or less iterations for each individual run.

Figures D.1, D.2, and D.3 illustrate the behavior of the Cheeseman-Stutz approximation and the cross-validation results for model classes with $K = 1, \ldots, 10$. Similarly in the synthetic data experiments the Cheeseman-Stutz approximation was determined for model classes $K = 1, \ldots, 10$. The synthetic data was generated by varying the number of generating clusters (G) from 2 to 8, and the Cheeseman-Stutz behavior was compared to the so called "gold standard", i.e., the number of components used in the generation of data. The property of interest in both set of experiments is the model class selected, i.e., the number of clusters for which the measures give the maximum value. The results are shown in Figures D.4–D.7.

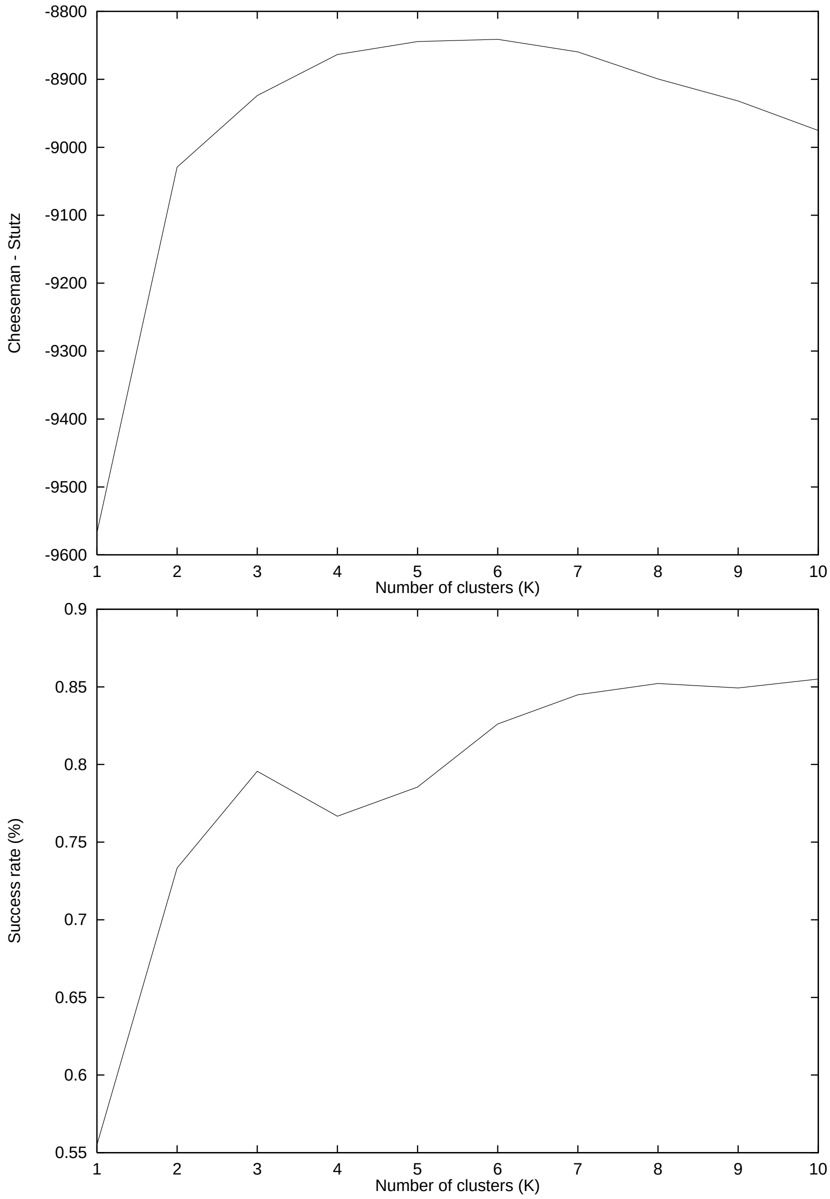
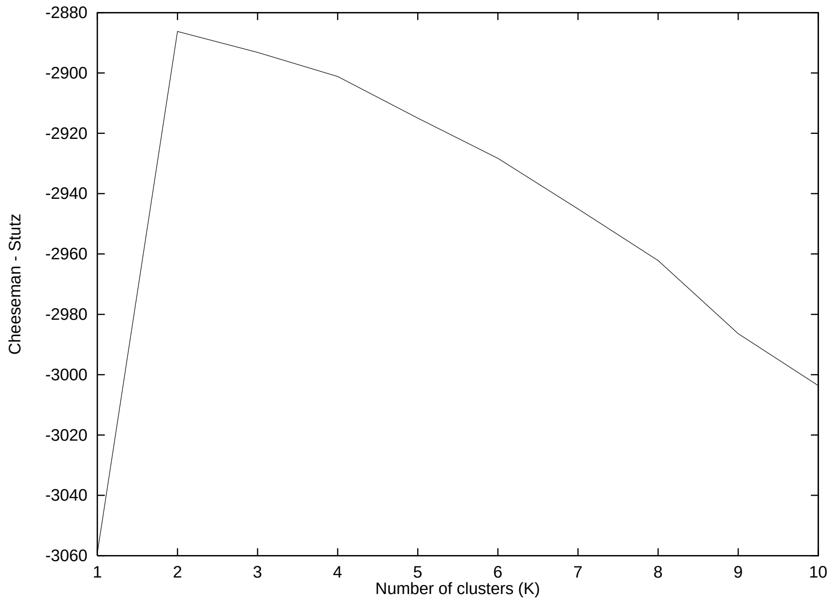


Figure D.1: The C-S measure (up) and the cross-validation results (below) with the Australian dataset.



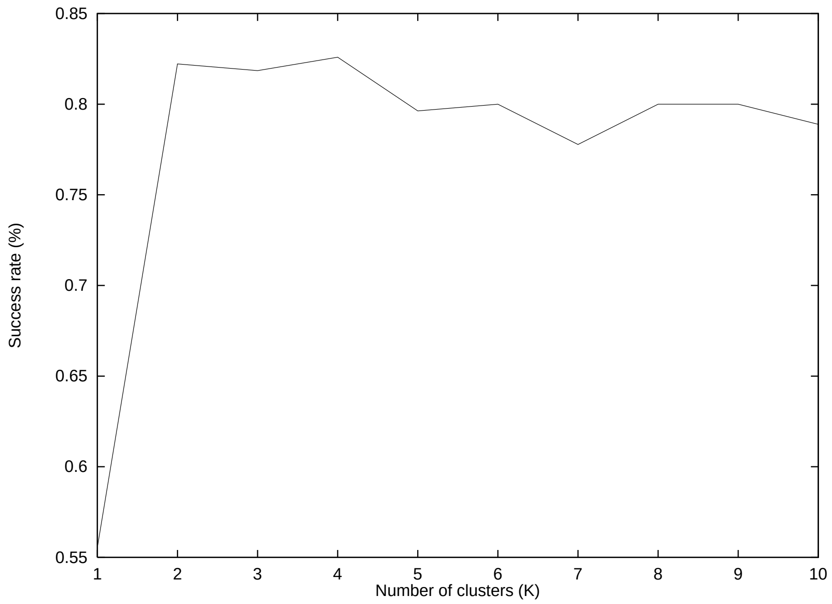
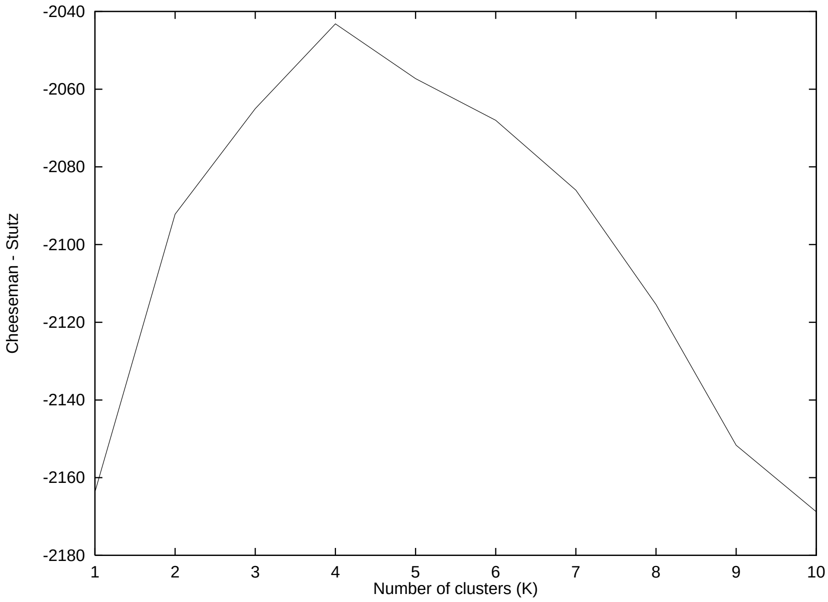


Figure D.2: The C-S measure (up) and the cross-validation results (below) with the Heart disease dataset.



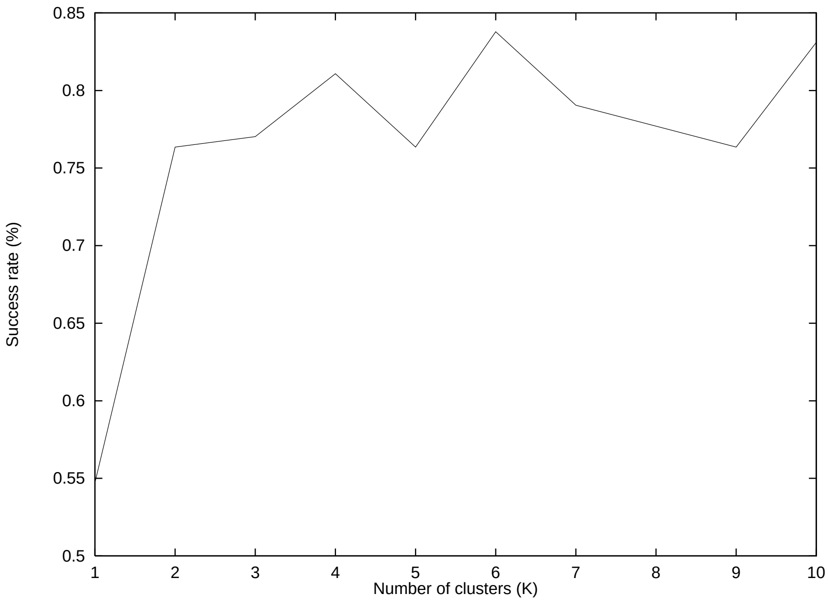


Figure D.3: The C-S measure (up) and the cross-validation results (below) with the Lymphography dataset.

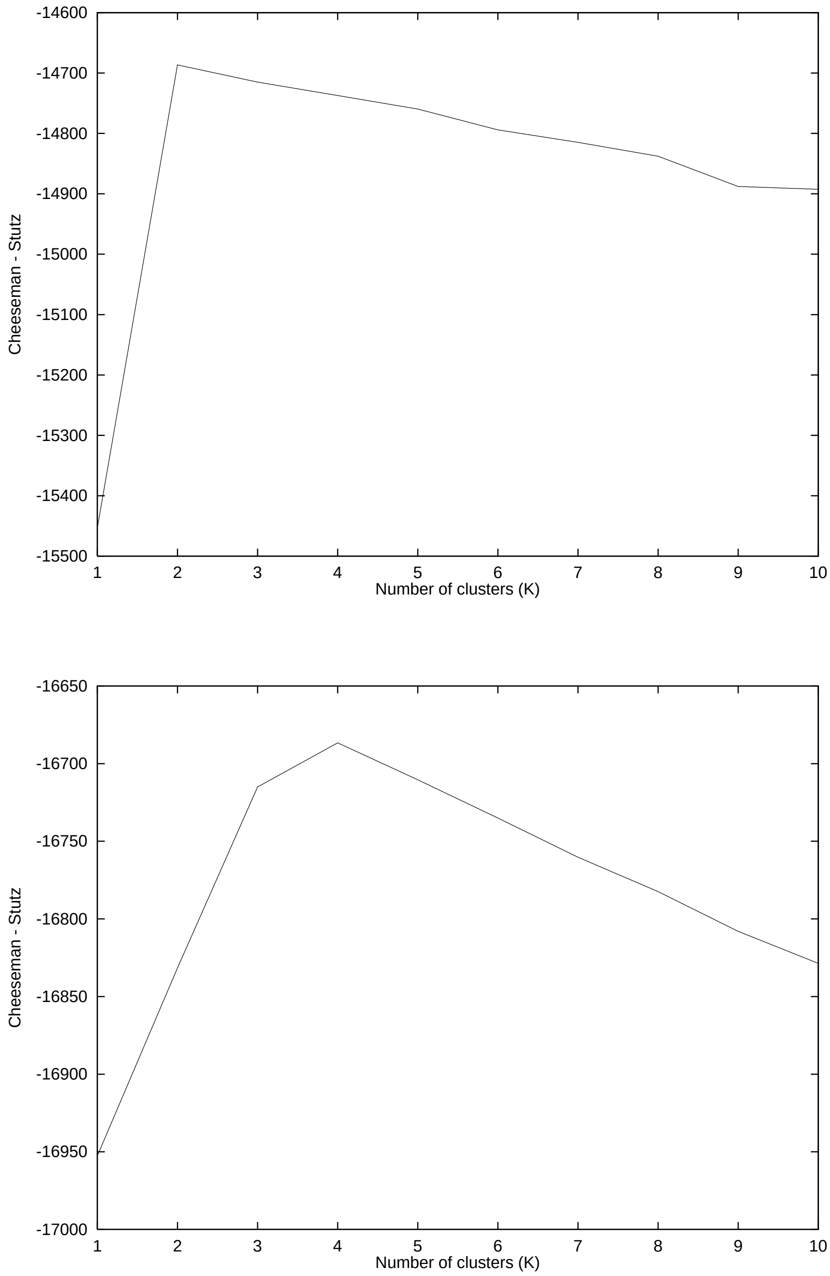


Figure D.4: The C-S measure for cluster counts 2 (up) and 4 (below) with the DG10 dataset.

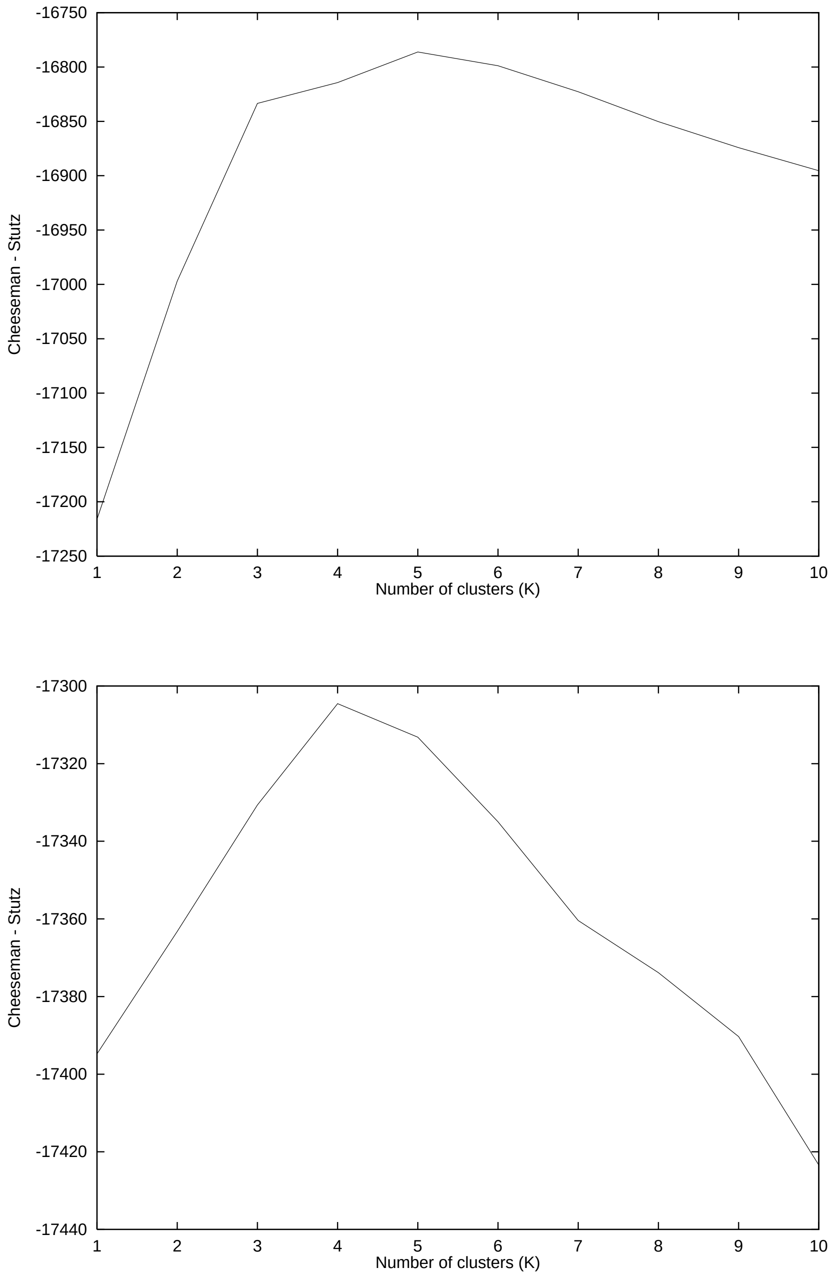


Figure D.5: The C-S measure for cluster counts 6 (up) and 8 (below) with the DG10 dataset.

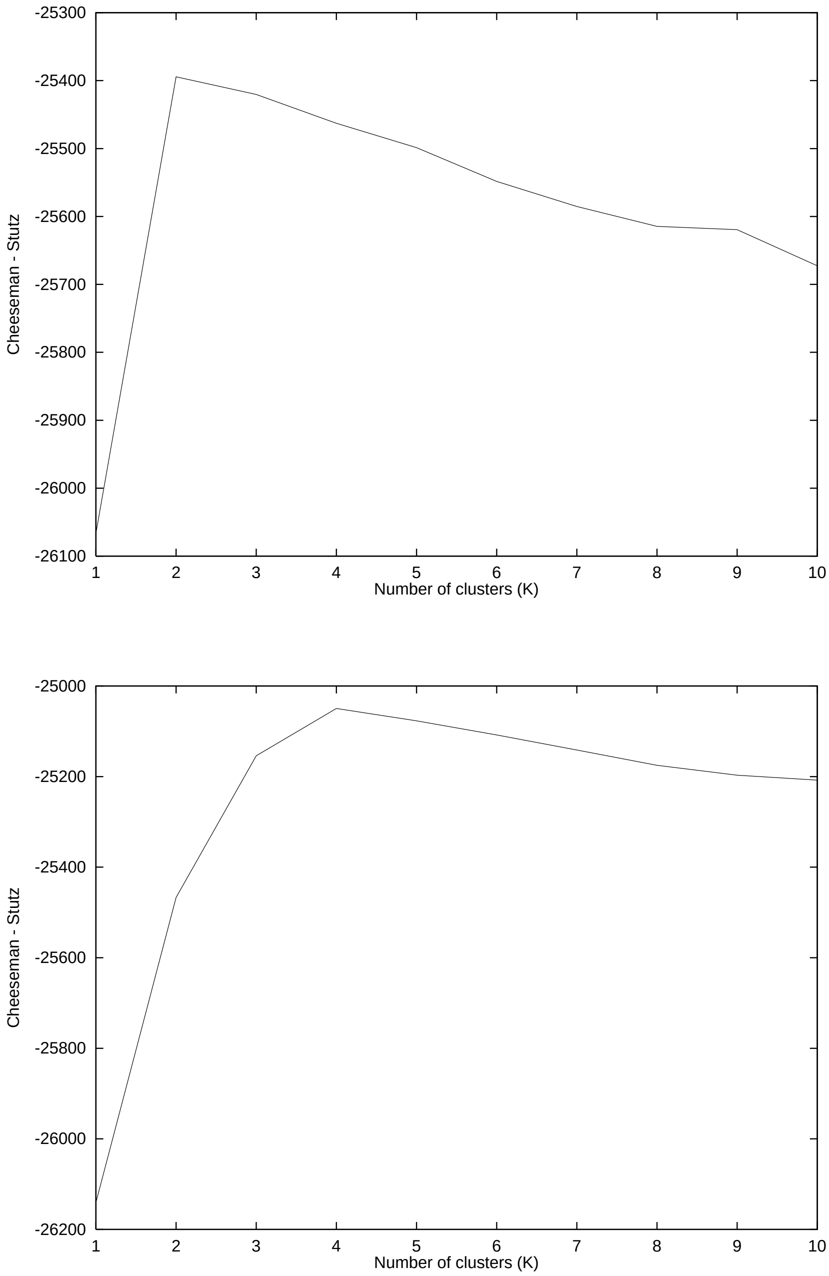
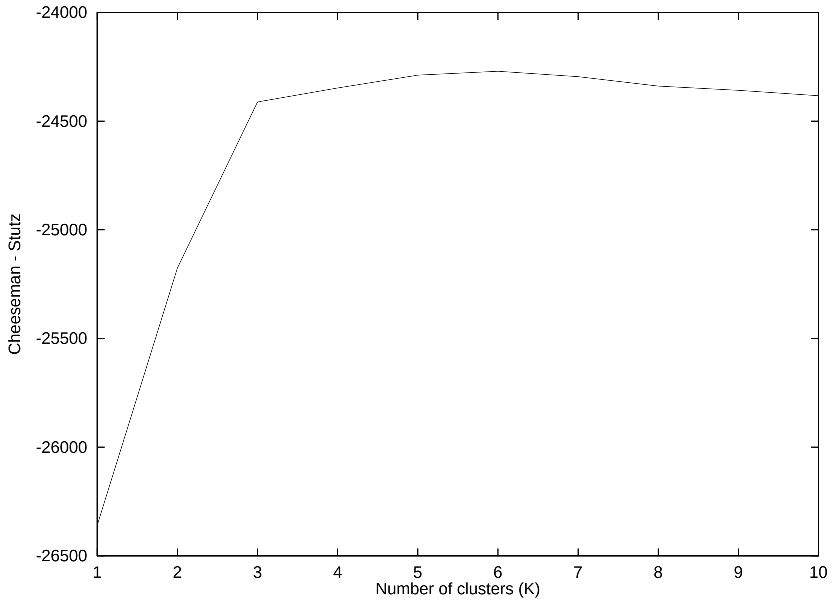


Figure D.6: The C-S measure for cluster counts 2 (up) and 4 (below) with the DG20 dataset.



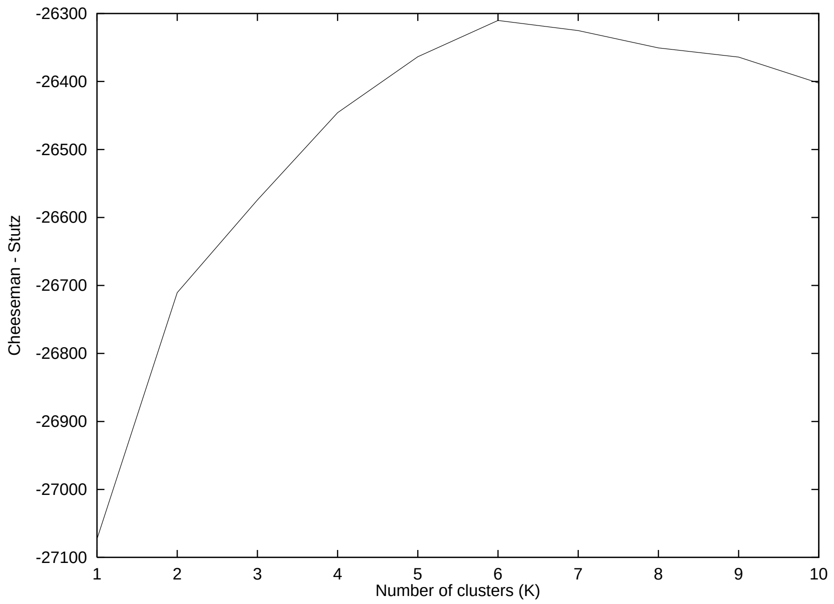


Figure D.7: The C-S measure for cluster counts 6 (up) and 8 (below) with the DG20 dataset.

ISSN 1238-8645
ISBN 951-45-7746-9
Helsinki 1997
Helsinki University Press

Henry Tirri: Plausible Prediction by Bayesian Inference **A-1997-1**