A PROBABILISTIC EXEMPLAR BASED MODEL

A thesis submitted to the University of Salford

for the degree of Doctor of Philosophy

July 1998

By

Andrés Florencio Rodríguez Martínez

Department of Computer & Mathematical Sciences

TIME Research Institute

University of Salford

Contents

List of Tables

List of Figures

Abstract

A central problem in case based reasoning (CBR) is how to store and retrieve cases. One approach to this problem is to use exemplar based models, where only the prototypical cases are stored. However, the development of an exemplar based model (EBM) requires the solution of several problems: (i) how can a EBM be represented? (ii) given a new case, how can a suitable exemplar be retrieved? (iii) what makes a good exemplar? (iv) how can an EBM be learned incrementally?

This thesis develops a new model, called a probabilistic exemplar based model, that addresses these research questions. The model utilizes Bayesian networks to develop a suitable representation and uses probability theory to develop the foundations of the developed model. A probability propagation method is used to retrieve exemplars when a new case is presented and for assessing the prototypicality of an exemplar.

The model learns incrementally by revising the exemplars retained and by updating the conditional probabilities required by the Bayesian network. The problem of ignorance, encountered when only a few cases have been observed, is tackled by introducing the concept of a virtual exemplar to represent all the unseen cases.

The model is implemented in C and evaluated on three datasets. It is also contrasted with related work in CBR and machine learning (ML).

DECLARATION

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institution of learning.

Partial results of this thesis have been presented jointly with the supervisors in the following papers:

- A probabilistic model for CBR, Proc. 2nd International Conference on Case Based Reasoning (ICCBR-97), Providence, RI, U.S.A., pp 623-642, 1997.

- A probabilistic exemplar based model, Proc. 3rd United Kingdom Case-Based Reasoning Workshop, Manchester, England, pp 167-176, 1997.

Acknowledgements

The development of this thesis was supported by a grant from CONACYT and IIE under the In-House IIE/SALFORD/CONACYT doctoral programme.

Special thanks are due to my supervisor in the University of Salford, Dr. Sunil Vadera and in México, Dr. Enrique Sucar.

I am grateful to Professor F.A. Holland and Dr. E. Wilde for their advice and continuous help during the development of this thesis.

I wish to express my thanks to the Instituto de Investigaciones Eléctricas (IIE), for giving me the opportunity and for supporting my participation in the programme, specially to Dr. Pablo Mulás del Pozo, Ing. Fernando Kohrs Aldape, and Dr. David Nieva.

I am grateful to Dr. David Aha, from Naval Research Laboratory U.S.A., for his valuable advice in the research proposal.

I also wish to thank the members of the informatics module of the programme for their continuous advice and encouragement, specially to Dr. Luis A. Pineda and Dr. Guillermo Rodríguez. Thanks are also due to Dr. Eduardo Morales and my fellow investigators Pablo Ibargüengoytia, Pablo de Buen and Sergio Santana.

Thanks are also due to the late Fis. Andrés Estebaranz for their untiring effects to make the In-House IIE/Salford CONACYT programme increasingly successful.

Chapter 1

INTRODUCTION

1.1 Background

Case based reasoning (CBR) is a problem solving paradigm that has attracted a lot of interest from both academics and practitioners. CBR has been defined [Riesbeck & Schank 1989] as a paradigm that solves new problems by adapting solutions that were used to solve similar problems in the past. It has been applied to a wide range of domains including planning, medical diagnosis, legal reasoning, design, and education [Marir & Watson 1994, Watson 1997, Allen 1994, Leake 1996]. Some notable applications are as follows.

- CLAVIER: This is a CBR system that provides interactive support to operators in the process of configuring the layout of composite parts for curing in a large convection heater, called an autoclave [Mark 1989, Barletta & Hennessy 1989, Mark et al. 1996].

- FormTool: This is a CBR system that is used for colour matching in a plastic production process. FormTool determines the colorants and loading to use for producing a specific colour of plastic and aims to minimise cost [Cheetham & Graf 1997].

- SMART: This is a CBR system that helps to diagnose and repair hardware and software problems. SMART is a help desk assistant. The user describes his or her problem and the CBR system retrieves cases that can help in the solution of the problem [Acorn & Walden 1992].

- Large customer service: This is a CBR system that is part of an integral system of customer service. The CBR system, which does not have a name, provides consistent and high quality customer service support to non-technical customers [Thomas et al. 1997].

- MEDIC: This is a medical diagnosis CBR system that helps in the planning and execution of a sequence of actions for diagnosing lung diseases [Turner 1989].

As these applications suggest, CBR has already resulted in substantial applications since its initial development by Schank and his group in the early 80's [Schank 1982]. However, several researchers have pointed out that there are significant issues that still have to be resolved before these systems achieve their full potential [Kolodner 1993, Riesbeck 1996]. The issues raised by these researchers can be classified into the following three categories.

1. The first category comprises the fundamental issues of indexing, case representation and manipulating cases [Kolodner 1993, Kolodner 1996, Riesbeck 1996]. For example:

   - determining the optimal level of abstraction for indices,
   - developing well defined indexing methodologies to reduce the costs of developing and applying indexing vocabularies,
   - developing (semi)automated index selection,

- determining general purpose indexing vocabularies that can be utilised in different domains,
- determining the size and the optimal level of abstraction of the cases,
- elaborating methods in order to adapt cases for new domains or tasks.

2. The second category comprises the knowledge engineering issues of building CBR systems more easily. This involves the development of tools that enable more people to implement CBR applications quickly and reliably [Kolodner 1993, Watson 1997, Riesbeck 1988].

3. The third category comprises the technological issues. In this category, the main issue is that of scaling up [Kolodner 1993]. How can a current retrieval algorithm that works with a few cases (perhaps hundreds) be extended so that it works efficiently for thousands of cases? Currently, some researchers, notably Veloso (1996), Kitano & Shimazu (1996), and Jabbour et al. (1988), have been addressing this problem and they have shown that the current technology can, in some cases, be extended to support large case bases.

The work presented in this thesis addresses the problems in the first of these categories. This category presents several challenges to the CBR research community, including the following.

- What is a case?
- How are the cases represented?
- How are the cases organised and indexed in the memory?
- What is the process of retrieving similar cases from the memory?
- What is the process of adapting the solution?

- How should the new solution be evaluated?
- How does a CBR system determine if the new case should be retained in memory?

Answers to the last four questions describe the main processes of all CBR systems [Aamodt & Plaza 1994]. Figure 1.1 shows these four main processes: (i) retrieving similar cases from memory that help in the solution of a new case, (ii) reusing or adapting selected similar cases to solve the new case, (iii) evaluation of the new solution and (iv) determining if the new case should be retained in the case memory. As can be seen in Fig. 1.1, the central part of these processes is the case memory. Thus, the organisation and management of the memory have an important role in the development of CBR systems.

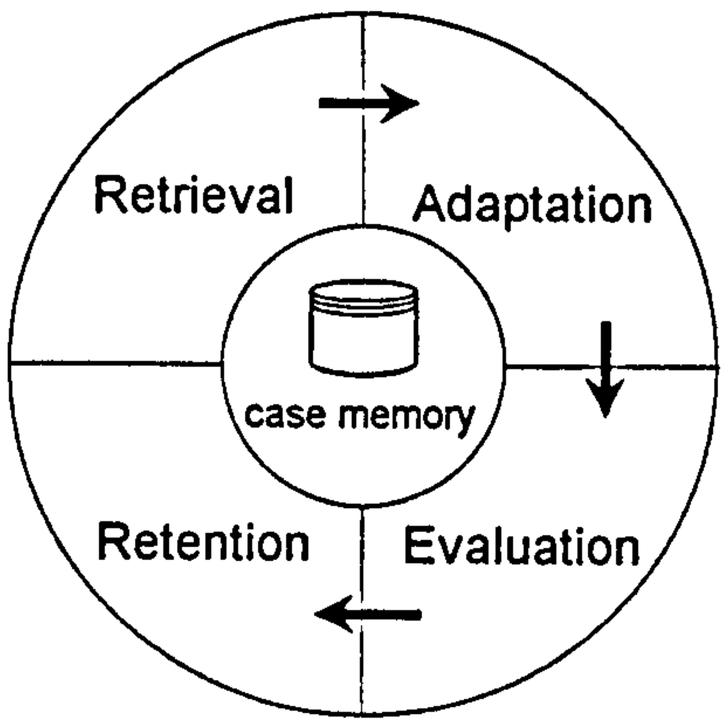


Figure 1.1: Processes in a case based reasoning model.

Organising and indexing cases in memory requires the solution of two problems. The first problem is the selection of those features that can be used to index and retrieve similar cases. The second problem is the organisation of the case memory so that the retrieval process is efficient and accurate.

A simple approach is to store a flat database of cases and scan all the cases to identify the most similar cases. Although simple, this approach has been

successfully used in a number of small applications of CBR [Barletta & Hennessy 1989, Mark 1989]. For applications where many more cases are involved, this simple organisation can be expected to be slow [Kolodner 1993].

A more sophisticated method is to partition the cases into clusters and organise them hierarchically. The hierarchy can then be searched more efficiently by following a path depending on the features of the new case. Different types of hierarchies have been proposed leading to different approaches. One approach is to use decision trees so that the leaf nodes contain the cases and where the internal nodes contain questions that can be used to partition the cases. So for example, systems like ReMind [Althoff et al. 1995] provide a tree induction algorithm that can be used to avoid examining all the cases. This kind of approach is particularly useful when large databases of cases are already available. However, when cases are not available in advance, and the domain is not well defined this approach is more difficult to apply.

Another approach is to use an abstraction hierarchy where each internal node is an abstraction of the cases represented by its children. These hierarchies are known as discrimination networks or redundant discrimination networks when the nodes represent overlapping regions of cases. The systems MEDIATOR [Kolodner & Simpson 1989], JULIA [Hinrichs 1989], and CASEY [Koton 1988] have used this approach and their outcomes have shown its utility. However, these systems require much more memory to store the network and the procedures for adding new cases are very expensive since the abstraction process needs to examine many nodes and the abstraction hierarchy may need to be restructured [Kolodner 1993].

Thus current approaches to CBR work well in some situations, but also have problems in other situations. In particular, for domains, sometimes called *weak domains* [Porter et al. 1990], where: (i) the categories or concepts are difficult to define by necessary and sufficient features, (ii) the categories can be non-disjoint,

(iii) the data are not structured, (iv) all the data do not exist in advance, and (v) there is uncertainty in how the categories are represented by cases, these approaches have the following problems:

- Most of these approaches require all the features and examples in advance. So for instance, the tree induction algorithms that have been used are descendants of ID3 [Quinlan 1996] that requires a fixed width table.

- Most of the commercial CBR tools are not incremental. For example, the tree induction algorithm used by ReMind requires all the cases in advance. Although academic systems such as MEDIATOR, JULIA, and CASEY are incremental, they require expensive and complex procedures to store new cases which can become impractical as the number of cases increases.

- Most of the approaches do not handle uncertainty explicitly. Most systems use a weighted sum of the differences between the new case and an existing case as a measure of similarity. This measure can result in overfitting in the presence of noisy data and can be sensitive to the weights selected [Tirri et al. 1996*a*]. In addition, this measure is difficult to justify theoretically.

An alternative approach, that is perhaps more applicable to weak domains, is to store only prototypical cases. This approach, known as the *exemplar based* model has its basis in cognitive theories, which postulate that concepts can be represented by exemplars [Rosch & Mervis 1975, Smith & Medin 1981, Medin & Schaffer 1978]. Exemplar based models do not necessarily require all the features or all the cases in advance. Hence, this thesis focuses on developing an exemplar based model. The next section describes the main problems of developing exemplar based models and presents the objective of this thesis.

1.2 The Problem and Thesis Objective

As mentioned above, an exemplar based model only stores prototypical cases. To understand the idea of exemplar based models, consider Fig. 1.2 which shows two categories, A, B (the solid lines) in a weak domain. The figure also shows some exemplars, e_i that represent regions (the dashed lines) that contain cases (the dots).

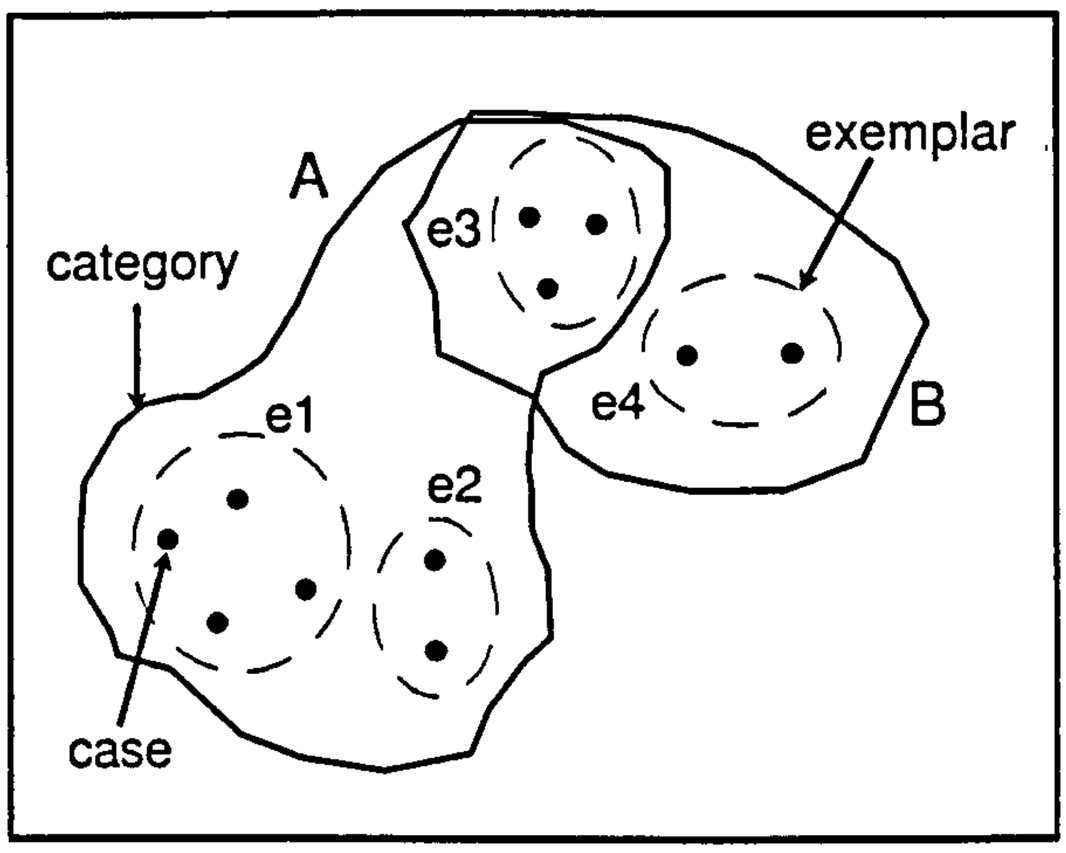


Figure 1.2: Cases, exemplars and categories in a weak domain.

In this example, the category A is represented by the exemplars e_1, e_2, and e_3 and the category B is represented by the exemplars e_3 and e_4. Also suppose that the exemplars e_1, e_2, e_3, and e_4 currently represent 4, 2, 3, and 2 cases respectively. Now suppose that a new case is given. The following two functions must be performed by an exemplar based model:

1. Determine the exemplar that best classifies a new case given the available information.

2. Determine how knowing the new case and its classification can be used to improve the accuracy of the model.

The first of these functions is a classification task, while the second can be viewed as a supervised learning task [Aha 1991].

The best known exemplar based model is encoded in the Protos system [Bareiss 1989]. It is therefore worth summarising the main characteristics of Protos (Chapter 5 describes Protos in more detail). Protos is a case based classification and case based knowledge acquisition model. The model uses a cognitive approach based on exemplars to represent concepts [Smith & Medin 1981]. Protos organises the case memory in a semantic network where the nodes represent categories, exemplars and features. The arcs represent the relationships between categories, exemplars and features. Protos uses different kinds of relations to relate its components. In the training phase, Protos learns these relations from user provided explanations. Based on the explanation, Protos uses heuristics to assign default weights to each relation. So for example, the functional relation *"enables"* has a weight of 0.9 and the definitional relation *"is equivalent to"* has a weight of 1.0. Furthermore, each relation can have an associated set of qualifiers, where each qualifier has a strength in the relation. For example, the qualifier *"moderately"* has a strength of 0.7 and the qualifier *"sometimes"* has a strength of 0.6. The values assigned to the qualifiers are also heuristically determined. So, when a relation is used in an explanation, Protos heuristically computes its actual weight as a function of its default weight and its associated qualifiers.

Remindings, censors, prototypicality, and difference links are the indices that Protos uses to classify new exemplars. Remindings are used to associate features with categories or particular exemplars. Censors are used as negative remindings. Prototypicality is used to provide a partial ordering on exemplars within a category and difference links are used to record important featural differences between exemplars. Protos also uses heuristics to attach weights to the remindings and censors in each category. These weights are used when new cases are

classified.

Protos learns when it either fails to classify a new case, or it misclassifies a case. The main learning operation is to retain a new case as an exemplar if it is not classified correctly and to update the remindings.

The remindings are heuristically learned from the feature-to-category explanations provided by a user. For example, given the relation "a cat has four legs", Protos regards "has" as a strong relationship and records a reminding from "four legs" to the category cats. Alternatively, Protos assumes that weaker relations such as *"is sometimes consistent with"* should not lead to remindings. For example, "the cat is sometimes ill" does not result in a reminding from ill to the category cats.

As this summary of Protos indicates, there are many heuristics that were used in the implementation of Protos. Some of these heuristics are hard to justify and lack foundation. For example, it uses a number of weights and a scheme for calculating the similarity, both of which are subjective [Porter et al. 1990]. Further, as the above example suggests, a classification task is required that involves uncertainty. Uncertainty management is a field within Artificial Intelligence (AI) and Statistics which has a long history. Amongst, the many methods of handling uncertainty, Bayesian networks have become respected and widely used [Pearl 1988, Neapolitan 1990, Dean et al. 1995, Jensen 1996].

Objective

Given the above background and motivation, the objective of this thesis is to develop an exemplar based model with foundations in Bayesian networks.

In particular, the developed model will address the following main questions:

1. Given an exemplar based representation of a weak domain, where the information is not well defined and there is uncertainty, determine the exemplar that best classifies a new case.

2. Determine how an exemplar based representation can improve its accuracy knowing a new case and its classification.

The thesis also aims to place the developed exemplar based model in the context of related CBR research and to evaluate the model empirically.

1.3 Organisation of the Thesis

To accomplish the above objective, this thesis is organised as follows.

Chapter 2 provides the background knowledge and describes the necessary concepts that are used in the thesis. First, approaches to concept representations are used to provide some background to the thesis, and then Bayesian networks are introduced and formally defined.

Chapter 3 develops the probabilistic exemplar based model. It first describes the representation used to organise the memory. Then it develops the classification and learning procedures by utilising Bayesian models. The chapter concludes with an example that illustrates the complete model.

Chapter 4 presents an empirical evaluation of the exemplar based model and the results of the evaluation. It starts by describing the test environment developed for the experiments and the experimental method. Then, it presents an evaluation of the different aspects of the model on several data sets.

Chapter 5 contrasts the work presented in this thesis in the context of other related work. In particular it describes CBR and exemplar based models, inductive learning models, and Bayesian probabilistic approaches.

Chapter 6 presents the conclusions of this thesis and describes the fields of research that have arisen during the development of this theory. Also, possible enhancements to the algorithm are briefly outlined.

This thesis is complemented with two appendixes. Appendix A presents a detailed illustration of the model and Appendix B presents a summary of the experimental results.

Chapter 2

BACKGROUND KNOWLEDGE

This chapter explains the basic concepts which are utilised in this thesis. Section 2.1 provides a description of approaches for concept representation which includes (i) their structure, (ii) their use for classifying new members, and (iii) the main challenges in utilising them. Section 2.2 describes the main concepts that must be understood in probabilistic reasoning. Subsection 2.2.1 describes the definitions and basic concepts up to the definition of the Bayes rule, which is the heart of Bayesian reasoning. Subsection 2.2.2 formally defines Bayesian networks and Subsection 2.2.3 describes the inference or probability propagation mechanism used for a subclass of networks that are singly connected. Subsection 2.2.4 describes an algorithm that is used for propagation in arbitrary networks and that utilises the propagation algorithm for singly connected networks. Finally, Subsection 2.2.5 deseribes part of a more specialised Bayesian model commonly utilised in diagnosis problems, and which is used in this thesis.

Much of the material presented in this chapter is based on the texts by Pearl (1988), Neapolitan (1990), Dean et al. (1995), and the article by Pearl et al. (1990). Readers, who are familiar with these concepts may omit the details of

this chapter.

2.1 Concept Representation

Concepts are the manner by which human beings classify things. Concept formation is a process that human beings use to define concepts. This process has been studied with different approaches in cognitive science [Bolton 1977, Smith & Medin 1981, Wittgenstein 1953, Cassirer 1953] and machine learning [Carbonell 1990] and continues to be an active research area in the development of intelligent systems. An important issue in concept formation is how the concepts are represented. Smith & Medin (1981) defined the following three types of concept representation schemes.

1. The *classical.*

2. The *probabilistic.*

3. The *exemplar based.*

The classical representation assumes that all members of a concept must share a set of features which are necessary and sufficient to belong to the concept. This assumption has a dominant position that is not adequate for weak domains, where the knowledge in the domain is not previously defined in an exact manner required by the classical approach [Porter et al. 1990]. For example, in representing the concept of a bird, the feature *flies* cannot be necessary since birds such as chickens and penguins cannot fly. However, that feature must be relevant in the concept representation since the majority of birds fly. Since the classical approach is not appropriate for weak domains, it is not described in any further detail in this section.

Both the probabilistic and exemplar based representations are used in this thesis and the following two subsections describe them together with their strengths and weaknesses.

2.1.1 Probabilistic representation

A probabilistic representation is a summary description of all members that describe the concept. This representation is defined by the set of features that have a high probability of occurring in members of the concept. When a feature is chosen as a part of the concept representation, a weight is associated with that feature. Normally, the weight given to the features is the conditional probability $P(feature \mid concept)$ that the feature is contained in the members of the concept. For example, suppose that it is required to represent the concept of furniture given the following 3 items of furniture.

Chair	Bookshelf	Desk
f_1 (physical object)	f_1 (physical object)	f_1 (physical object)
f_2 (rigid)	f_2 (rigid)	f_2 (rigid)
f_3 (has backrest)	f_7 (has crosspiece)	f_5 (has legs)
f_4 (has seat)	f_8 (has large boxes)	f_8 (has large boxes)
f_5 (has legs)	f_9 (large size)	f_{10} (for office)
f_6 (small size)		f_{11} (medium size)

Assuming that the features are independent, a possible probabilistic representation of the furniture concept could be the following.

Furniture:

Feature	Weight	Feature	Weight
f_1	1.0	f_7	0.33
f_2	1.0	f_8	0.66
f_3	0.33	f_9	0.33
f_4	0.33	f_{10}	0.33
f_5	0.66	f_{11}	0.33
f_6	0.33		

The summary description of this representation is based on the assumption that all features are important in the representation of the concept. The weight of each feature was computed by dividing the number of times that the feature appears by the number of members.

In order to determine, if a new member is classified in a concept, a process must be executed. For example, in the general features model proposed by Smith & Medin (1981), the classification of a new member is based on determining if the sum of the weights of features that match is greater than or equal to the threshold value. Smith and Medin's general featural model used the classification algorithm shown in Fig. 2.1.

As can be appreciated in this simple example, the strengths of this representation are: (i) the concept representation is not limited to a set of features necessary and sufficient that all members of the concept must share, and (ii) a new member is classified in the concept through a classification process. However, the main challenges that this representation has are: (i) how to determine the features and their weights so that the best represent a concept and (ii) how to establish a classification procedure that is accurate.

The above describes a probabilistic approach. Its main weakness is that when information is converted from the cases to the features, information about typical instances of a class is lost. The following subsection describes exemplar based models, where this information is retained.

Classification algorithm.

Input: Summary description SD and new member i.

Output: boolean variable b, 0=not, 1=yes.

$b = 0$

While i has features {

if f_i matches $f_j \in SD$ {

Add weight to accumulator acc

if $acc > threshold$ then $b = 1$

}

}

Figure 2.1: The classification algorithm of the general features model.

2.1.2 Exemplar based representation

In an exemplar based representation, a concept is described by a collection of exemplars where an exemplar can be an instance or a representation of a subset of instances in the concept. For example, in the above example, the members: chair, bookshelf, and desk that describe the furniture concept are subsets of the furniture concept. Figure 2.2 shows the exemplar based representation of the furniture concept.

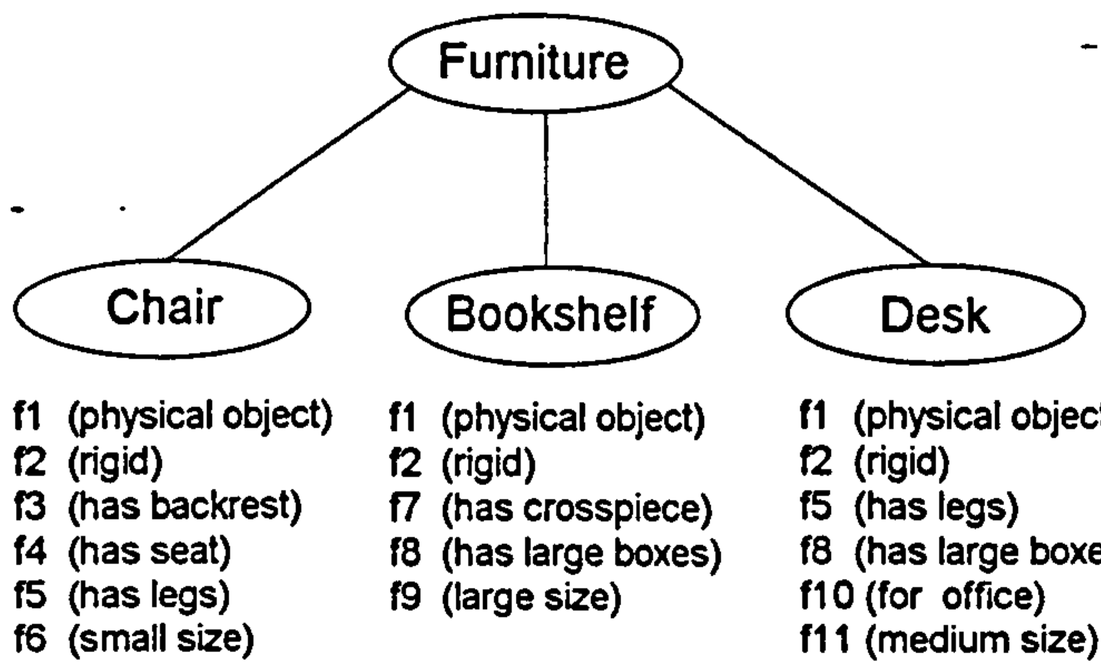


Figure 2.2: Exemplar based representation of the furniture concept.

If a new example needs to be classified in this representation, a match process between the new example and all the exemplars that represent the concept must be done. In the features approach, two exemplars are similar if they have more common features than different ones. That is, the similarity of two exemplars is increased by the number of shared features and decreased with the number of different features [Tversky & Gati 1989]. Then, classifying a new example in this representation depends on the similarity between the new example and the exemplars that represent the concept. If the similarity between the new example and one exemplar in the concept is greater than a threshold value, then the new example belongs to the concept.

The main strengths of this representation are: (i) the concept representation is not limited to a set of necessary and sufficient features that all members of the concept must posses and (ii) a new member is classified in the concept through a matching process. However, its main challenges are: (i) how to determine the exemplars that best represent the concept and (ii) how to establish the similarity measure between two exemplars.

In this thesis, these challenges are tackled by utilising Bayesian networks which are described in the next section.

2.2 Probabilistic Reasoning

Probabilistic reasoning is an approach that it is supported by probability theory. The aim of probability theory is to provide a coherent account of how a belief should change in the light of partial or uncertain information [Pearl 1991]. This section presents one approach for the use of probability theory in AI, namely Bayesian networks that is used in this thesis. Bayesian networks, also known as probabilistic, causal or belief networks, are graphical representations of the dependencies between random variables in a specific application domain. This representation allows the codification of knowledge in the form of dependencies and independencies, and also allows inferences in the form of probabilistic propagation based on a graphical representation.

2.2.1 Basic concepts: Bayes rule

Probability is formally defined as follows [Neapolitan 1990].

Definition 2.1 *Let Ω be the set of outcomes, called sample space, of an experiment, $\mathcal{F}$ a set of events relative to Ω, and P a function which assigns a unique real number to each $A \in \mathcal{F}$. Suppose P satisfies the following axioms:*

$$\begin{aligned} 0 \leq P(A) &\leq 1 \\ P(\Omega) &= 1 \\ P(A \text{ or } B) &= P(A) + P(B) \end{aligned} \tag{2.1}$$

if A and B are disjoint subsets of $\mathcal{F}$. Then the triple $(\Omega, \mathcal{F}, P)$ is called a probability space and P is called a probability measure on $\mathcal{F}$.

In a probability space $(\Omega, \mathcal{F}, P)$, a set of events $\{B_1, B_2, \cdots, B_n\}$ are mutually exclusive and exhaustive if for each $i \neq j$:

$$B_i \cap B_j = NULL \qquad and \qquad \bigcup_{i=1}^{n} B_i = \Omega$$

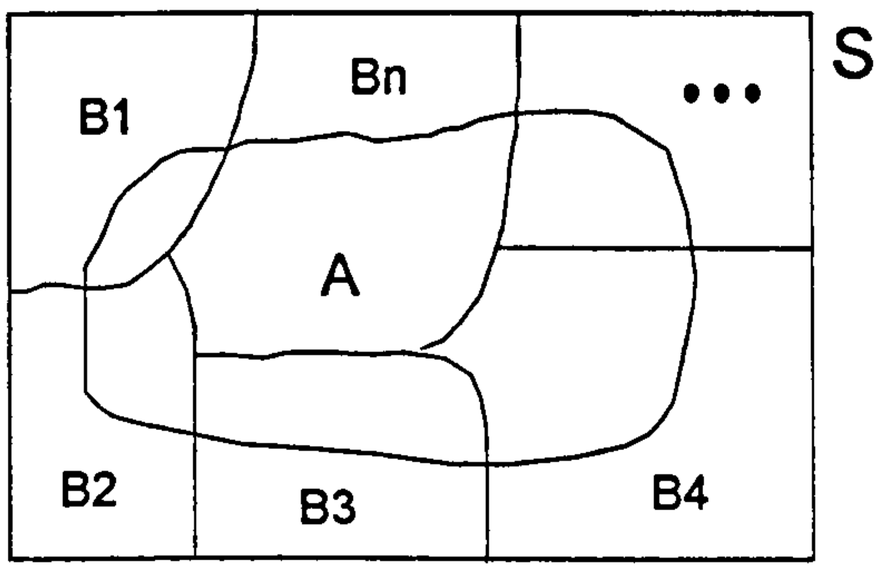


Figure 2.3: An example of events mutually exclusive and exhaustive in a space S.

Now, suppose that $B_1, B_2, \cdots, B_n$ are a set of events mutually exclusive and exhaustive in a probabilistic space S as shown Fig. 2.3. Let A be another event in the same space. Then,

$$A = A \cap S = A \cap (B_1 \cup B_2 \cup \cdots \cup B_n) = (A \cap B_1) \cup (A \cap B_2) \cup \cdots \cup (A \cap B_n)$$

Then, by the third axiom of the definition of probability [1],

$$P(A) = P(A, B_1) + P(A, B_2) + \cdots + P(A, B_n) \tag{2.2}$$

In general, equation 2.2 can be written as:

$$P(A) = \sum_{i}^{n} P(A, B_i) \tag{2.3}$$

Conditional probability is defined with the following equation:

$$P(A \mid B) = \frac{P(A, B)}{P(B)} \tag{2.4}$$

[1] In the following, the notation (A, B) means the conjunction $(A \cap B)$ of events

Where $P(A \mid B)$ is read as the probability of A given B.

Now, from equation 2.3 and the definition of conditional probability in equation 2.4, the following formula is obtained:

$$P(A) = \sum_{i}^{n} P(A \mid B_i)P(B_i) \tag{2.5}$$

This equation, which is known as the *addition rule*, provides the basis for hypothetical reasoning. For example, the probability of an event A is a weighted sum over the probabilities in all the distinct ways that A might be realised.

Another important rule to manipulate events involving probabilities is the *chain rule*. This rule enables one to factor a joint distribution into a product of conditional probabilities. The chain rule is defined as follows.

Given a set of n events, the probability of a joint event $(E_1, E_2, \ldots, E_n)$ can be written as a product of n conditional probabilities:

$$P(E_1, E_2, \ldots, E_n) = P(E_n \mid E_{n-1}, \ldots, E_2, E_1) \ldots P(E_2 \mid E_1)P(E_1) \tag{2.6}$$

Then, applying the chain rule to the joint probability $P(A, B)$ (i.e., $P(A, B) = P(B \mid A)P(A)$), and the definition of conditional probability in equation 2.4:

$$P(A, B) = P(B \mid A)P(A) = P(A \mid B)P(B) \tag{2.7}$$

so the formula called the *Bayes rule* is obtained as:

$$P(A \mid B) = \frac{P(B \mid A)P(A)}{P(B)} \tag{2.8}$$

In probabilistic reasoning, Bayes rule is very useful. For example, suppose an uncertain domain where there is a hypothesis H and evidence E then equation 2.8 gives:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} \tag{2.9}$$

This establishes that the probability of the hypothesis given certain evidence is obtained by multiplying the conditional probability $P(E \mid H)$ by $P(H)$. Both

these probabilities, the conditional probability $P(E \mid H)$ and the hypothesis prior probability $P(H)$ can be obtained from experts or from data based on the previous knowledge. $P(E)$ is a normalising constant. In the following, $P(H)$ is called the *prior probability*, and $P(H \mid E)$ is called the *posterior probability*. This rule can be extended so that a recursive updating of the posterior probability can be made, once new evidence has been obtained. This is calculated with the formula:

$$P(H \mid E(n), E) = P(H \mid E(n)) \frac{P(E \mid E(n), H)}{P(E \mid E(n))} \tag{2.10}$$

where $E(n)$ denotes the evidence observed in the past, and $\dot{P}(H \mid E(n))$ assumes the role of prior probability in order to compute the new posterior $P(H \mid E(n), E)$, i.e., the probability of H given all the past evidence and the new data observed E.

The generalisation of the Bayes rule of equation 2.8, for a set of n mutually exclusive and exhaustive hypotheses $\{H_1, H_2, \ldots, H_n\}$ is referred to as Bayes theorem in the literature and expressed as:

$$P(H_j \mid E) = \frac{P(E \mid H_j)P(H_j)}{\sum_{i=1}^{n} P(E \mid H_i)P(H_i)} \tag{2.11}$$

Although the Bayes theorem and rule (equations 2.11 and 2.8) were very popular in the first expert systems utilised for diagnosis [Gorry & Barnett 1968, Dombal et al. 1974], they are difficult to apply to real problems. This is because they assume that: (i) all the hypothesis are mutually exclusive and exhaustive, and (ii) all the pieces of evidence are conditionally independent from each other given a hypothesis.

These assumptions restrict the expressivity of probabilistic reasoning for more realistic applications. A realistic application is typically interested in looking for relationships among a large number of hypothesis and evidences (variables). Nevertheless, to make probabilistic reasoning on a large set of n variables $X_1, X_2, \ldots, X_n$ requires the definition of a *joint distribution function* $P(X_1, X_2, \ldots, X_n)$ that

would require a table with 2^n entries to store the probability distribution. The next subsection presents the Bayesian network formalism that allows the representation of more realistic applications.

2.2.2 Bayesian networks

A Bayesian network is a graphical model that efficiently encodes the joint distribution of a large number of variables [Heckerman 1995]. A Bayesian network for a set of variables $X_1, X_2, \ldots, X_n$ is formed of:

1. a directed acyclic graph (DAG) that encodes a set of conditional independence assertions about the variables, and

2. a set P of local probability distributions associated with each variable.

First, an explanation of the notation followed in this thesis is given. Capital letters, e.g., X, represent variables while lower case letters designate the values that the variables may have, for example $X = x$ and $Y = y$. Additionally, this subsection presents the set of axioms for the probabilistic relation: X **is independent of** Y **given** Z where X, Y and Z can be a single variable or sets of variables. Second, the relation between probabilistic models and graphical representations of DAGs is established. Finally, this subsection presents a formal description of the properties of Bayesian networks.

Definition 2.2 *Let* **U** *be a finite set of variables with discrete values. Let* X, Y, *and* Z *be three disjoint subsets of variables of* U. X *and* Y *are said to be* **conditionally independent given Z** *if*

$$P(x \mid y, z) = P(x \mid z) \; whenever \; P(y, z) > 0 \tag{2.12}$$

This independence will be denoted as $I(X, Z, Y)$. *Thus,*

$$I(X, Z, Y) \; iff \; P(x \mid y, z) = P(x \mid z) \tag{2.13}$$

where x, y, and z are any assignment of values to the variables in the sets X, Y and Z respectively.

This definition holds in a numeric representation of the probability P. It is interpreted as follows. Knowing the state of Z, the knowledge of Y does not change the belief already gained in X. Now, in order to characterise the conditional independence relation as a logical condition, the following axioms are required[2] [Pearl et al. 1990]:

Symmetry:

$$I(X, Z, Y) \Rightarrow I(Y, Z, X) \tag{2.14}$$

Decomposition:

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z, Y) \;\&\; I(X, Z, W) \tag{2.15}$$

Weak union:

$$I(X, Z, Y \cup W) \Rightarrow I(X, Z \cup W, Y) \tag{2.16}$$

Contraction:

$$I(X, Z \cup Y, W) \;\&\; I(X, Z, Y) \Rightarrow I(X, Z, Y \cup W) \tag{2.17}$$

Intersection (for P strictly positive):

$$I(X, Z \cup W, Y) \;\&\; I(X, Z \cup Y, W) \Rightarrow I(X, Z, Y \cup W) \tag{2.18}$$

These axioms allow the derivation of theorems that may not be obvious from the numerical representation of probabilities. The next step is to relate these axioms with graphical representations.

A directed acyclic graph (DAG) $D = (V, E)$ is characterised by a set of nodes (or vertices) V and a set of edges E that connect certain pairs of nodes in V.

[2]Normal logical operators are needed, e.g., $\Rightarrow$ is the implication, and & is the conjunction.

Nodes in V represent the random variables while the edges or arcs represent conditional dependence relations between the nodes linked. A model M is said to be graphically represented by D if there exists a direct correspondence between the elements in the set of variables U of M and the set of vertices V of D such that the topology of D reflects the properties of M. The correspondence between $I(X, Z, Y)$ and a DAG is made through a separability criterion called *d separation* that is defined as follows.

Definition 2.3 *If X, Y, and Z are three disjoint subsets of nodes in a DAG D, then Z is said to* **d separate** *X from Y, denoted $< X \mid Z \mid Y >_D$ if along every path between a node in X and a node in Y there is a node W satisfying one of the following two conditions: (i) W has converging arrows and none of W or its descendants are in Z, or (ii) W does not have converging arrows and W is in Z.*

For example, consider the DAG of Fig. 2.4. If $X = \{B\}$ and $Y = \{C\}$, they are *d separated* by $Z = \{A\}$ but they are not by a $Z = \{A, E\}$. In both cases, there is one trajectory between B and C, which is through D. Since this unique trajectory has converging arrows, condition (i) is satisfied when $Z = \{A\}$ and it is not satisfied when $Z = \{A, E\}$. In the first case, B and C are *d separated* since $D \in Z$. In the second case, B and C are not *d separated* since $E \in Z$.

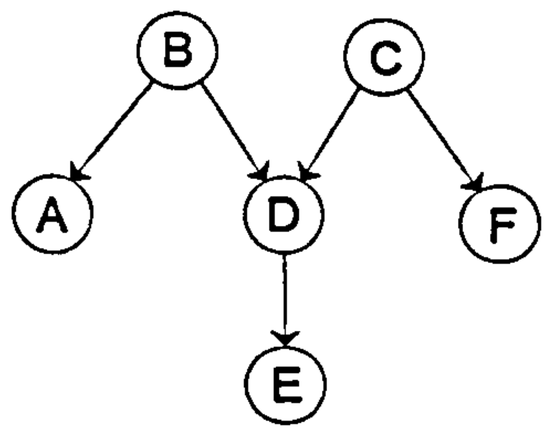


Figure 2.4: A DAG for exemplifying *d separation*.

The following definitions complete the formal description of Bayesian networks.

Definition 2.4 *A DAG D is said to be an* **I map** *of a dependency model M if every d separation condition displayed in D corresponds to a valid conditional independence relationship in M, i.e., if for every three disjoint sets of nodes X, Y, and Z, the following holds:*

$$< X \mid Z \mid Y >_D \Longrightarrow I(X, Z, Y)_M. \tag{2.19}$$

A DAG is a **minimal I map** of M if none of its arrows can be deleted without destroying its *I mapness.*

Definition 2.5 *Given a probability distribution P on a set of variables V, a DAG $D = (V, E)$ is called a* **Bayesian network** *of P iff D is a minimal I map of P.*

In other words, given a set of variables with a probabilistic model P, a Bayesian network is a graphical representation which permits the representation of the dependencies and independencies between the variables. The structure of the network represents knowledge about the variables of the process. This knowledge consists of two sets of probabilities: (i) conditional probabilities of every node given all its parents, and (ii) prior probabilities of the root nodes. Figure 2.5 presents an elementary Bayesian network and its relation with Bayes rule (equation 2.8). In this case, the hypothesis happens to be the root node, and the evidences are the leaf nodes but this is not a restriction in Bayesian networks. In this case, prior probabilities $P(H)$ are required in the roots of the networks. The other nodes require an associated matrix of conditional probabilities between each one of them and their parents (the upper extreme of the arcs). Thus, the evidence nodes are observed, and the question is to *infer* the new value of the

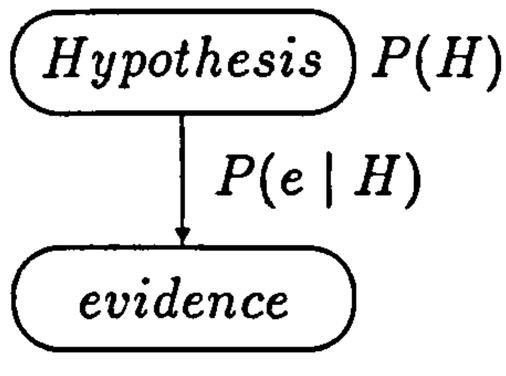


Figure 2.5: A simple Bayesian network.

probability of the hypothesis, i.e., $P(H \mid e)$. Notice that, in general a node can contain either evidence node or be a hypothesis. Different algorithms have been developed to propagate these probabilities given new evidence.

Beyond the definitions, several theorems have been published in order to formalise Bayesian networks (e.g. [Geiger & Pearl 1988], [Geiger et al. 1989]). The following theorem, called *Strong completeness* [Geiger & Pearl 1988] includes many of the previous theorems and legitimises the use of DAGs as a language for representing probabilistic dependencies. The complete proofs can be found in the indicated reference.

Theorem 2.1 *Strong completeness*

For every DAG D, there exists a distribution P such that for every three disjoint sets of variables X, Y, and Z the following holds:

$$< X \mid Z \mid Y >_D \ iff \ I(X, Z, Y)_P \tag{2.20}$$

Summarising the formal definition of Bayesian networks, definition 2.2 introduces the notion of conditional independence and establishes the notation $I(X, Z, Y)$. In graphical representations, definition 2.3 establishes a condition that holds between nodes (or subsets of nodes) in a directed graph. Definition 2.4 relates the notion of conditional independence $I(X, Z, Y)$ in a model M with the *d separation* property of directed graphs. Next, definition 2.5 explains what a

Bayesian network is. Finally, the theorem 2.1 states that Bayesian networks can be used to perform probabilistic reasoning mechanism in a sound manner.

It is important now to distinguish three kinds of Bayesian networks:

Trees: This is a DAG where any node can have at most one parent. Figure 2.6(a) shows a typical network considered as a tree.

Singly connected networks (polytrees): This is a DAG which contains one and only one path between any pair of nodes in the network. An example is shown in Fig. 2.6(b).

Multiply connected networks: This is a DAG without the restrictions of trees or polytrees. Figure 2.6(c) is multiply connected since there are two paths between two nodes.

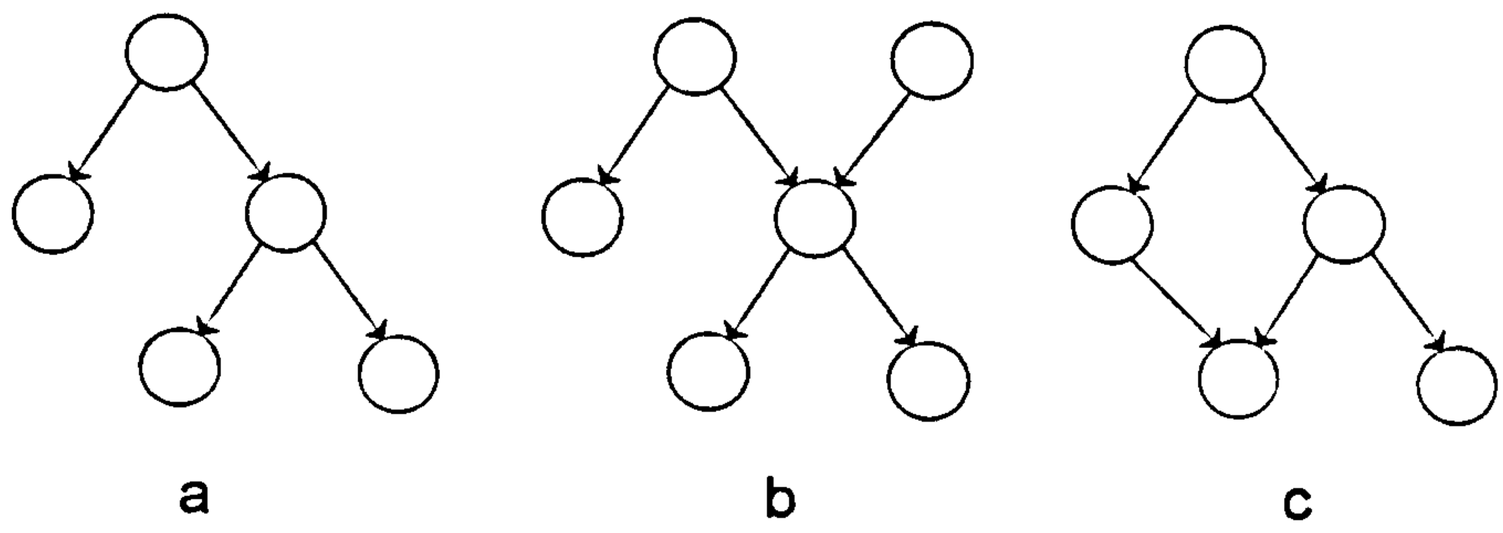


Figure 2.6: Examples of Bayesian networks. (a) is a tree, (b) is singly connected and (c) is multiply connected.

The multiply connected network is the most general and expressive when modelling specific processes. However, propagation (and therefore, reasoning in multiply connected networks) is known to be NP hard [Cooper 1990]. Trees and singly connected networks are less expressive but the probability propagation algorithms for them are more efficient than multiply connected networks.

Although, the proposed model in this thesis uses a multiply connected Bayesian network to determine the best exemplars, the probability propagation method in

a singly connected network is also explained since it facilitates understanding of the complex algorithm for propagation in multiply connected networks. So, the following two subsections describe probability propagation methods for these networks.

2.2.3 Probability propagation in a singly connected network

In order to understand the propagation algorithm for singly connected networks, consider a typical node as shown in Fig. 2.7.

This figure shows a node X that has m parents $Z_1, \ldots, Z_m$ and n children $Y_1, \ldots, Y_n$. Consider that the node X can take k discrete values $x_1, x_2, \ldots, x_k$. Now, suppose that some nodes have been instantiated, i.e., their values have been observed. Let $e = e_X^- \cup e_X^+$ denote the evidence, where e_X^- represents the evidence contained in the subtree rooted at X, and e_X^+ represents the evidence from the rest of the network. In Fig. 2.7, the subtree rooted at X is a portion of the network containing only the nodes X and all its descendents Y_i. The rest of the network corresponds to the structure formed by all nodes minus $X \cup descendents(X)$.

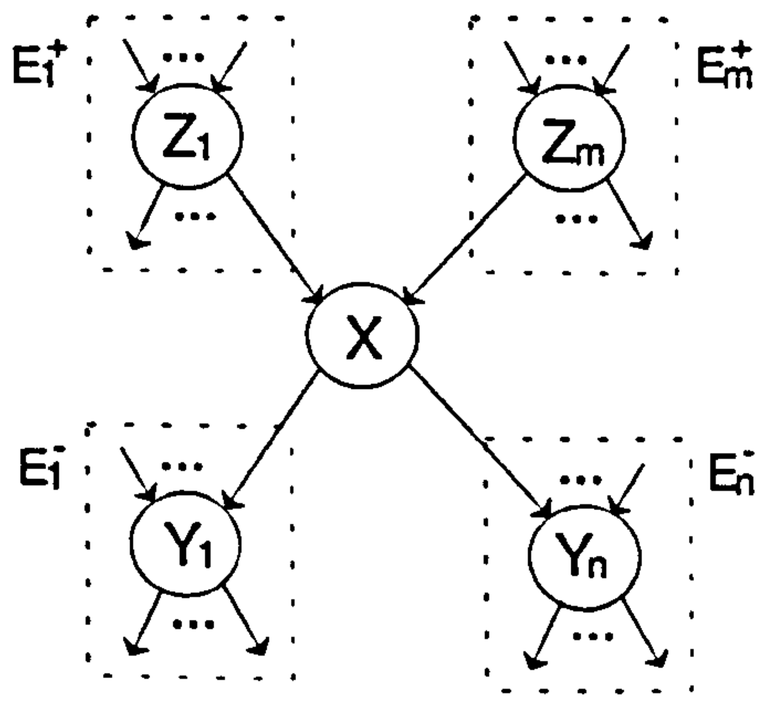


Figure 2.7: A typical node in a singly connected network.

Thus, the goal is to find the posterior probability $P'(x)$ given the observed evidence e, i.e.,

$$P'(x) = P(x \mid e) = P(x \mid e_X^-, e_X^+)$$

Using Bayes rule gives

$$P'(x) = \frac{P(e_X^-, e_X^+ \mid x)P(x)}{P(e_X^-, e_X^+)}$$

Since e_X^- and e_X^+ are independent given X, this becomes

$$P'(x) = \frac{P(e_X^- \mid x)P(e_X^+ \mid x)P(x)}{P(e_X^-, e_X^+)}$$

by Bayes rule again and the definition of conditional probability,

$$P'(x) = \frac{P(e_X^- \mid x)P(x \mid e_X^+)P(e_X^+)}{P(e_X^-, e_X^+)}$$

$$P'(x) = \alpha P(e_X^- \mid x)P(x \mid e_X^+) \tag{2.21}$$

where $P'(x)$ represents the posterior probability of $X = x$ given all the evidence provided, and $\alpha = [P(e_X^- \mid e_X^+)]^{-1}$ is a normalising constant to obtain $\sum_x P'(x) = 1$.

Notice that this formula corresponds to a vector, with one element for each possible value of X. Now, let the following functions be defined:

$$\lambda(x) = P(e_X^- \mid x) \tag{2.22}$$

and

$$\pi(x) = P(x \mid e_X^+) \tag{2.23}$$

Vector $\lambda(X)$ represents the *diagnostic support* that node X receives from its

descendants, while $\pi(X)$ represents the *causal support* attributed by all non descendants of X, and received through its parents. Then, the updated belief in $X = x$ can be obtained by *fusing* these two supports and equation 2.21 becomes:

$$P(x \mid e) = \alpha\lambda(x)\pi(x) \tag{2.24}$$

Since $\lambda(x)$ represents the support that X receives from all its descendants, it is necessary to fuse the support from each one of its descendants. For example in Fig. 2.7, $\lambda(x)$ corresponds to the evidence provided by nodes $Y_1, \ldots, Y_n$. Thus, equation 2.22 can be rewritten as:

$$\begin{aligned}\lambda(x) &= P(e_X^- \mid x) \\ &= P(e_{Y_1}^-, \ldots, e_{Y_n}^- \mid x) \\ &= P(e_{Y_1}^- \mid x) \ldots P(e_{Y_n}^- \mid x)\end{aligned} \tag{2.25}$$

since $e_{Y_1}^-, \ldots, e_{Y_n}^-$ are conditionally independent given x. Furthermore, renaming these terms as:

$$\lambda_{Y_i}(x) = P(e_{Y_i}^- \mid x) \quad 1 \leq i \leq n \tag{2.26}$$

then, equation 2.25 can be expressed as:

$$\lambda(x) = \prod_{i=1}^{n} \lambda_{Y_i}(x) \tag{2.27}$$

Similarly, the causal support that X receives from its non descendants, through its parents $Z_1, \ldots, Z_m$ (equation 2.23) can be expressed as:

$$\begin{aligned}\pi(x) &= P(x \mid e_X^+) \\ &= P(x \mid e_{Z_1}^+, \ldots, e_{Z_m}^+) \\ &= \sum_{z_1,\ldots,z_m} P(x \mid z_1, \ldots, z_m) P(z_1, \ldots, z_m \mid e_X^+) \\ &= \sum_{z_1,\ldots,z_m} P(x \mid z_1, \ldots, z_m) \prod_{i=1}^{m} \pi_X(z_i)\end{aligned} \tag{2.28}$$

where $P(x \mid z_1, \ldots, z_m)$ will be an element of the matrix obtained as previous knowledge, and stored in the node X. $\pi_X(z_i) = P(a \mid e^+_{Z_i})$ is calculated in node Z_i and sent as causal support to X. Thus, substituting equations 2.27 and 2.28 in equation 2.24, the following is obtained:

$$P'(x) = \alpha \prod_{i=1}^{n} \lambda_{Y_i}(x) \sum_{z_1,\ldots,z_m} P(x \mid z_1, \ldots, z_m) \prod_{i=1}^{m} \pi_X(z_i) \qquad (2.29)$$

Equation 2.29 summarises Pearl's algorithm for probability propagation. It is best known as the message passing algorithm since $\lambda_{Y_i}(x)$, and $\pi_X(z_i)$ can be seen as messages that other nodes send to node X in order to update its probability vector. Thus, this posterior probability can be calculated from the previous knowledge $P(x \mid z_1, \ldots, z_m)$, the messages $\lambda_{Y_i}(x)$ from its children and a message $\pi_X(z_i)$ from its parent Z_i.

The detailed algorithm can be consulted in the book by Pearl (1988), and easily readable in the book by Neapolitan (1990).

2.2.4 Probability propagation in trees of cliques

This subsection presents an approach for probability propagation in multiply connected networks called propagation in trees of cliques [Lauritzen & Spiegelhalter 1988]. Other algorithms for propagation in networks are given by Cooper (1984), and by Horvitz et al. (1989). The propagation algorithm presented in this subsection is used in Chapter 3. A reader already familiar with this propagation algorithm may skip this subsection.

The basis of this method is the following formula[3]:

$$P(V) = K \prod_{i=1}^{p} \psi(W_i) \qquad (2.30)$$

where V designates a finite set of propositional variables, and P represents a joint probability distribution of V. K represents a constant and let $\{W_i$ such that

[3]The material of this section was taken from the book by Neapolitan (1990).

$1 \leq i \leq p\}$ be a collection of subsets of V. Also, ψ is a function which assigns a unique real number to every combination of values of the propositional variables in W_i. Then ($\{W_i$ such that $1 \leq i \leq m\}$,ψ) is called a potential representation of P, and W_i are called *cliques*. A clique is defined as a subset of nodes in which every pair of nodes of the clique is connected. Also, the subset must be maximal, i.e., there is no other complete set which is a subset [Golumbic 1980].

The algorithm developed by Lauritzen & Spiegelhalter (1988) indicates: (i) how to obtain the collection W_i of subsets of V, and (ii) how to compute the functions $\psi(W_i)$. In other words, this method modifies the original multiply connected network in order to obtain a tree of cliques, from which probability propagation can be made utilising the functions $\psi(W_i)$. This propagation is similar to Pearl's algorithm for trees, which is a subset of the algorithm described in Subsection 2.2.3. The following subsections describe these two parts of the algorithm.

Obtaining a tree of cliques

The cliques W_i of equation 2.30 must follow a series of conditions. The procedure of Fig. 2.8 obtains the set of cliques with the required properties.

These steps are better explained with the aid of an example taken from the book by Neapolitan (1990).

Figure 2.9 presents the original Bayesian network. Notice that it is multiply connected since there is more than one path between node F and H. This network requires, as all the Bayesian networks, the prior probability of the roots and the conditional probability matrices of the other nodes given their parents. The first step in the procedure of Fig. 2.8 is trivial, i.e., only delete the direction of the arcs. The second step, the moralization, is obtained when the pairs of parents of all nodes (if they exist) are *married*. This is done with the addition of an

1. Delete the direction of the arcs, i.e., the DAG is converted to an undirected acyclic graph.
2. *Moralize* the graph.
3. Triangulate the graph.
4. Order the nodes according to a criterion called the maximum cardinality search.
5. Determine the cliques of the triangulated graph.
6. Order the cliques according to their highest labelled vertices to obtain an ordering of the cliques with the running intersection property.

Figure 2.8: Procedure to convert a network in a tree of cliques.

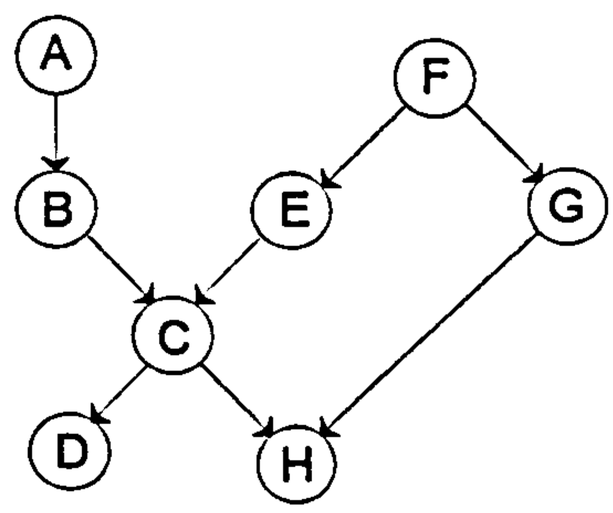


Figure 2.9: Original multiply connected network.

arc between these parent nodes. Figure 2.10 presents the moral DAG which is obtained by adding the arc between nodes B and E (parents of C), and the arc between C and G (parents of H).

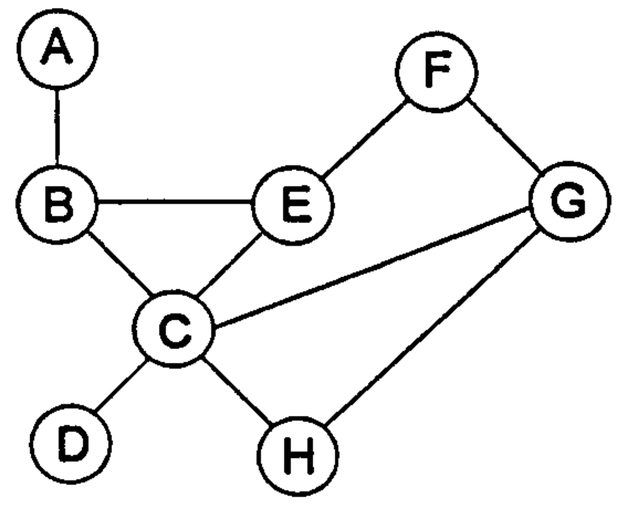


Figure 2.10: Undirected moralized graph.

Next, the triangulation step takes place. An undirected graph is called triangulated if every simple cycle of length strictly greater that 3 possesses a chord. In the original network, after the moralization, the nodes $[F, E, C, G]$ form a simple cycle of size 4. Thus, in order to triangularize the undirected graph, the arc between E and G is added. Figure 2.11 shows the triangularized graph.

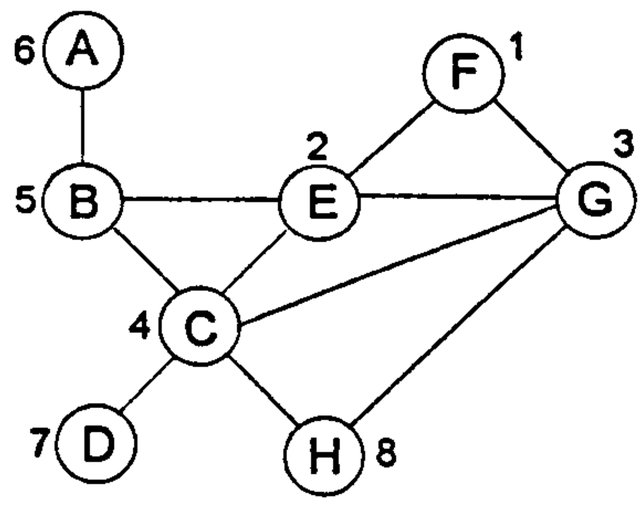


Figure 2.11: Triangulated and ordered undirected graph.

This figure also show the ordering step indicated in the procedure of Fig. 2.8 which is now explained. An order of the nodes, according to a criterion known as the maximum cardinality search, is obtained as follows. First, 1 is assigned to an arbitrary node. To number the next node, select a node that is adjacent to the largest numbered node, which breaks the arbitrary ties. In Fig. 2.11, number

1 was assigned to node F. Number 2 has to be assigned to one of the adjacent nodes to F, i.e., E or G (two nodes are adjacent if there is an arc between them). Node E was chosen arbitrarily. Number 3 was assigned to node G (could be B or C), and so on until all the nodes are numbered. The next step, to determine the cliques of the triangulated graph, is now described. A clique is a subset of nodes which is complete, i.e., every pair of nodes of the clique is adjacent. Also, the subset must be maximal, i.e., there is no other complete set which is a subset. In the triangulated graph of Fig. 2.11, the following cliques are found: $\{A, B\}$, $\{B, E, C\}$, $\{E, G, F\}$, $\{C, D\}$, $\{E, C, G\}$, and $\{C, G, H\}$.

Finally, the ordering of the cliques is required. An ordering $[Clq_1, Clq_2, \ldots, Clq_p]$ of the cliques has the running intersection property if for every $j > 1$ there exists an $i < j$ such that

$$Clq_j \cap (Clq_1 \cup Clq_2 \cup \ldots \cup Clq_{j-1}) \subseteq Clq_i. \tag{2.31}$$

In the example, an ordering of the cliques according to their highest number is the following: $Clq_1 = \{E, G, F\}$, $Clq_2 = \{E, C, G\}$, $Clq_3 = \{B, E, C\}$, $Clq_4 = \{A, B\}$, $Clq_5 = \{C, D\}$, and $Clq_6 = \{C, G, H\}$. This ordering has the running intersection property. For example:

$$Clq_4 \cap (Clq_1 \cup Clq_2 \cup Clq_3) = \{B\} \subseteq Clq_3 \tag{2.32}$$

$$Clq_5 \cap (Clq_1 \cup Clq_2 \cup Clq_3 \cup Clq_4) = \{C\} \subseteq Clq_2$$

Before defining the structure of the tree of cliques, two parameters need to be defined.

$$S_i = Clq_i \cap (Clq_1 \cup Clq_2 \cup \ldots \cup Clq_{i-1})$$

$$R_i = Clq_i - S_i.$$

These parameters will be used in the propagation of probabilities and in the

definition of the structure. As an example, $S_4 = \{A, B\} \cap \{B, C, E, G, F\} = \{B\}$ and $R_4 = \{A, B\} - S_4 = \{A\}$.

Once the set of ordered cliques has been obtained, the next step is the definition of the structure of the tree of cliques. The first clique is the root of the tree. Now, for the rest of the nodes, i.e., for each i such that $2 \leq i \leq p$, there exists at least one $j < i$ such that

$$S_i = Clq_i \cap (Clq_1 \cup Clq_2 \cup \ldots \cup Clq_{i-1}) \subseteq Clq_j. \tag{2.33}$$

Then Clq_j is a parent of Clq_i. In the case of more than one possible parent, the choice is arbitrary.

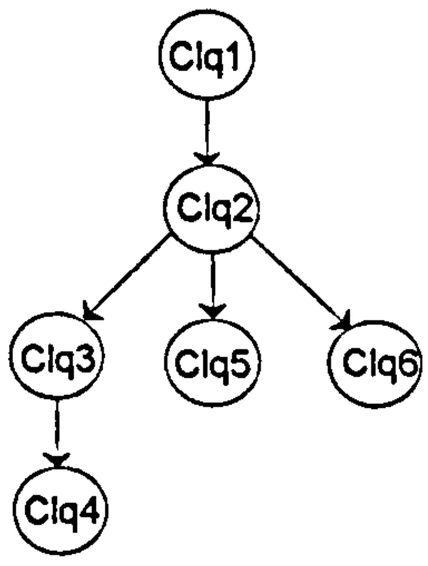


Figure 2.12: Resultant tree of cliques.

Figure 2.12 shows the final modification of the Bayesian network of Fig. 2.9 into the tree of cliques. The next subsection briefly describes the algorithm for probability propagation in this tree of cliques.

Probability propagation

The cliques obtained in the previous subsection are the W_i subsets indicated in equation 2.30. The functions ψ are defined by the following theorem.

Theorem 2.2 *Let G be the DAG representing a Bayesian network, G_m the moral graph relative to G, G_u a graph formed by triangulating G_m as discussed in the*

previous subsection. Let $\{Clq_i$ *such that* $1 \leq i \leq p\}$ *be the cliques of* G_u*. For each node* $v \in V$*, assign a unique clique* Clq_i *such that*

$$v \cup parents(v) \subseteq Clq_i. \tag{2.34}$$

This is always possible, since parents of the original graph are married and therefore $\{v\} \cup parents(v)$ *is a complete set in* G_m *and thus in* G_u*. If a complete set is a subset of more than one clique, choose one of them arbitrarily but keeping each node* v *assigned to only one clique. Denoting as* $f(v)$ *the clique assigned to* v*, and for* $1 \leq i \leq p$*,*

$$\psi(Clq_i) = \prod_{f(v)=Clq_i} P(v \mid parents(v)). \tag{2.35}$$

where $f(v) = Clq_i$ *represents only the nodes* v *that are assigned to* Clq_i*. If there is no* v *represented in the clique, it is assigned the value 1. Then*

$$(\{Clq_i \text{ such that } 1 \leq i \leq p\}, \psi) \tag{2.36}$$

is a potential representation of P*.*

The complete proof can be found in the text by Neapolitan (1990). The function $parents(v)$ represents the set of nodes which are parents of node v in the original network.

For example, assigning A and B to the clique $\{A, B\}$, C to the clique $\{B, E, C\}$, D to the clique $\{C, D\}$, E, F and G to the clique $\{E, G, F\}$, and H to the clique $\{C, G, H\}$:

$$\begin{aligned}
\psi(A, B) &= P(B \mid A)P(A) \\
\psi(B, E, C) &= P(C \mid B, E) \\
\psi(C, D) &= P(D \mid C) \\
\psi(E, G, F) &= P(E \mid F)P(G \mid F)P(F) \\
\psi(C, G, H) &= P(H \mid C, G) \\
\psi(E, C, G) &= 1.
\end{aligned} \tag{2.37}$$

When the new tree is defined, it is ready to accept the instantiation of variables as evidence, and to compute the posterior probability of all the nodes through probability propagation in the tree of cliques. This is done in a similar way to the algorithm of message passing for trees and polytrees described in Subsection 2.2.2.

The λ message that a node sends to its parents is calculated with the formula:

$$\lambda_{Clq_i}(S_i) = \sum_{R_i} \psi(Clq_i) \tag{2.38}$$

where the sum is made over all the possible values of the variables in the set R_i

The π message that the nodes send to their children is computed as:

$$\pi_{Clq_j}(S_i) = \sum_{Clq_j - S_i} P'(Clq_j) \tag{2.39}$$

where the sum is made of all the possible values of the variables in the set $Clq_j - S_i$.

The ψ function is updated when a clique Clq_j receives a λ message from its child Clq_i as:

$$\psi(Clq_j) = \lambda(S_i)\psi(Clq_j) \tag{2.40}$$

For the root clique, the posterior probability once that all the λ messages have been received from its children is given by

$$P'(Clq_{root}) = \psi_{root}(Clq_{root}) \tag{2.41}$$

Finally, the posterior probability of a single variable, when the probabilities of all the cliques have been determined, is calculated using the formula:

$$P'(v) = \sum_{\substack{w \in Clq_i \\ w \neq v}} P(Clq_i). \tag{2.42}$$

The complete algorithm can be consulted in the book by Neapolitan (1990).

2.2.5 Probabilistic causal method

Figure 2.13 shows a network known as the probabilistic causal model. It consists of a two level DAG where the roots are considered the causes of the manifestations

of the leaf nodes. Although the network looks to be simple, it presents several problems in the definition of its conditional probabilities. The probabilistic causal model was first studied by Peng & Reggia (1994), and then further developed by Pearl (1988) and by Neapolitan (1990).

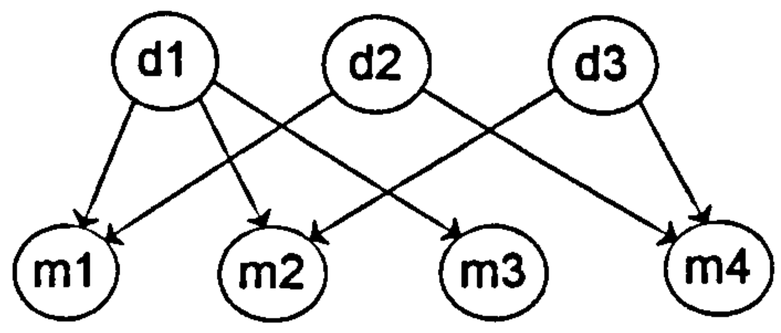


Figure 2.13: A DAG representing a probabilistic causal model.

In this network, $D = \{d_1, d_2, d_3\}$ represents the set of diseases, and $M = \{m_1, m_2, m_3, m_4\}$ represents the set of manifestations[4] respectively.

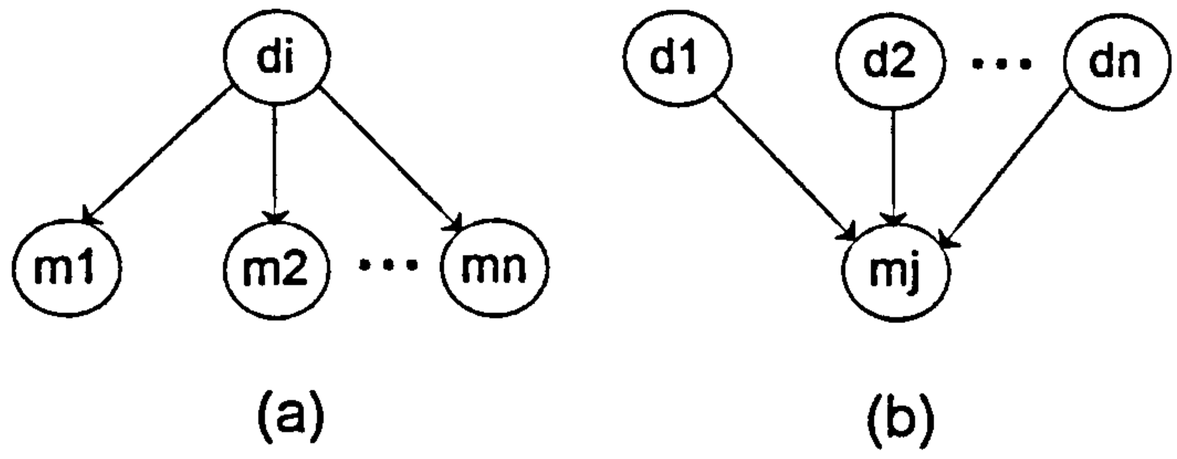


Figure 2.14: Causal relation between hypotheses or causes, and manifestations.

Notice that there are two types of relationships between the nodes in Fig. 2.13. Figure 2.14(a) shows a common relationship where one disease has many manifestations. However, Fig. 2.14(b) shows a relation where one manifestation can be caused by several diseases. For example, the high fever event in medicine is caused by many different diseases, e.g., influenza, tuberculosis, and kidney infection. Any of these diseases is likely to cause high fever, but the presence of two of these diseases is only more likely to cause fever. This relation between a manifestation and several causes is known as the *noisy or* since it remains the *or*

[4]The names are traditionally taken from the medical domain.

gate utilised in digital electronics. In the probabilistic case, the noisy or relation is used when any member of a set of diseases is likely to cause a specific event, but this likelihood does not significantly change when a patient suffers several of these diseases.

One of the problems in these kinds of networks is the initialisation of the network with the prior and conditional probabilities. Normally, the conditional probability for describing the arcs of the network in Fig. 2.14(b) contains 2^n independent parameters. It would be very difficult for a physician to estimate the probability of high fever given influenza, no tuberculosis and infection, or the probability of no influenza nor tuberculosis but with infection, and so on with the 8 combinations. A method for computing the conditional probability matrix of a disease given a set of manifestations is now explained. This method is based on the following two assumptions:

Accountability. An event m_j is false, $P(m_j) = 0$, if all conditions listed as causes of m_j are false.

Exception independence. If an event m_j is a consequence of two conditions d_1 and d_2, then the inhibition of the occurrence of m_j under d_1 is independent of the mechanisms of inhibition of m_j under d_2.

Consider the example mentioned above. Influenza alone is a cause of high fever unless an inhibitor is present. If tuberculosis alone also causes fever except when another inhibitor is present, then the exception independence mechanism assumes that both these inhibitors are independent. Then, let q_{ij} denote the probability that a manifestation m_j is inhibited when only disease d_i is present, i.e., $q_{ij} = P(\neg m_j \mid d_i \; alone)$. Then, by the exception independence assumption:

$$P(\neg m_j \mid d_1, d_2, \ldots, d_n) = P(\neg m_j \mid d_1)P(\neg m_j \mid d_2) \ldots P(\neg m_j \mid d_n)$$

$$= \prod_{i:d_i=true} q_{ij}$$

In general, let **d** be the set of assignments of the set of diseases, and let $T_d = \{i : d_i = true\}$, i.e., the set of all diseases actually present. Then, the conditional probability matrix can be calculated with the following formula:

$$P(m_j \mid \mathbf{d}) = \begin{cases} \prod_{i \in T_d} q_{ij} & \text{if } \neg m_j \\ 1 - \prod_{i \in T_d} q_{ij} & \text{if } m_j \end{cases} \tag{2.43}$$

For example, in the network of Fig. 2.13, the following are the formulas of equation 2.43:

$$\begin{aligned} P(\neg m_1 \mid +d_1, +d_2) &= q_{11}q_{21} \\ P(\neg m_1 \mid +d_1, \neg d_2) &= q_{11} \\ P(\neg m_1 \mid \neg d_1, +d_2) &= q_{21} \\ P(\neg m_1 \mid \neg d_1, \neg d_2) &= 1. \end{aligned}$$

where $+d_i$ means $d_i = true$ and the quantities for m_1 are 1 minus the conditional for $\neg m_1$.

These equations will be utilised to obtain the parameters needed in the probability initialisation of the proposed model in the thesis, which is described in Chapter 3.

2.3 Summary

This chapter presented the background knowledge required to follow the techniques developed in this thesis. Section 2.1 presented two approaches to concept representation: (i) the probabilistic representation described in Subsection 2.1.1 and (ii) the exemplar based representation described in Subsection 2.1.2. A combination of these two representations is utilised in the probabilistic exemplar based model described in Chapter 3. Section 2.2 presented the probabilistic theory that supports the probabilistic exemplar based model. Subsection 2.2.1

presented the basis of probability theory until the deduction of Bayes rule (equation 2.8). Subsection 2.2.2 described what a Bayesian network is and presented the axioms and theorems that allow the utilisation of DAGs as a language for knowledge representation and inference. Subsection 2.2.3 presented a brief description of the propagation algorithms for singly connected networks. Subsection 2.2.4 described the algorithm for probability propagation in multiply connected networks.

The propagation method for multiply connected networks (trees of cliques) will be utilised in Chapter 3 as the probability propagation method of the probabilistic exemplar based model. Subsection 2.2.5 described a technique for the computation of conditional probabilities in causal models utilised in diagnosis. This technique will also be used in Chapter 3.

The next chapter presents the utilisation of the techniques presented in this chapter, for the development of a probabilistic exemplar based model. The model addresses the issues of retrieval, storing and learning in case based reasoning.

Chapter 3

A PROBABILISTIC EXEMPLAR BASED MODEL

The previous two chapters of the thesis provide the motivation and theory for developing a probabilistic exemplar based model whose foundations are provided by Bayesian networks. This chapter develops such a model. First, in Section 3.1 the problems of developing a probabilistic exemplar based model are described. Next, Section 3.2 describes the knowledge representation used by the proposed model. Then, the problems raised are tackled in Section 3.3, which develops the classification process, and Section 3.4 which develops the learning process. The chapter concludes with an illustrative example in Section 3.5 and a summary in Section 3.6.

3.1 The Problem

Chapter 1 provided the basis for the thesis by arguing that most current CBR tools are unable to cope with domains where knowledge is not predefined, may have varying features, and which contain uncertainties. This motivation leads to the following problem definition.

Given a set of cases of a weak domain where:

1. the categories or concepts are difficult to define by necessary and sufficient features,

2. the categories can be non-disjoint,

3. the data are not structured,

4. all the data do not exist in advance, and

5. there is uncertainty in how the categories are represented by cases.

then, the problem is to develop an exemplar based model that addresses the classification and learning issues.

That is, given an existing exemplar based model, how can it be used to determine the category of a new case. Further, given a sequence of training cases, which exemplar based model best represents the domain? Since in practice, not all the data are available in advance, the developed model must be incremental and its accuracy must improve as more data become available. Of course, the developed model should have good foundations.

In order to provide some insight into the problem, consider the diagram shown in Fig. 3.1. Figure 3.1 shows a weak domain in which there are two categories A and B (solid lines).

The category A has nine cases (the points) $c_1, c_2, c_3, c_4, c_6, c_7, c_8, c_9$, and c_{10}, and the category B has five cases c_3, c_4, c_5, c_9, and c_{11}. Note that the cases c_3, c_4 and c_9 are common cases that occur in both categories.

The main problem is to proceed from a view like the one shown in Fig. 3.1 to an exemplar based view like the one shown in Fig. 3.2 where the exemplars e_6, e_8, e_9, and e_{11} represent sets of similar cases (dashes lines). That is, instead of storing all the cases, only the prototypical cases are stored. Although conceptually, this is an elegant idea, attempting to develop it raises the following difficult questions:

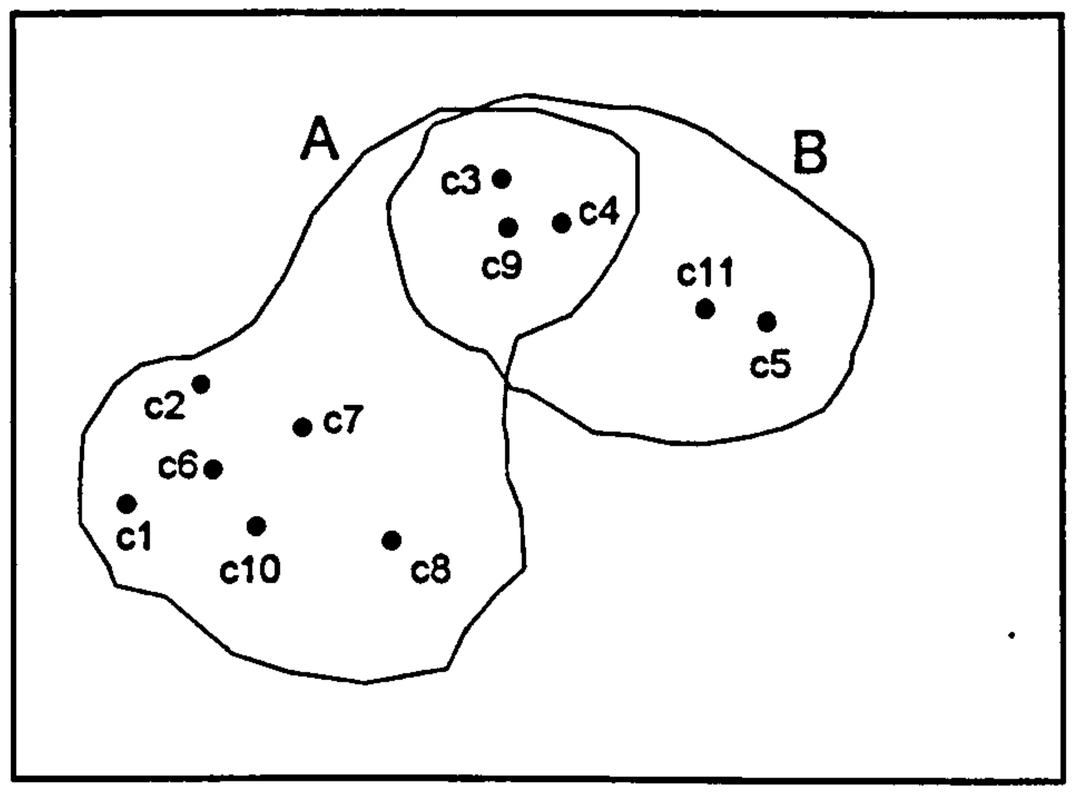


Figure 3.1: Example of a weak domain.

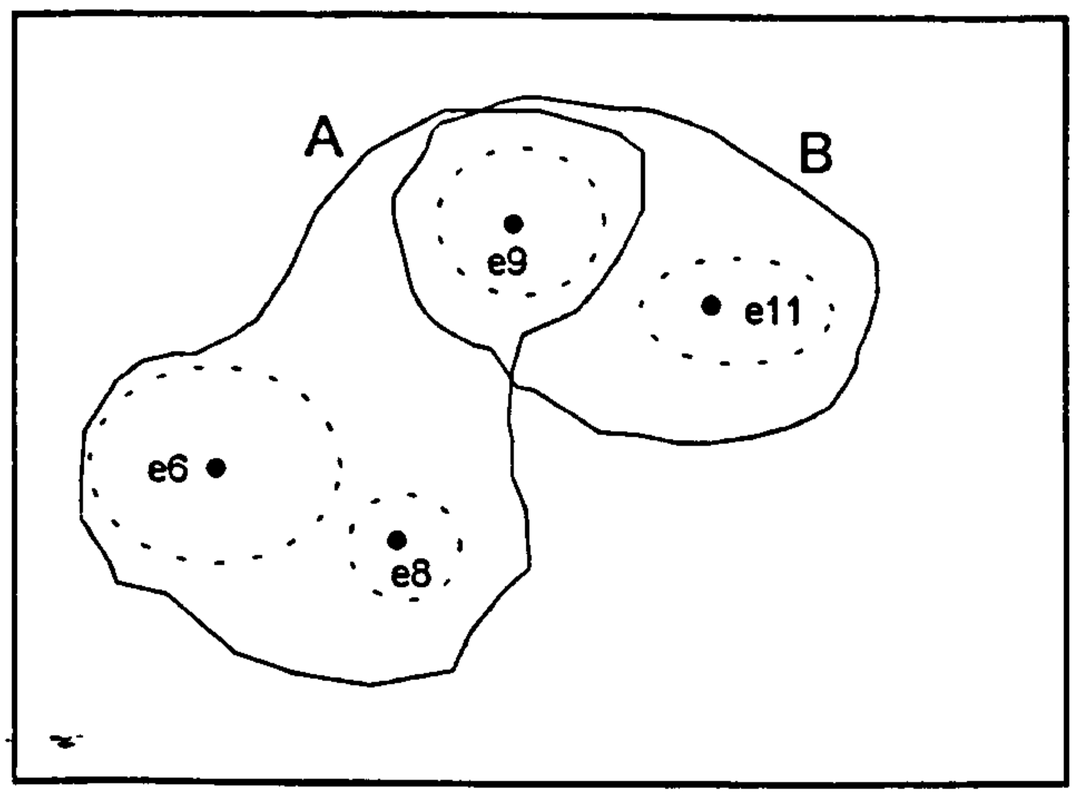


Figure 3.2: Exemplar based view in weak domain.

1. What is a good representation of the model?

2. How can a new case be classified?

3. What notion of similarity can be adopted?

4. What makes a good exemplar?

5. How can the model be learned incrementally?

The following sections of this chapter develop the model by addressing these questions.

3.2 The Knowledge Representation

One way of representing the information in Fig. 3.2 is to use a network in which nodes can be used to denote exemplars, features, and categories. Thus, Fig. 3.3 shows the network representing the exemplar based model shown in Fig. 3.2. In this representation, the dashed lines show the relationship between categories and exemplars, and the solid lines show the relationship between exemplars and their features. So for example, category A has the exemplars e_6, e_8, and e_9 and exemplar e_6 has the features f_1, f_2, and f_3. Notice that exemplars can be shared by categories, and features can be shared by exemplars.

As it stands, Fig. 3.3 is not an adequate representation of an exemplar based model since it does not contain any information about the degree of dependency between a category and its exemplars and an exemplar and its features. So for example, a car can have features such as colour, engine, and make. But, which of them is more relevant in the representation of a car? The above representation would not differentiate between the strong dependency: an object being a car and having an engine, and the weak dependency: an object being a car and its colour.

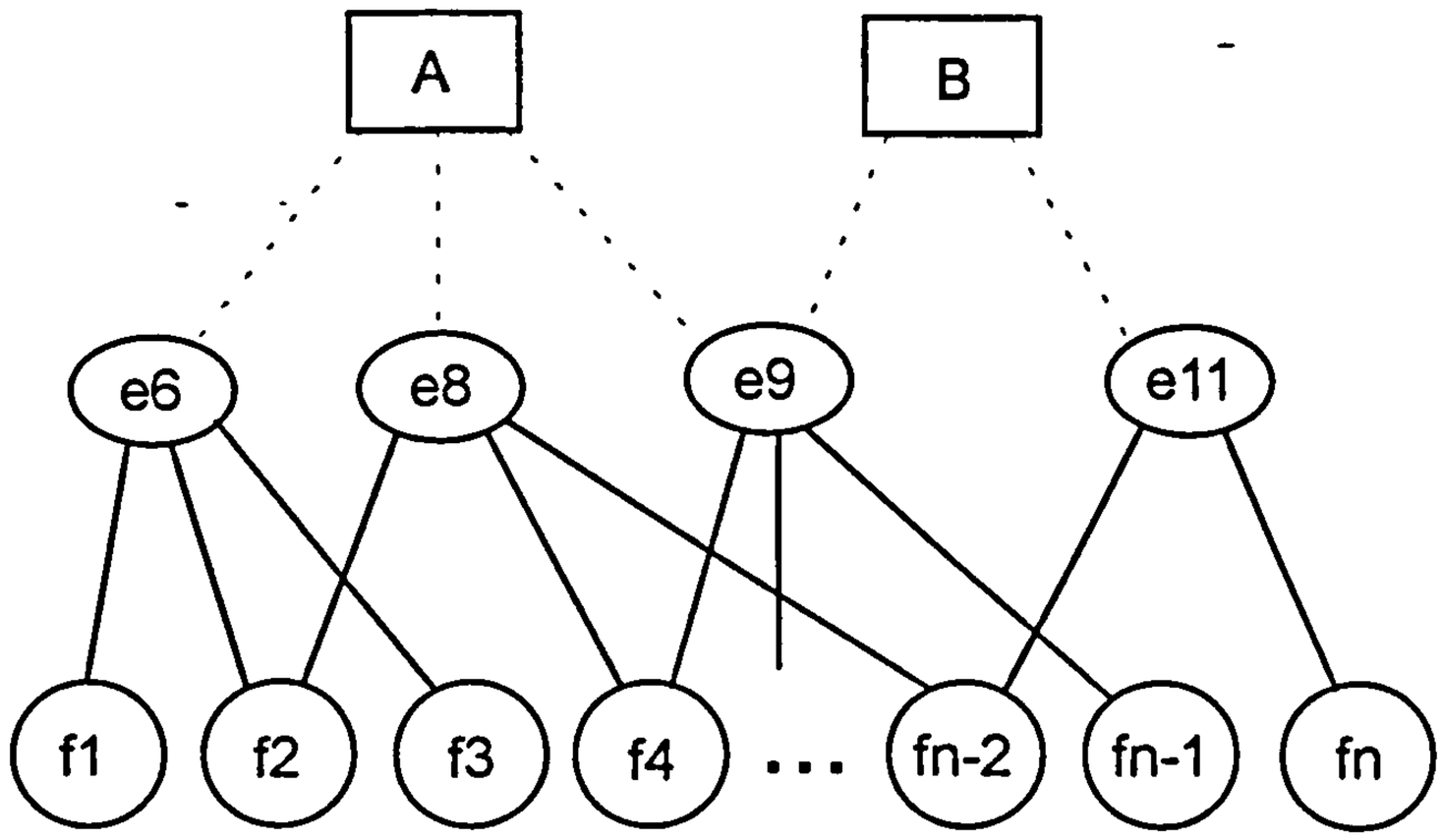


Figure 3.3: A basic exemplar based representation.

Hence, to include the strengths of such dependencies, the relationships between exemplars and features are represented as probabilistic dependencies. That is, each feature f_j, that is a leaf node in the network, is labelled with the conditional probability $P(f_j \mid e_1 \cdots e_k)$, where $e_1 \cdots e_k$ are the exemplars that share the feature f_j. Similarly, the importance of an exemplar in the category is represented by probabilistic dependencies. Each exemplar e_i, which is an intermediate node in the network, is labelled with the conditional probability $P(e_i \mid JC)$, where JC is the joint category formed by the parents of e_i. This probability is the prior probability of the exemplar when no evidence is available. With this additional information, the network of Fig. 3.3 becomes a hybrid representation. Figure 3.4 shows this new mixed representation. The probabilistic representation, which is the lower network in Fig. 3.4, is a Bayesian network of the kind introduced in Chapter 2. The exemplar based representation, which is the upper network in Fig. 3.4, shows the exemplars that describe a category.

More formally, the representation can be summarised as follows.

- A domain has a set of categories $\{C_1, \cdots, C_n\}$.

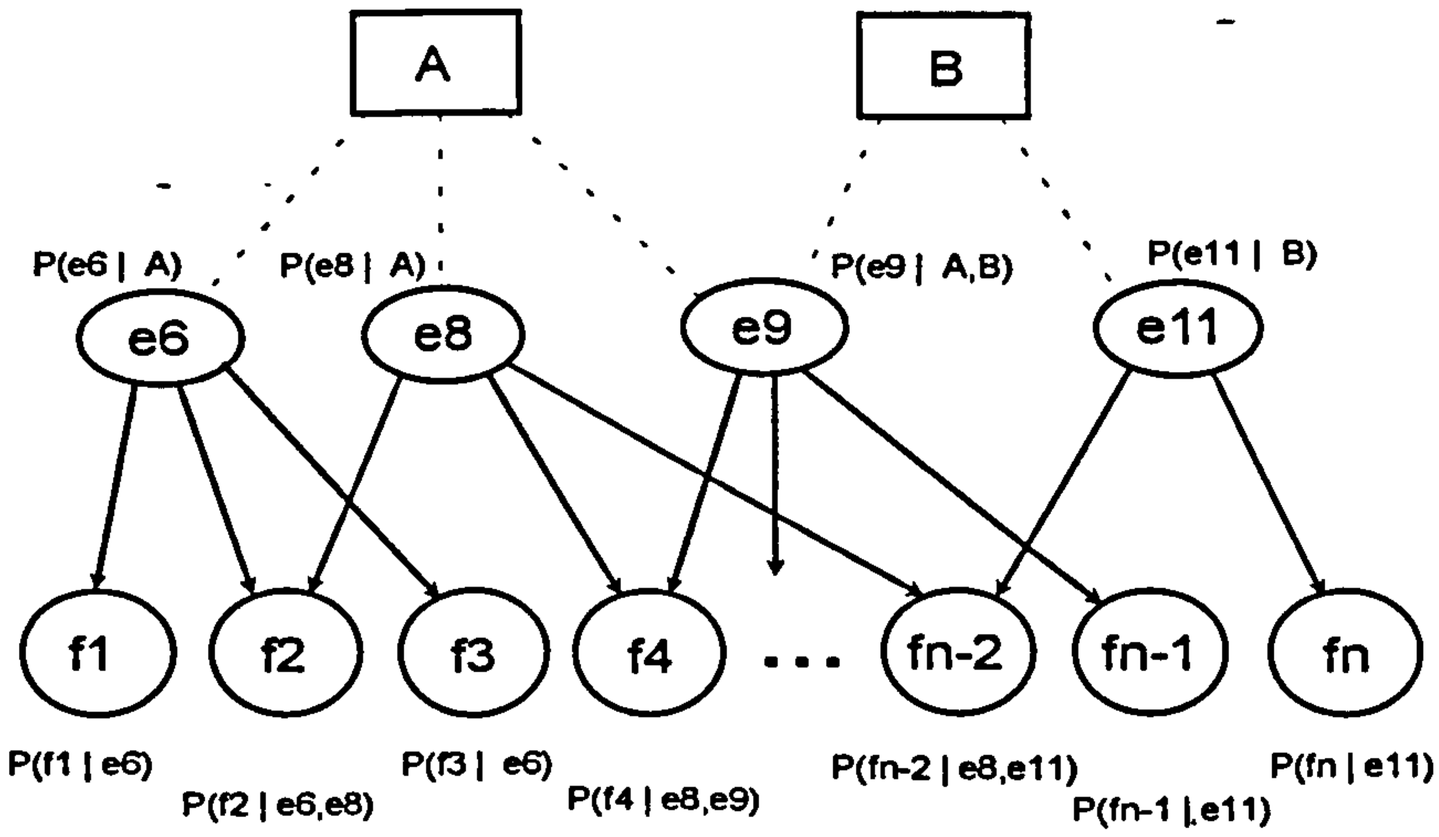


Figure 3.4: A probabilistic exemplar based representation.

- A category C_i is represented by a set of exemplars and their conditional probabilities $\{ep_{i1}, \cdots, ep_{im}\}$, where $ep_{ij} = (e_{ij}, P(e_{ij} \mid JC))$.

  - An exemplar e_{ij} is represented by a set of features and their conditional probabilities $(fp_{ij1}, \cdots, fp_{ijo})$, where $fp_{ijk} = (f_{ijk}, P(f_{ijk} \mid parents(f_{ijk})))$.

  - A case c_x is represented by its features $(f_1, \cdots, f_p)$.

For simplicity, a feature f_j is assumed to be a binary variable. However, if continuous variables occur, these can be discretised using a simple method such as dividing the range of values into a number of intervals required or a more sophisticated method as proposed in Dougherty et al. (1995).

3.3 The Classification Process

Given the above representation, how can the following questions, raised earlier, be addressed:

- How can a new case be classified?

- What notion of similarity can be adopted?

The majority of current CBR systems address these questions by adopting a similarity metric, which is a weighted sum of the differences between a new case and a stored case [Kolodner 1993, Aha 1991]. The main problem with this approach is that the weights of the similarity metric need to be estimated. Some applications of CBR have used expert judgement to estimate the weights. Obtaining the weights is a wide open research area. For example, some approaches adopt flexible weighting schemes [Aha & Goldstone 1992], statistics methods [Mohri & Tanaka 1994] and context sensitive feature selection algorithms [Aha & Bankert 1994, Domingos 1997]. Obtaining the relevant features in a context and the weights associated with these features are the goals that these approaches are trying to achieve. A survey of different approaches can be found in the published work of Wettschereck & Aha (1995) and more recently in Wettschereck et al. (1997).

In this thesis, the notion of similarity adopted is that two cases are similar if they are represented by the same exemplar. But how can one determine if a new case is represented by a particular exemplar? Since in the above representation, the lower network that relates exemplars and features is a Bayesian network, the degree to which a new case with features $f_{1nc}, \cdots, f_{qnc}$ is represented by an exemplar e can be computed by:

$$P(e \mid f_{1nc}, \cdots, f_{qnc}) \tag{3.1}$$

This computation can be carried out by using the propagation methods introduced in Chapter 2.

Given this capability of calculating the extent to which an exemplar represents a new case, all the exemplars could be investigated, in theory at least. However, probabilistic propagations methods can be computationally expensive (NP-hard

in general) [Cooper 1990] and investigating all the exemplars is therefore not practical.

Hence, first it is necessary to rank the categories in order of the likelihood of them containing a suitable exemplar. This ranking has to be performed in a way that avoids missing suitable exemplars but is computationally efficient. This ranking can be obtained by utilizing an observation by Smith & Medin (1981) who point out that:

> "the features that represent a concept are salient ones that have a substantial probability of occurring in instances of the concept".

Thus, the important features will have high values of occurrence given an exemplar, i.e., high values of $P(f_j \mid e)$. Hence, a reasonable way of ranking the categories is to obtain the contribution of the features of the exemplar that are present in the new case, averaged over the number of features in the exemplar e_i:

$$Rank(e_i) = \frac{\sum_{f \in e_i} P(f \mid e_i)}{nfe_i} \tag{3.2}$$

where

$$P(f \mid e_i) = 0 \quad when \ f \notin nc$$

In this equation, nc is a new case and nfe_i is the number of features in the exemplar e_i.

Then, the categories can be ranked in order of the rank of their exemplars. Once the ranking is obtained, a suitable investigation strategy can be adopted. For example, the list of categories can be investigated in order of rank until a good exemplar is found. In the context of this model, a good exemplar is one that has a value of $P(e \mid nc)$ above a threshold that is normally dependent on the application. Adopting this strategy, the classification process can be summarized as the algorithm in Fig. 3.5.

Classify(nc)

Input: A new case described by a set of features $nc = \{f_a, \cdots, f_g\}$
Results: the exemplar e_c that best classifies the new case and
the category list CL that classify e_c

The following local variables are used:
H is a list of categories
E, CE are lists of exemplars
C_c is the current category
done is a boolean variable

Step 1. Determine and rank the categories (hypotheses)

for all e_j do

$$Rank(e_i) = \frac{\sum_{f \in e_i} P(f|e_i)}{nf e_i}$$

where $P(f \mid e_i) = 0 \quad when \ f \notin nc$
end(for)
set CE to the list of candidate exemplars in descending order of rank
set H to the list of categories ranked in descending order of rank of its best exemplar

Step 2. Determination of an Exemplar

$e_c = nil$
$C_c = first(H)$ $first(H)$ returns $\emptyset$ when H is empty
$done = false$

while (not *done*) and ($C_c \neq \emptyset$) do begin

In the exemplar-features Bayesian network

for each $e_i \in C_c$ do
compute $P(e_i|nc)$
end(for)
set E to the list of exemplars in C_c ranked in descending order of $P(e_i|nc)$

$e_c = first(E)$
if $P(e_c \mid nc) > threshold$ then
$done = true$
else
$C_c = next(H)$
end(if)
end(while)

if done then
CL = all categories that contain (e_c)
else
$e_c = nil$
$CL = \emptyset$
end(if)

return (e_c, CL)

Figure 3.5: Classification algorithm.

3.4 The Learning Process

The last section described how to classify a new case given an existing exemplar based model and its representation. This section describes how the model and its representation are learned. There are two aspects of learning involved. First, the exemplar based model needs to be learned and second, the parameters of the model need to be estimated. Both aspects need to be done in an incremental fashion. Subsection 3.4.1 develops the algorithm for learning the exemplar based model and Subsection 3.4.2 describes how the parameters required by the exemplar based model are estimated.

3.4.1 Learning the model

The last section described how to classify a new case given an existing exemplar based model and its representation. This subsection describes how the model and its representation are learned. In particular, the following questions, that were raised earlier in Section 3.1 are addressed:

1. What makes a good exemplar?

2. How can the model be learned incrementally?

To answer these questions, consider the situation shown in Fig. 3.6 where there is a category C that is represented by three exemplars e_1, e_2 and e_3. Suppose a new training case with category C arrives, then there are two situations, shown in Fig. 3.7, that can arise:

(a) The new case is not classified by the exemplars in C.

(b) The new case is correctly classified by an exemplar in C.

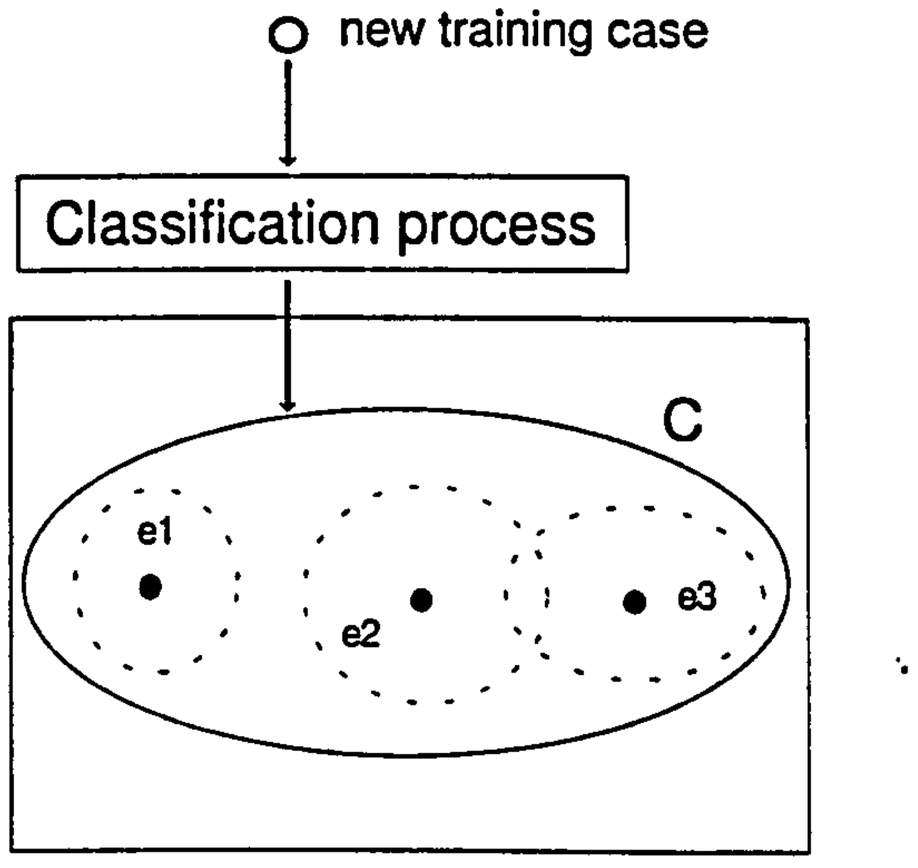


Figure 3.6: Classifying a new case in a category C.

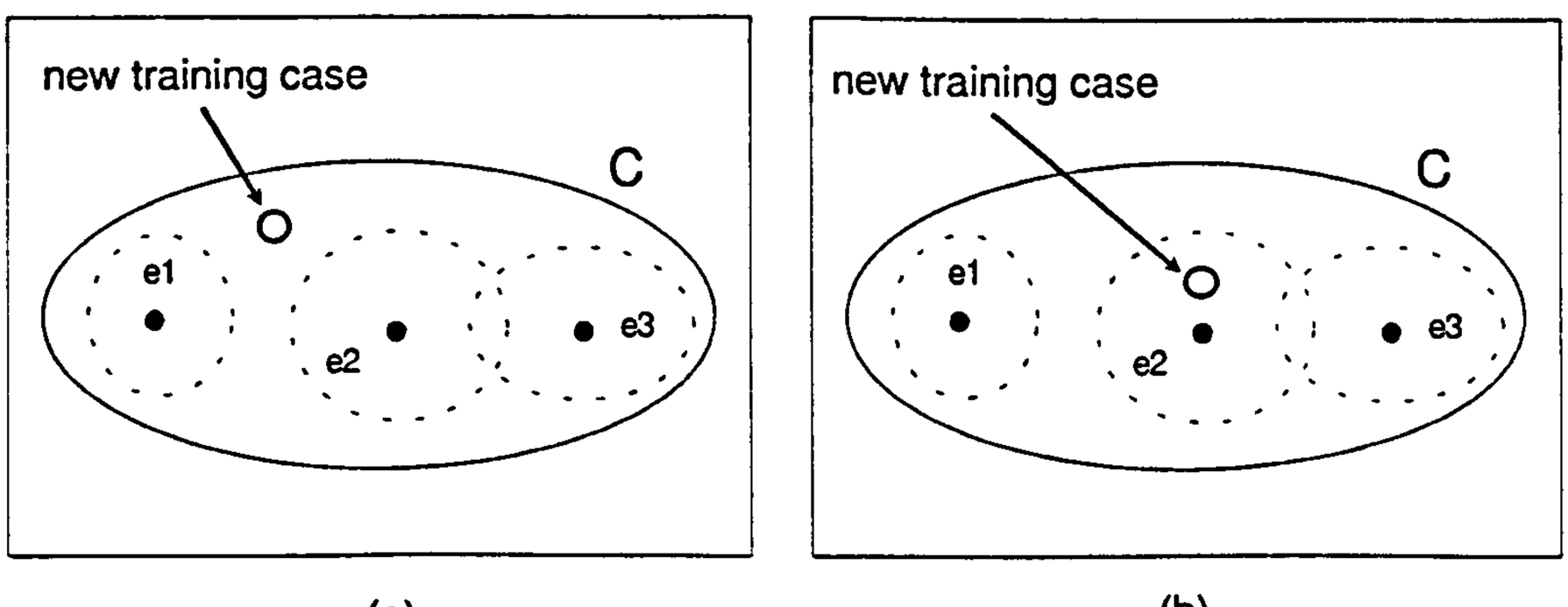


Figure 3.7: Situations in the classification process.

In the first case, clearly the new case shoud be retained as a new exemplar since it must be different from the other exemplars. In the second case, which is illustrated in Fig. 3.7(b), criteria need to be developed for deciding which of the two, the new case or exemplar, will be the best representative of all cases in the region.

For exemplar based models these criteria have to be based on the notion of *prototypicality*. Before describing the measure of prototypicality used in this thesis, it is necessary to first describe the idea of a summary representation. In Section 3.2 an exemplar was represented as a Bayesian network with dependencies from the exemplar to its features. In general, an exemplar may not have the same features as all the similar cases that it represents. For example, in Fig 3.8, the exemplar e_2 may have the features f_4, f_6, and f_9 while the union of all the features of the cases it represents may be f_3, f_4, f_6, f_7, and f_9. In general, a *summary representation* is a Bayesian network where all the features of the similar cases are included. Figure 3.8 shows the summary representation of the exemplar e_2.

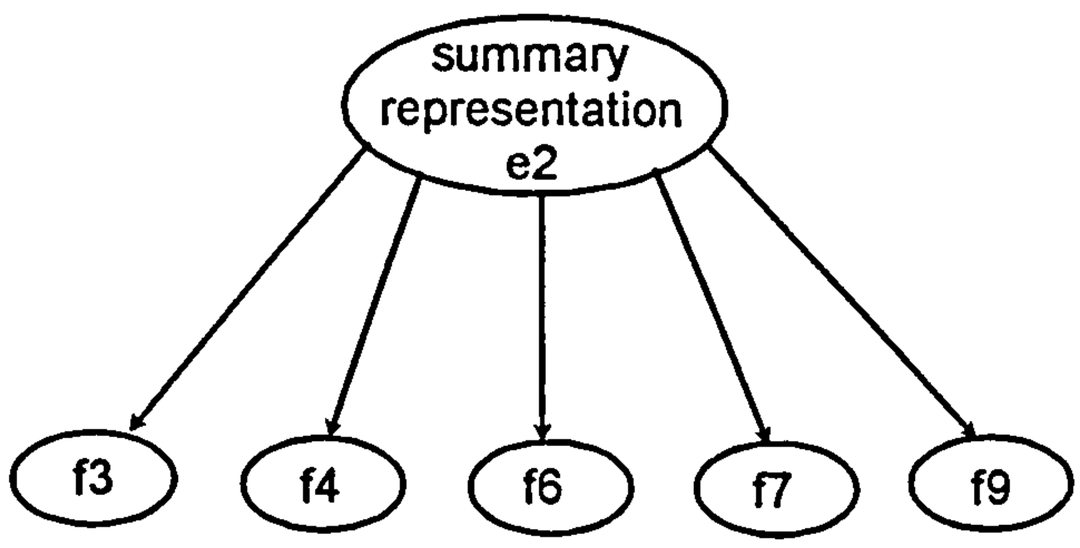


Figure 3.8: A summary representation of the exemplar e_2.

Returning to the notion of prototypicality, the problem can be summarised as shown in Fig 3.9. In this figure there are several exemplars, each with a summary representation, as shown on the right of the figure. The basic problem is to develop a measure of prototypicality so that the best prototype can be selected.

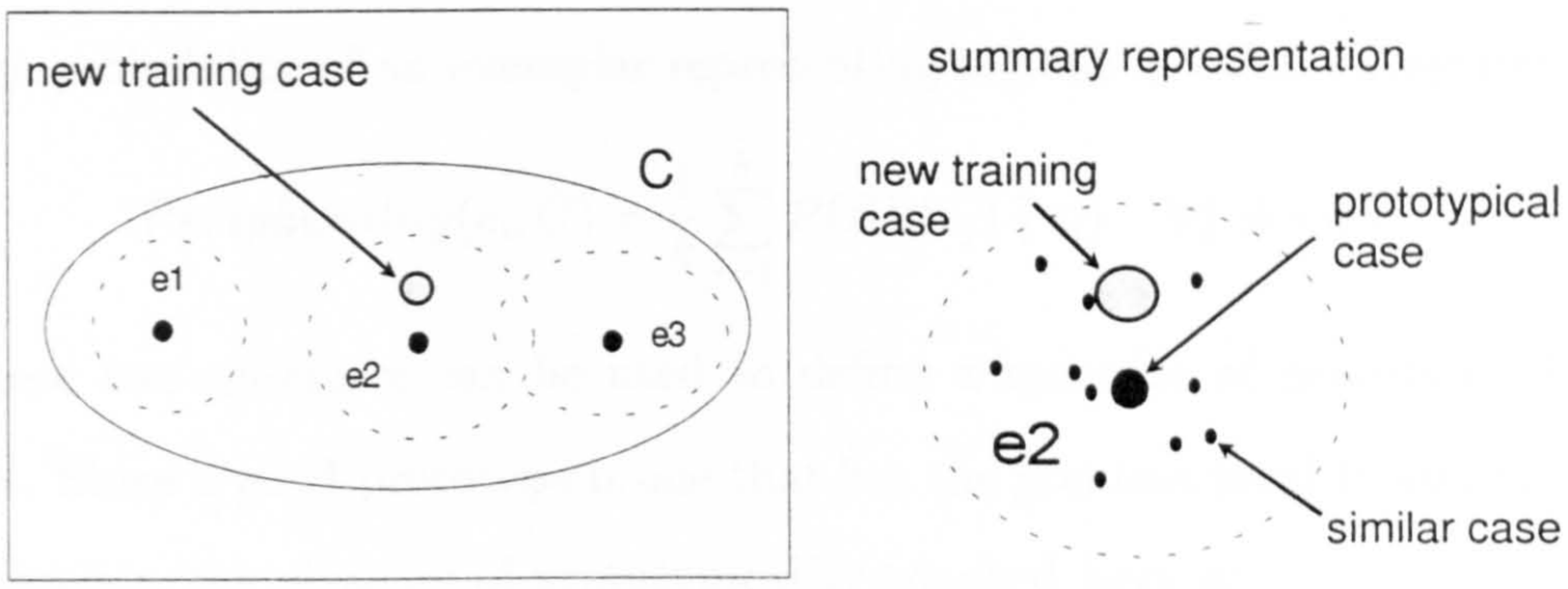


Figure 3.9: A summary representation of an exemplar.

Rosch & Mervis (1975) argued that a case is an ideal prototype if (i.e. it may not exist):

- it has the highest family resemblance with other members in the same category, (this is known as *focality* [Biberman 1995]) and
- it has the least family resemblance with members of other categories (this is known as *peripherality* [Biberman 1995]).

In the contex of the model being developed here, *family resemblance* is viewed as the collection of similar cases and which have a summary representation. In terms of regions, a case that maximizes the probability of covering a region can be considered to have the highest family resemblance. Since the summary representation denotes regions, and takes the form of a Bayesian network, a suitable measure of focality of an exemplar e_i is the probability of covering a region:

$$Focality(e_i) = P(SR(e_i) \mid e_i) \tag{3.3}$$

where $SR(e_i)$ denotes the summary representation of the region that contains e_i.

Likewise, a suitable measure of peripherality is obtained by working out the average probability of an exemplar representing regions in other categories:

$$Peripherality(e_i, C) = \frac{1}{k}\sum_{j=1}^{k} P(SR(e_j) \mid e_i) \quad \forall j \neq i \in C \tag{3.4}$$

These two measures can be used to define a measure of prototypicality as follows. Since a good prototype is one that has the greatest focality and the least peripherality, the measure of prototypicality adopted here is:

$$Prototypicality(e_i, C) = Focality(e_i) - Peripherality(e_i, C) \tag{3.5}$$

This measure of prototypicality can now be used to decide which case makes the better exemplar in a region.

The above considerations lead to the following learning algorithm shown in the Fig. 3.10.

Input: A training case described by a set of features $nc = \{f_a, \cdots, f_i\}$ and a set of categories $L = \{C_1, \cdots, C_p\}$ to which it *jointly* belongs.
Results: Updated exemplar base model.

1. $Classify(nc)$ (as given in Fig. 3.5)

Classification outcomes are stored in the following local variables:
CL is a list of categories that classify the nc
e_c is the exemplar that best classifies the nc

2. if $(CL = \emptyset)$ then
$e_c = nc$
add_exemplar(e_c, C_i) for each $C_i \in L$
return
end(if)

3. if $L = CL$ then
In the joint category $JC \in L$ do
$pe_c = prototypicality(e_c, JC)$
$pnc = prototypicality(nc, JC)$
if $(pnc > pe_c)$ then
nc replaces e_c in the definition of JC
end(if)
else
$e_c = nc$
add_exemplar(e_c, C_i) for each $C_i \in L$
end(if)

Figure 3.10: Learning algorithm.

3.4.2 Learning the probabilities

The last subsection concluded with the algorithm for learning an exemplar based model. However to use it, the probabilities that define the Bayesian network which represents the exemplars are required. Since the model is incremental, and the cases are not retained, estimating the probabilities in a manner that enables a good exemplar based model to be learned is a non-trivial problem.

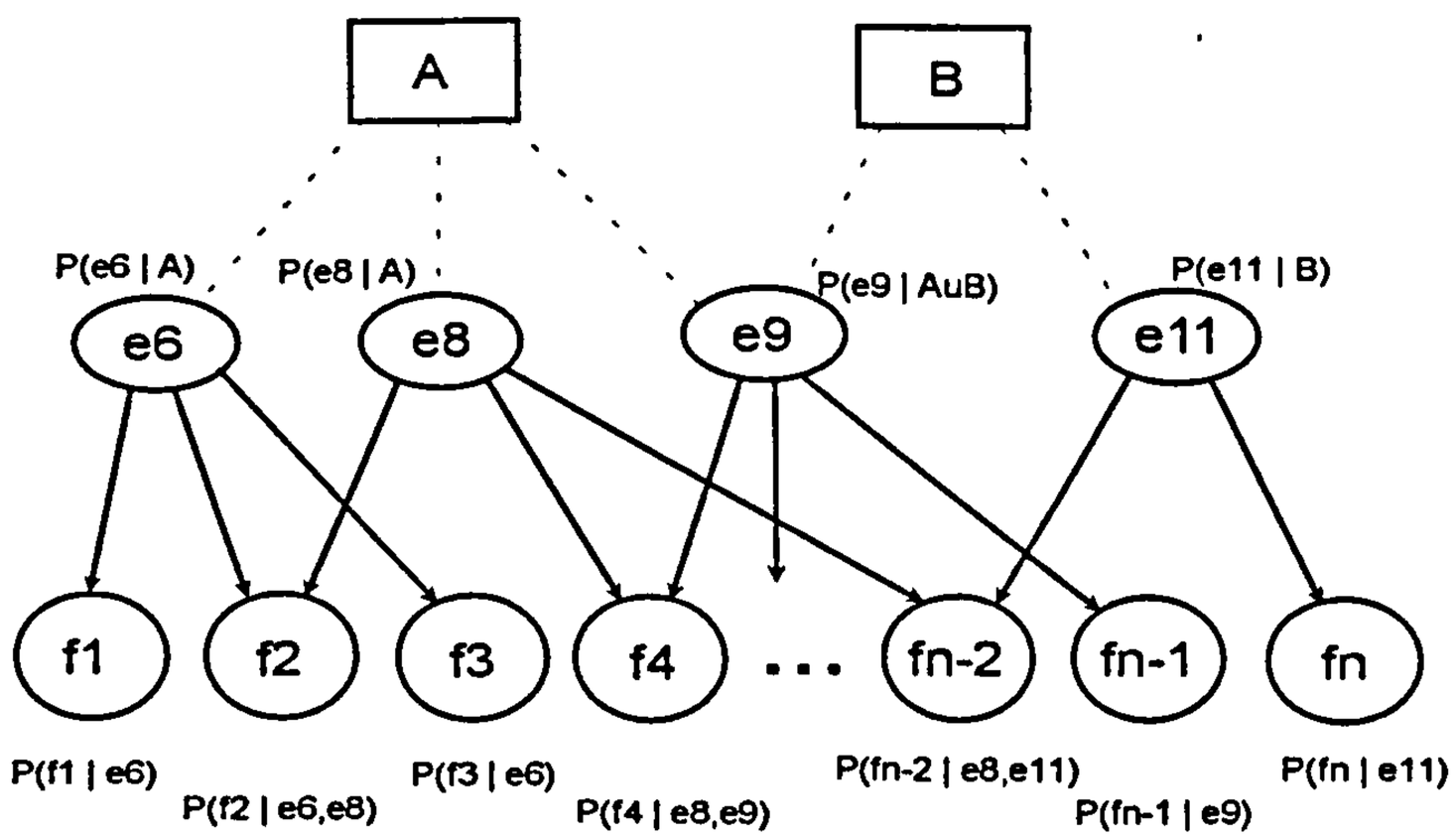


Figure 3.11: A probabilistic exemplar based model.

Figure 3.11 shows a Bayesian network that represents the state of the exemplar based model after some cases have been seen. The parameters that need to evolve as new cases are seen include:

1. prior probabilities of the exemplars in the joint category $P(e \mid JC)$ and
2. the conditional probability $P(f \mid parents(f))$.

If there are many cases available in advance, the prior probabilities can be easily estimated. However, since the model is incremental, the prior probabilities

need to start from a position of ignorance and then improve in accuracy as new cases arrive. The following describes how each of these probabilities is estimated.

Computing the prior probabilties

Estimating prior probabilities from data is a problem that has been addressed by statisticians. The most common and widely used method is to utilise a beta distribution [Lindgren 1976]. This distribution takes the following form, where x represents a possible value of the prior probability.

$$\beta(a,b) = \frac{(a+b+1)!}{a!b!}x^a(1-x)^b \quad where\ a,b > 0. \tag{3.6}$$

The expected value of x is the estimated value of the prior probability. It can be shown that if an event E occurs k times out of n then [Neapolitan 1990]:

$$Prior(E) = \frac{k+a+1}{n+a+b+2} \tag{3.7}$$

The values of a and b determine the form of the distribution and reflect the confidence in the average being the prior value. In the context of this model, a uniform distribution must be assumed since no information is available about the distribution of the data. This uniform distribution, which reflects ignorance, is obtained by setting $a = b = 0$.

Thus, given a category, the following equation can be used to compute and update the prior probabilities:

$$P(e \mid C) = \frac{number\ of\ cases\ in\ e + 1}{number\ of\ cases\ in\ C + 2} \tag{3.8}$$

Notice, when there are no cases, this returns a value of 0.5, which represents ignorance.

Computing the conditional probabilities

Estimating the conditional probabilities $P(f|parents(f))$ is much more difficult. To illustrate the difficulty, suppose a feature f has the exemplars e_1 and e_2 as

parents. Then, Table 3.1 shows the conditional probabilities that need to be computed.

Table 3.1: Conditional probabilities of f given e_1, e_2.

$P(f \mid e1, e2)$,
$P(\neg f \mid e1, e2)$,
$P(f \mid e1, \neg e2)$,
$P(\neg f \mid e1, \neg e2)$,
$P(f \mid \neg e1, e2)$,
$P(\neg f \mid \neg e1, e2)$,
$P(f \mid \neg e1, \neg e2)$,
$P(\neg f \mid \neg e1, \neg e2)$

In general, 2^{n+1} probabilities need to be estimated for n parents. In particular, there may not be enough cases in the intersection of the parent events, even if there are enough cases in the regions represented by the parents. This means that estimates of probabilities such as $P(f \mid \neg e1, e2)$ could only be based on a small number of cases and would therefore be inaccurate even when many cases have been seen.

To overcome this problem, the noisy or model [Peng & Reggia 1994] described in Chapter 2 is considered. If this model can be adopted, then instead of requiring $P(f|parents(f))$ only $P(f \mid e_i)$ is needed. To see if the noisy or model can be used, consider the assumptions that it makes [Pearl 1988]:

Accountability An event m_j is false, $P(m_j) = 0$, if all conditions listed as causes of m_j are false.

Exception independence If an event m_j is a consequence of two conditions d_1 and d_2, then the inhibition of the occurrence of m_j under d_1 is independent of the mechanisms of inhibition of m_j under d_2.

In the context of this model, the exception independence assumption can be interpreted as requiring that the absence of the feature given one exemplar is

independent of the absence of the feature given another exemplar. The extent to which this assumption holds depends on the way the exemplars are selected. In Section 3.4, the selection scheme uses a measure of prototypicality that aims to reduce the possibility of selecting exemplars that represents similar regions. That is the selection scheme used minimizes the posibility of the exception independence assumption being broken.

The accountability assumption requires that if a case is not represented by the parent exemplars of a feature, then that feature does not occur in the case. Although this may hold when an accurate exemplar based model has been learned, it clearly does not hold while it is still learning. To overcome this problem, an additional virtual exemplar is added in the representation of each category. This additional exemplar can be viewed as representing all the cases that have not yet been seen. With this additional exemplar, the revised model is illustrated in Fig. 3.12. As the figure shows, this introduces dependencies between the virtual exemplar and the features. But how can the strengths of the dependencies be estimated, since the virtual exemplar represents unseen cases?

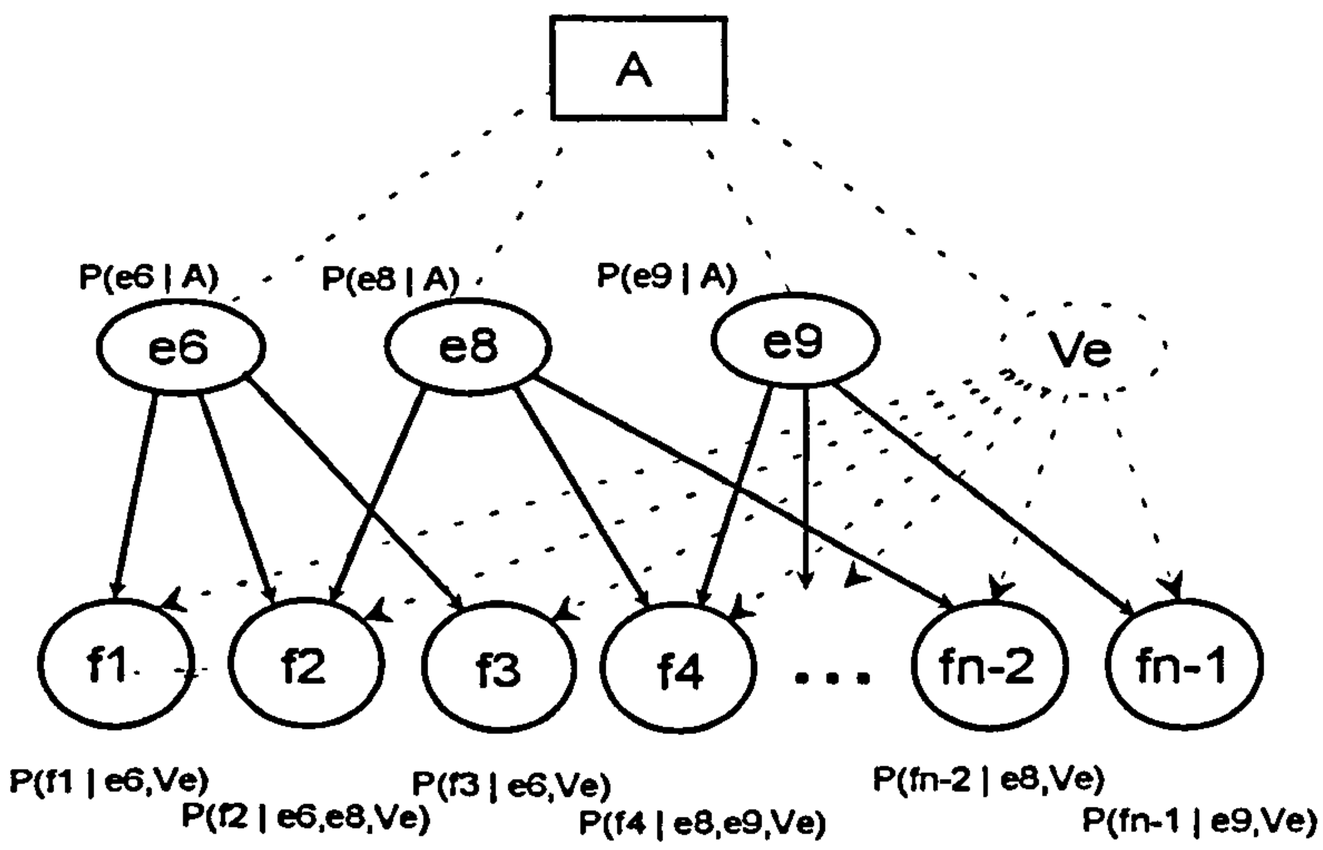


Figure 3.12: Virtual exemplar.

Estimating the strengths of these dependencies is therefore a task that requires predicting the behaviour of the dependencies as more cases are observed. This behaviour can be expected to have the following characteristics:

- The strengths of the dependencies should be the highest initially when no cases have been seen and ignorance is greatest.
- As more cases are observed, the strengths of the dependencies can be expected to decay since the virtual exemplar will represent fewer unseen cases.
- There is always a small chance that a new case will be in the region represented by the virtual exemplar even after many cases have been observed.

There may be several functions that satisfy these characteristics. However, a common function that is often used to represent decay is the exponential function. For example, it is used in modelling radio active decay and maintenance modelling [Chatfield 1978]. Hence, the exponential function is adopted and takes the form:

$$P(f \mid Ve) = \lambda e^{-\lambda * \alpha * n}$$

$$or$$

$$0.1 \ if \ P(f \mid Ve) < 0.1$$

where n is the number of cases in a category and α is a scaling parameter that determines the rate of decay. The lower bound of 0.1 in this function reflects that a new case will be in the region represented by the virtual exemplar even after many cases have been seen. The parameter α can be obtained by deciding the minimum value of the probability (last characteristic above) and deciding the number of cases for which the probability should be a minimum. Then, the above equation can be rearranged to obtain α as follows.

$$\alpha = -\frac{1}{\lambda * n} \ln \frac{P(f \mid Ve)}{\lambda} \tag{3.9}$$

This completes the description of how the probabilities can be learned incrementally, thereby allowing the use of the classification procedure of Fig. 3.5 and the learning procedure given in Fig. 3.10.

The next section gives an example that illustrates the whole process.

3.5 An Example

This section presents an example to illustrate the classification and learning processes. First, a training case is presented to show the stages of the learning process and second, a test case is presented to illustrate the classification process.

Suppose that the probabilistic exemplar based model is required for learning whether a person in a university is a teacher or a student. These two categories are not necessarily disjoint. For instance, a member of staff may be studying for a higher degree and would therefore be a teacher and a student.

Suppose 16 training cases have been observed, and a threshold of 0.6 is used.[1] This results in three exemplars for the TEACHER category as shown in Table 3.2 and two exemplars for the STUDENT category as shown in Table 3.3.

In these tables, the numbers in the exemplars indicate the actual cases that have been classified and are represented by the exemplar and the numbers in the features indicate the frequency of the feature in the exemplar. So, for example, W. Philips is known to represent 6 actual cases and the feature (age old) occurs five times. Notice that the exemplar A.Smith is in both categories. This means that A.Smith is both a TEACHER and a STUDENT.

Figure 3.13 shows the information in a more convenient format.

[1]Given that the model normally retains the early cases as exemplars, a low threshold is needed in order to obtain a small exemplar based model suitable for illustrative purposes.

Table 3.2: Exemplars in the category: TEACHER.

Exemplar: W. Philips (6)	Exemplar: L. Pintos (2)	Exemplar: A. Smith (3)
(age old) (5)	(age adult) (2)	(age old) (2)
(attention sleeping) (6)	(dressing formal) (1)	(dressing formal) (3)
(money much) (3)	(money few) (2)	(attention middle) (3)
(study very-much) (3)	(attention total) (2)	(money few) (1)
		(study very-much) (3)

Table 3.3: Exemplars in the category: STUDENT.

Exemplar: L. Garcia (5)	Exemplar: A. Smith (3)
(age adult) (5)	(age old) (2)
(dressing informal) (4)	(dressing formal) (3)
(attention middle) (4)	(attention middle) (3)
(money few) (2)	(money few) (1)
(study few) (1)	(study very-much) (3)

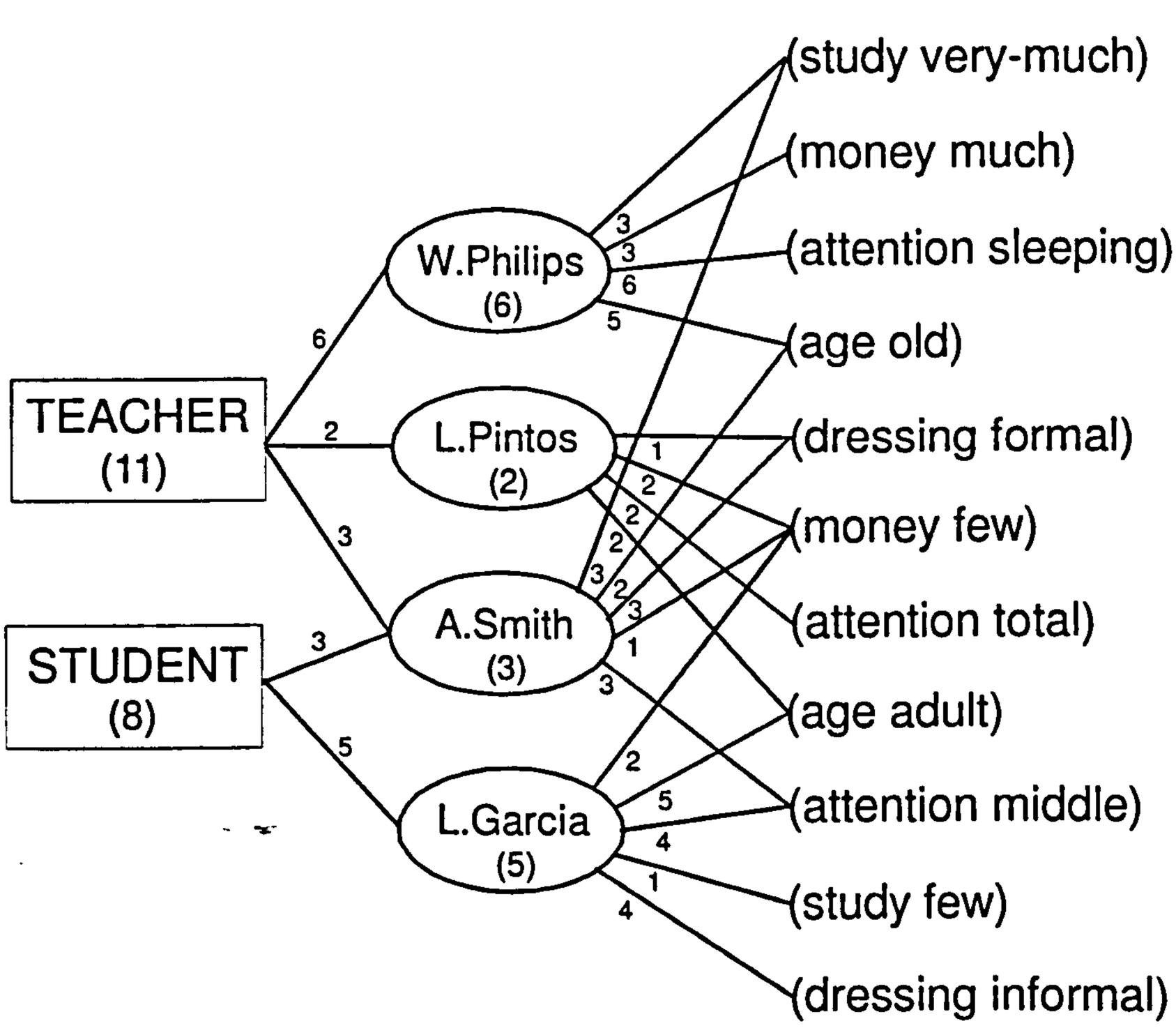


Figure 3.13: Exemplars model after sixteen training cases.

Now, suppose that the following new training case is given.

new training case : C.Pinan

categories: {TEACHER}

features : f_{1nc} : (age adult)

f_{2nc} : (attention total)

f_{3nc} : (money few)

f_{4nc} : (dressing formal)

f_{5nc} : (study very-much)

Learning process

The proposed model learns from the new training case in two stages. In the first stage, it determines which exemplars best classify the new training case. In the second stage, two actions can be performed: (i) if the new training case was not classified then, the new training case will be a new exemplar in the category or the joint category that it represents and (ii) if the new case was classified by an exemplar then, the new training case will compete with the exemplar that classified it, in order to determine the best exemplar that will represent the subset of similar cases in the category.

First stage: classification

In the first stage, the probabilistic exemplar based model builds a Bayesian network [Heckerman 1995] as shown in Fig. 3.14. The structure of this Bayesian network has two levels. The nodes in the lower level are the features (evidences) of the exemplars. The nodes in the top level are the exemplars (hypotheses) in the category.

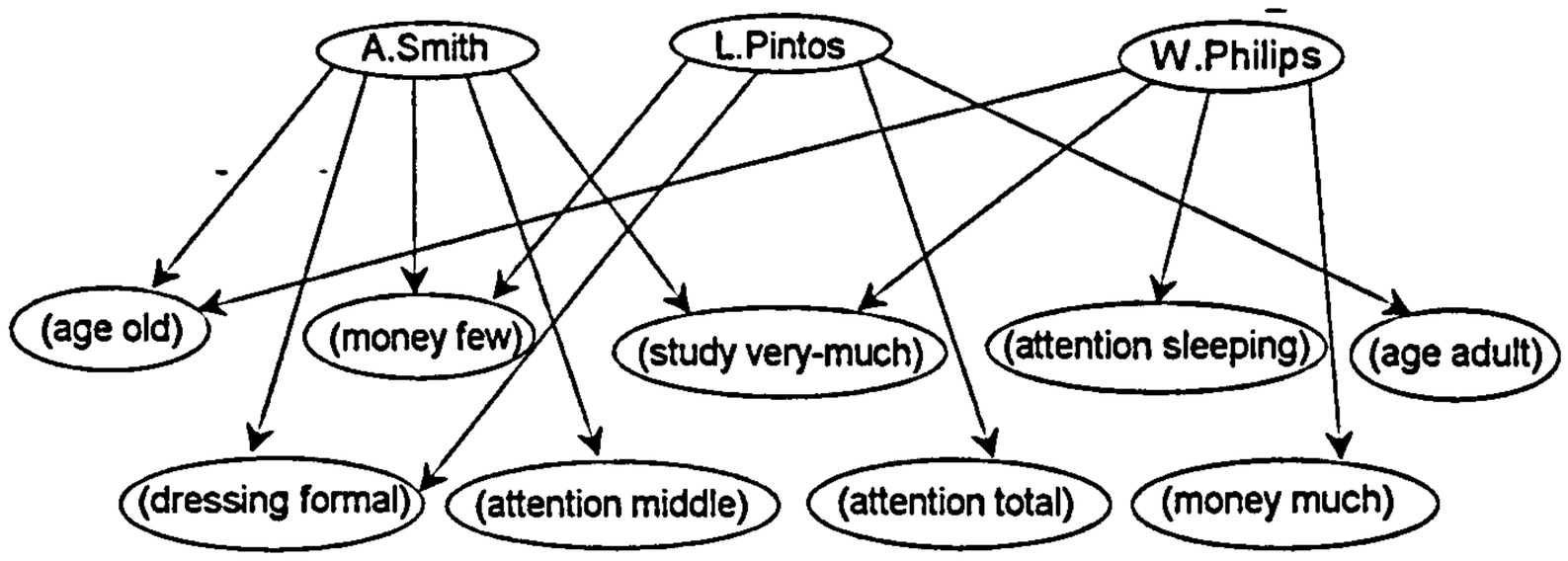


Figure 3.14: Bayesian network to classify a new training case.

The prior and conditional probabilities required in the Bayesian network are based on the number of cases that have been used to train the model. The prior probabilities $P(e_i \mid C_j)$ are computed by using Equation 3.8.

In the above example, the prior probabilities of the exemplars, that represent the category TEACHER: W.Philips, L.Pintos, and A.Smith, are 7/13, 3/13, and 4/13 respectively.

Since the noisy or model is adopted, an estimate of the conditional probabilities of the features given their parent exemplars are also required.

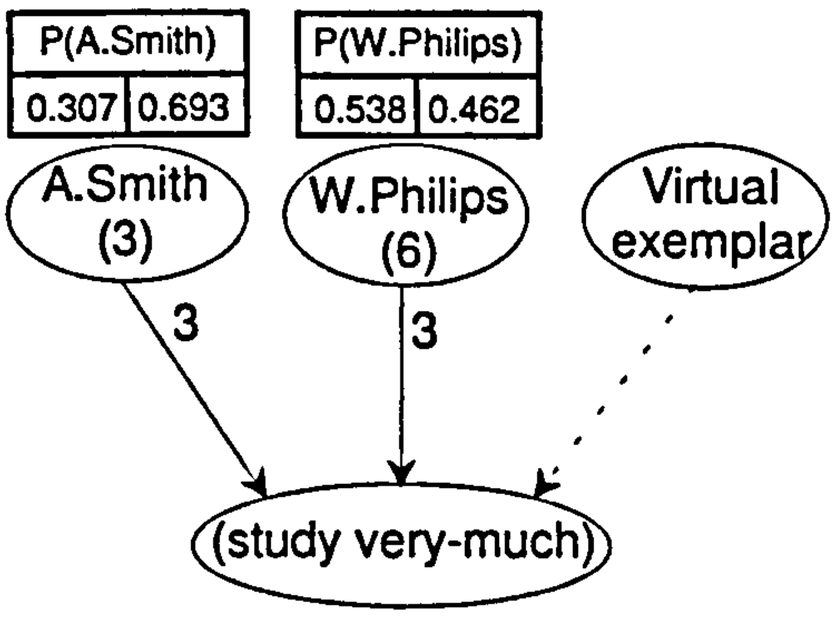
In general, these conditional probabilities are computed using the following equation.

$$P(\neg f \mid parents(f)) = \prod_{e_k = true \wedge e_k \in parents(f)} (1 - P(f \mid e_k)) \qquad (3.10)$$

The whole matrix of the conditional probabilities of the feature (study very-much) is computed as follows. Suppose, for conciseness, the feature (study very-much) is represented by f, the virtual exemplar is denoted by VE, and the exemplars E.Smith and W.Philips are represented by e_1 and e_2 respectively, then:

$$P(\neg f \mid e_1, e_2, VE) = (1 - P(f \mid e_1))(1 - P(f \mid e_2))(1 - P(f \mid Virtual\_exemplar))$$

where $(1 - P(f \mid e_i))$ is the conditional probability of the feature in the exemplar



P(study very-much | A.Smith, W.Philips)		
	(study very-much)	¬(study very-much)
A.Smith, W.Philips	0.9247	0.0753
A.Smith, ¬W.Philips	0.8498	0.1502
¬A.Smith, W.Philips	0.6237	0.3763
¬A.Smith, ¬W.philips	0.2475	0.7525

Figure 3.15: Part of the Bayesian network for the feature (study very-much).

i and $P(f \mid virtual\_exemplar)$ is the conditional probability of the feature in the virtual exemplar that represents the cases that have not been seen in the category. In order to compute the conditional probability of a feature given the virtual exemplar, Equation 3.9 is used. Thus, the negative part of the conditional probability is computed as follows.

$$P(\neg f \mid +e_1, +e_2, VE) = (1 - P(f \mid e_1))(1 - P(f \mid e_2))(1 - \lambda e^{-\lambda * \alpha * n})$$
$$P(\neg f \mid +e_1, \neg e_2, VE) = (1 - P(f \mid e_1))(1 - \lambda e^{-\lambda * \alpha * n})$$
$$P(\neg f \mid \neg e_1, +e_2, VE) = (1 - P(f \mid e_2))(1 - \lambda e^{-\lambda * \alpha * n})$$
$$P(\neg f \mid \neg e_1, \neg e_2, VE) = (1 - \lambda e^{-\lambda * \alpha * n})$$

where $+e_i$ denotes the presence of e_i.

Suppose, for illustrative purposes, the parameters λ and α are set to 0.4 and 0.1 respectively.The values of $P(f \mid e_i)$ are obtained by using a variation of

Equation 3.8. Thus, since e_1 represents 3 cases all of which have the feature f, an estimate of $\frac{4}{5}$ is obtained for $P(f \mid e_1)$. Hence, the conditional probabilities are:

$$P(\neg f \mid +e_1, +e_2, +VE) = (1 - \tfrac{4}{5})(1 - \tfrac{4}{8})(1 - 0.4 * e^{-0.4*0.1*12}) = 0.08$$
$$P(\neg f \mid +e_1, \neg e_2, +VE) = (1 - \tfrac{4}{5})(1 - 0.4 * e^{-0.4*0.1*12}) = 0.15$$
$$P(\neg f \mid \neg e_1, +e_2, +VE) = (1 - \tfrac{4}{8})(1 - 0.4 * e^{-0.4*0.1*12}) = 0.38$$
$$P(\neg f \mid \neg e_1, \neg e_2, +VE) = (1 - 0.4 * e^{-0.4*0.1*12}) = 0.75$$

Notice that since the virtual exemplar is only needed for computing the effect of ignorance on the conditional probabilities of the features, only those situations where the virtual exemplar is present need to be considered.

The whole matrix of conditional probabilities for the feature (study very-much) is shown in Table 3.4.

Table 3.4: Conditional probability of feature (study very-much).

	f	$\neg f$
$+e_1, +e_2, +VE$	0.92	0.08
$+e_1, \neg e_2, +VE$	0.85	0.15
$\neg e_1, +e_2, +VE$	0.62	0.38
$\neg e_1, \neg e_2, +VE$	0.25	0.75

In order to establish whether evidence is present (i.e. positive) or not (i.e. negative), the features of the exemplars are matched with the features of the new training case. If a feature of an exemplar matches then, the evidence is positive, otherwise it is negative. In the example of this section, Fig. 3.14, the positive features are: (dressing formal), (money few), (study very-much), (attention total), and (age adult), while the negative features are: (age old), (attention middle), (attention sleeping), and (money much).

Once the prior probabilities of all the exemplars, the condititional probabilities of all the features, and the positive and negative features are known then, the

posterior probabilities of the exemplars given the evidence can be computed using the propagation method described in Chapter 2. In this example, this results in the following posterior probabilities .

$$\begin{aligned} \text{P(L.Pintos} \mid f_{1nc}, f_{2nc}, f_{3nc}, f_{4nc}, f_{5nc}) &= 0.94 \\ \text{P(A.Smith} \mid f_{1nc}, f_{2nc}, f_{3nc}, f_{4nc}, f_{5nc}) &= 0.17 \\ \text{P(W.Philips} \mid f_{1nc}, f_{2nc}, f_{3nc}, f_{4nc}, f_{5nc}) &= 0.04 \end{aligned}$$

Hence, the exemplar L.Pintos classifies the new training case.

Second stage: learning

As the new training case was classified by an existing exemplar then, in the second stage, the goal is to determine whether the new case is a better exemplar. This is done by computing the prototypicality measure for both the new case, and the exemplar that classified it:

$$\begin{aligned} &\text{Prototypicality(L.Pintos, TEACHER)} = \\ &\qquad \text{Focality(L.Pintos)} - \text{Peripherality(L.Pintos, TEACHER)} \\ &\text{Prototypicality(C.Pinan, TEACHER)} = \\ &\qquad \text{Focality(C.Pinan)} - \text{Peripherality(C.Pinan, TEACHER)} \end{aligned}$$

where focality and peripherality are defined by:

$$Focality(e_i) = P(SR(e_i) \mid e_i)$$

$$Peripherality(e_i, C) = \frac{1}{k} \sum_{j=1}^{k} P(SR(e_j) \mid e_i) \quad \forall j \neq i \in C$$

The conditional probabilities $P(SR(e_j) \mid e_i)$ are obtained by propagating probabilities in the Bayesian network that consists of the summary representations of all the exemplars in the category. So for example, Fig. 3.16 shows

the Bayesian network for the TEACHER category that includes the summary representations of the exemplars in that category. Notice that the summary representation of A.Smith includes a feature (has computer), which is not a feature of A.Smith but a feature of a case represented by the exemplar A.Smith.

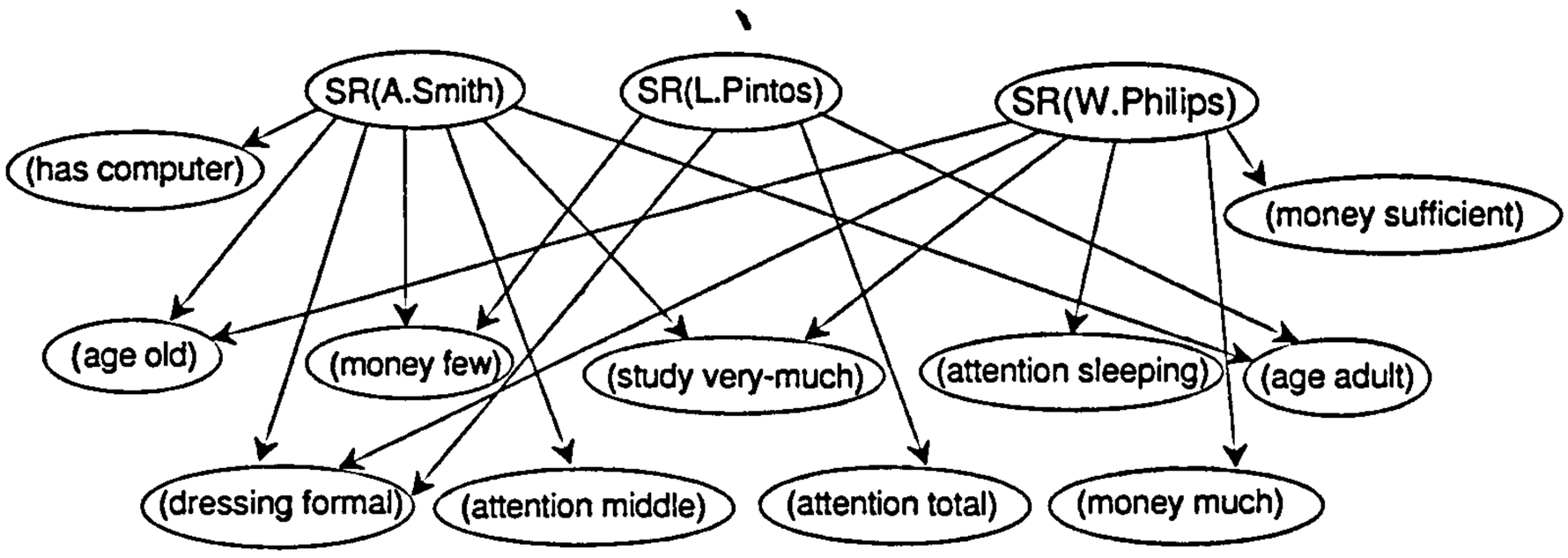


Figure 3.16: Bayesian network of the summay representation in TEACHER category.

Once the propagation has been done, the following prototypicality values are obtained:

Prototypicality(L.Pintos, TEACHER) = 0.95 - 0.01 = 0.94

Prototypicality(C.Pinan, TEACHER) = 0.99 - 0.04 = 0.95

As can be seen, the new exemplar, C.Pinan, has a prototypicality higher than the exemplar L.Pintos. Thus, C.Pinan will be the new exemplar that represents the subset of similar cases in the TEACHER category. Figure 3.17 shows the updated organisation structure after the exemplar C.Pinan is selected.

Classification process

Now, suppose that the model was trained with the seventeen previous cases and the following new test case is given.

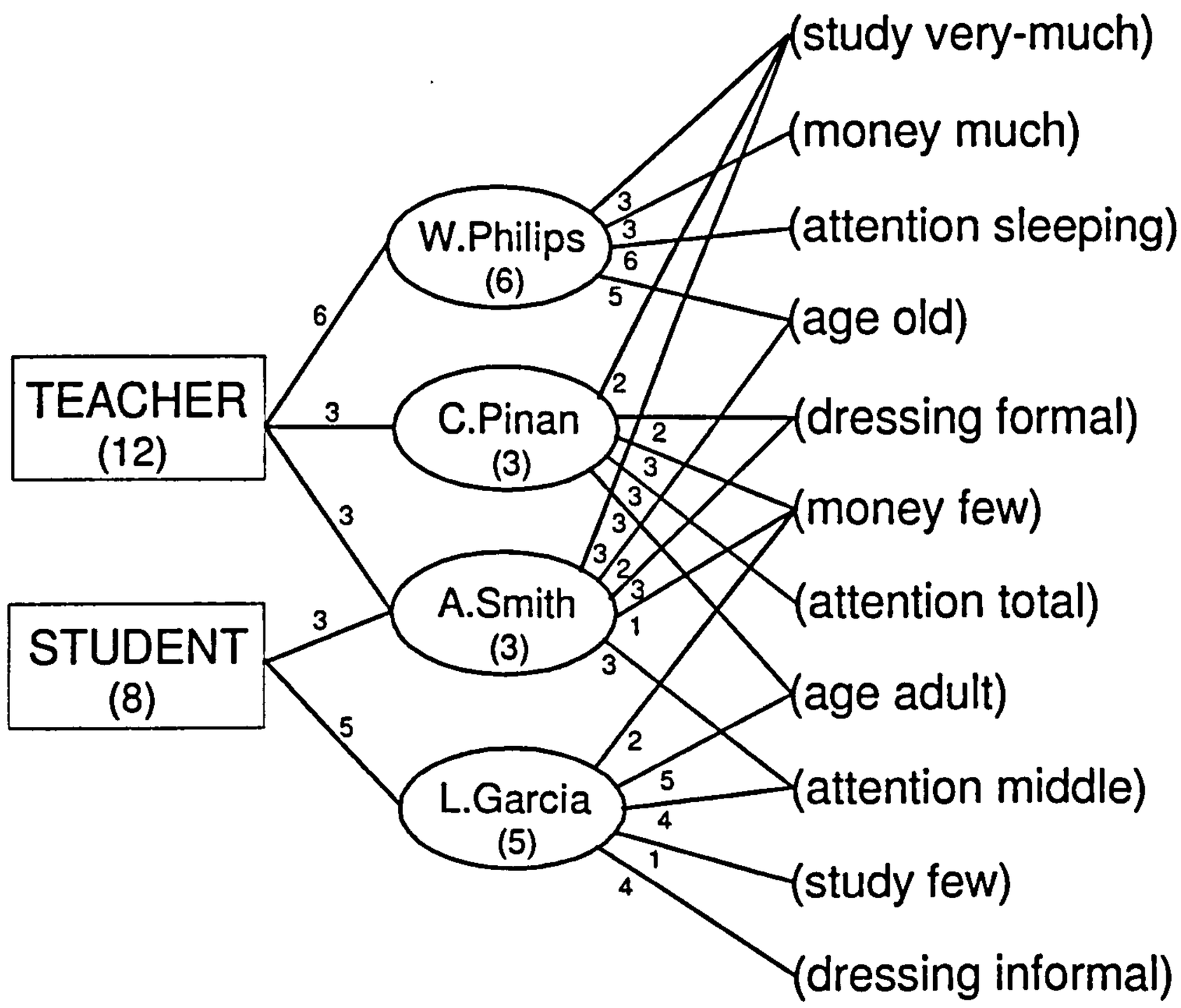


Figure 3.17: Updated organisation structure.

new case :J.Perez

features : - (age old)
(dressing formal)
(money few)
(study very-much)

The probabilistic exemplar based model also classifies a test case in two stages. In the first one, it determines the hypotheses, which are the categories that potentially contain suitable exemplars. In the second stage, it computes the posterior probabilities of the exemplars in each category given the new case. The categories are ranked and investigated according to the most promising exemplars.

First stage: hypotheses definition

First, each exemplar is ranked using the equation:

$$Rank(e_i) = \frac{\sum_{f \in e_i} P(f \mid e_i)}{nfe_i}$$

where

$$P(f \mid e_i) = 0 \quad when \ f \notin nc$$

where the conditional probability is computed using:

$$P(f \mid e_i) = \frac{frequency\ of\ f\ in\ e_i + 1}{cases\ represented\ by\ e_i + 2}$$

Table 3.5 gives the conditional probabilities obtained for this example, and Table 3.6 presents the ranks of the exemplars.

Now, from Table 3.6, the rank of the categories can be determined as the rank of their highest ranked exemplar. So for example, the weight of the categories

Table 3.5: Conditional probabilities of all selected exemplars.

feature f	selected exemplar e_i	frequency	represented cases for e_i	$P(f \mid e_i)$
age(old)	A.Smith	2	3	0.60
	W.Philips	5	6	0.75
dressing(formal)	A.Smith	3	3	0.80
	C.Pinan	2	3	0.60
money(few)	L.Garcia	2	5	0.43
	A.Smith	1	3	0.40
	C.Pinan	3	3	0.80
study(very-much)	A.Smith	3	3	0.80
	C.Pinan	2	3	0.60
	W.Philips	3	6	0.50

Table 3.6: Ranking of selected exemplars.

exemplar e_i	total weight	features in e_i	rank(e_i)	categories
A.Smith	2.60	5	0.52	TEACHER STUDENT
C.Pinan	2.00	5	0.40	TEACHER
W.Philips	1.25	4	0.31	TEACHER
L.Garcia	0.43	5	0.09	STUDENT

TEACHER and STUDENT is 0.52 since A.Smith is the highest ranked exemplar in those categories.

Once the hypotheses are established, the second stage of the classification process is performed.

Second stage:hypothesis confirmation

In this example both, the TEACHER and STUDENT categories are ranked the same. Suppose the TEACHER category is investigated first. The Bayesian network for the TEACHER category is shown in Fig. 3.18. This network is used to evaluate $P(e_i \mid J.Perez)$ by probabilistic propagation.

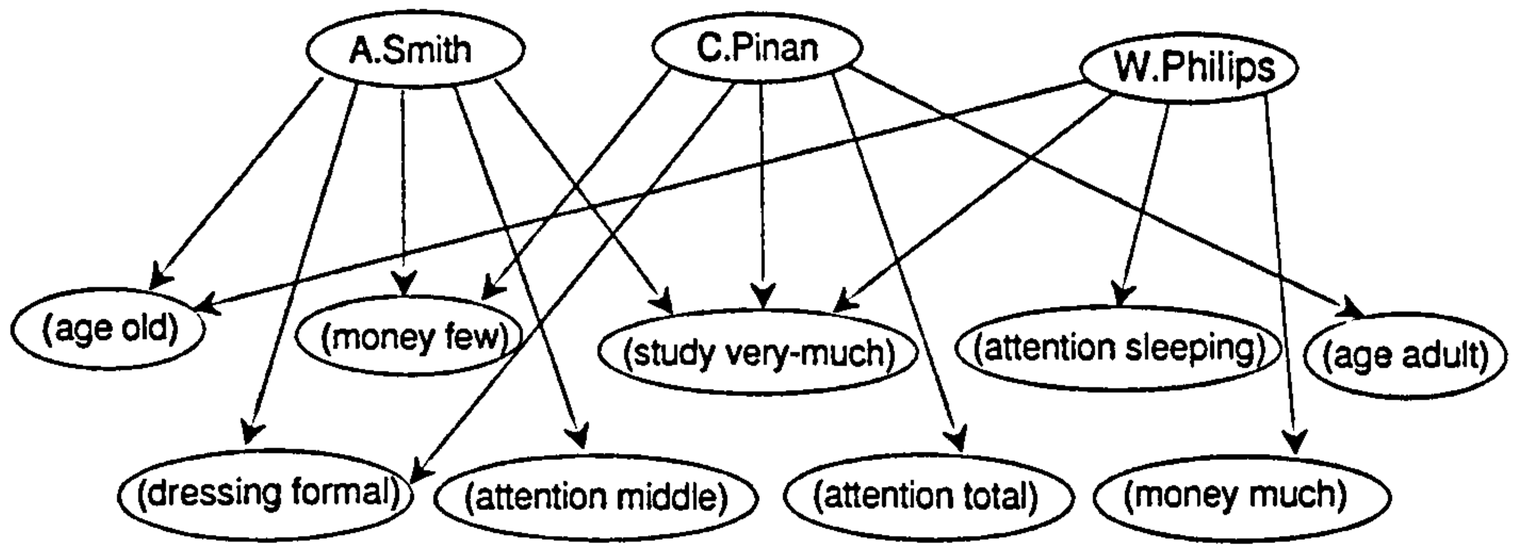


Figure 3.18: Bayesian network used to classify the test case.

After propagation in the Bayesian network of Fig. 3.18 the posterior probabilities of the examples in the TEACHER category are the following:

Exemplar: A.Smith	Prob: 0.76
Exemplar: W.Philips	Prob: 0.13
Exemplar: C.Pinan	Prob: 0.09

Hence, since A.Smith is a good exemplar for J.Perez, he is considerd both a teacher and student, since A.Smith is an exemplar in both these categories.

3.6 Summary

This chapter presented the development of a probabilistic exemplar based model whose foundations are provided by Bayesian networks as a solution to the classification and learning problems in weak domains.

Section 3.1 described the following problems of developing a probabilistic exemplar based model: (i) what makes a good exemplar? (ii) what notion of similarity can be adopted? (iii) what knowledge representation can be used? (iv) how can a new case be classified? (v) how can the model learn incrementally? The subsequent sections of the chapter described how these problems were addressed and developed the probabilistic exemplar based model.

Section 3.2 presented the knowledge representation used for the proposed probabilistic exemplar based model. The representation adopted was based on Bayesian networks, where the bottom layer consisted of features and a higher layer consisted of nodes representing exemplars. The exemplars were grouped into categories which were not necessarilly disjoint.

Given the representation based on Bayesian networks, Section 3.3 described how to take advantage of Bayesian propagation methods to classify new cases. First, exemplars are ranked and categories assume the rank of their best exemplar. Then the categories are investigated in order of their rank until a suitable exemplar is found.

Section 3.4 presented the learning process used in the proposed model. The main problems addresed in this section were: (i) how can the conditional probabilities be estimated? (ii) how can the model learn incrementally? These were

addressed using the noisy or model and assuming a virtual exemplar to represent unseen cases. The model learns incrementally by considering whether a new case is a better prototype than an existing exemplar used to classify it. This decision is based on a measure of prototypicality that takes account of the focality and peripherality of the exemplar. The measures of focality and peripherality are computed by utilising the conditional probability of an exemplar representing a region of similar cases, which are described by a summary representation.

The chapter concluded with an illustrative example that showed the main features of the model.

Chapter 4

AN EMPIRICAL EVALUATION OF THE MODEL

The previous chapters of the thesis have developed a probabilistic exemplar based model. The main aim was to develop a model that learned incrementally, does not store all the cases, and produces accurate classification. The previous chapter presented the theory of the model using Bayesian networks as a basis. This chapter carries out an empirical evaluation of the extent to which the aims are achieved.

The chapter is organised as follows. Section 4.1 describes the experimental method, and Section 4.2 then describes the results obtained. Section 4.3 concludes the chapter with a summary.

4.1 Experimental Method

The objectives of the experiment are:

1. to evaluate the accuracy of the model as it learns incrementally and

2. to determine the number of exemplars retained as more cases are observed.

To evaluate the performance of the model with respect to these objectives, an experimental environment was developed and implemented in the C language. Figure 4.1 presents the top level flow diagram of the environment.

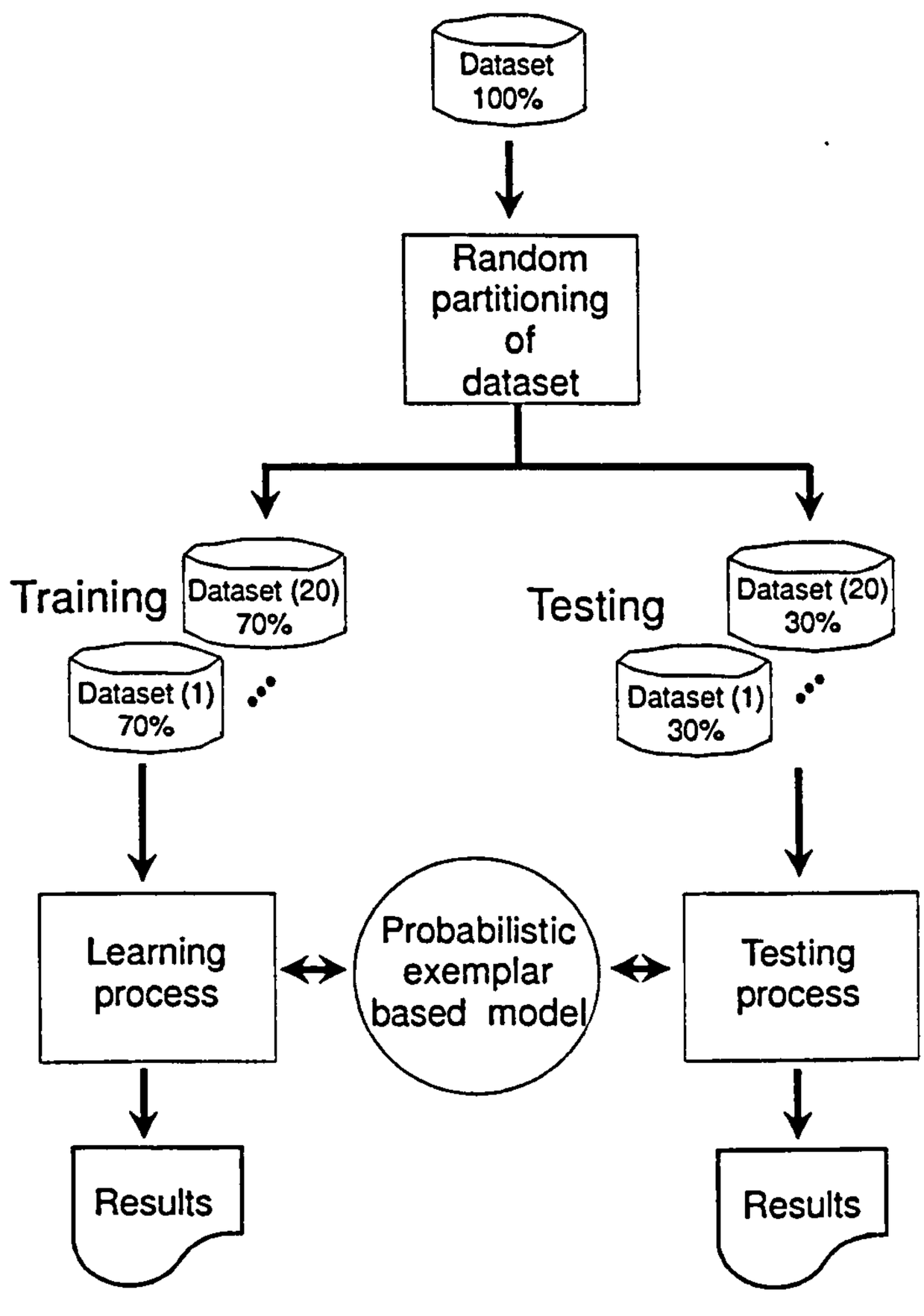


Figure 4.1: An experimental environment.

This experimental environment was used to apply the model on three datasets that are independent of this work and are publicly available [Merz & Murphy 1996]. The datasets selected are:

Votes. This dataset records the voting behaviour of the U.S.A. house of representative congressmen on 16 issues and their party affiliation. The records are classified into two categories, democrats, or republicans.

Zoo. This dataset consists of records describing animals. Each record has features like hair, legs, feathers, etc. and is classified into seven classes of animals labelled class-1, ..., class-7.

Audiology. This dataset consists of records that describe a set of illnesses in the domain of clinical audiology.

Table 4.1 summarises the characteristics of each dataset.

Table 4.1: A summary of the datasets.

Dataset name	Number of cases	Number of features	Values in features	Number of concepts	Missing values
Votes	435	16	2	2	Y
Zoo	101	16	2	7	N
Audiology	226	69	2	24	Y

These datasets were selected for various reasons. First, an important aim of the model is to be able to handle polymorphic cases; that is not all the cases should have the same features. Hence, the votes dataset was selected because it has unknown values and is therefore a reasonable test of this aim. The model aims to retain exemplars by using a measure of prototypicality. Hence, the Zoo dataset was selected because most people have intuitive exemplars of animals, and these can be compared with the exemplars retained by the model. The audiology dataset was selected primarily because it was used to evaluate Protos. Ideally, the aim was to carry out the same experiment as the one used to evaluate Protos and compare the results. However, this is not possible since the experiment involved substantial interaction with human experts. A case was presented to Protos and it attempted to classify it. The classification was then displayed to an expert who

then modified the model, by providing explanations, so that the classification agreed with the expert's classification. The audiology dataset includes only the final classification given by the expert and does not include any information about the reliability of the classification. The book describing Protos includes an appendix that requires experts to rank alternative classifications but does not include the data. Unfortunately, Bareiss (1989) no longer has the data which would enable the experiment to be repeated. Nevertheless, given the relationship of this thesis with Protos, the model had to be attempted on the audiology dataset.

For each of these datasets, the experiments aimed to evaluate the accuracy and the number of exemplars retained. This was done with the following experimental method:

1. Repeat 20 times

   (a) Randomise data set - i.e. order of cases.

   (b) Select 70% randomly for training and the remainding 30% for testing.

   (c) Train the model with the 70%.

   (d) Test the model with the 30%.

In addition to the above experiment, an attempt was made to obtain some results that could be compared with those obtained when Protos was applied to the audiology dataset. These results were obtained by approximating the procedure adopted by Bareiss to evaluate Protos, except without help from an expert. This procedure involved presenting the first 200 cases incrementally, and recording the number of the exemplars retained in each category. Then, the accuracy of the final model was tested on 26 new cases.

The following section presents the results of these experiments.

Table 4.2: Averages results for the votes dataset.

No.	Category	Training cases	Testing cases	Exemplars	Accuracy ±95% conf. int.
1	Republicans	119.20	47.8	2.1	96%±1.9%
2	Democrats	185.05	81.95	4	84%±2.0%

4.2 Results

4.2.1 Votes dataset

Table 4.2 presents the average results together with the standard deviations obtained for the votes dataset when the experiment described in Section 4.1 was carried out. The results for each of the 20 trails are given in Appendix B.

The overall accuracy obtained for this dataset was 89%. This, together with the results given in Table 4.2 show that the model has worked very well for this dataset. The number of exemplars in both categories is very low. The extent of the compression can be indicated by the ratio:

$$compression\ ratio = 1 - \frac{no.\ exemplars\ in\ category}{no.\ of\ training\ cases\ in\ category}$$

Thus, for this data set, the compression ratio for both categories is above 97%. An interesting question to ask is:

are the results better for those models with more exemplars?

Figure 4.2 presents a graph of the average accuracy against the number of the exemplars for the 20 trails. As the figure shows, for this dataset the accuracy actually reduces when more exemplars are retained and the best results are obtained when the least exemplars are retained.

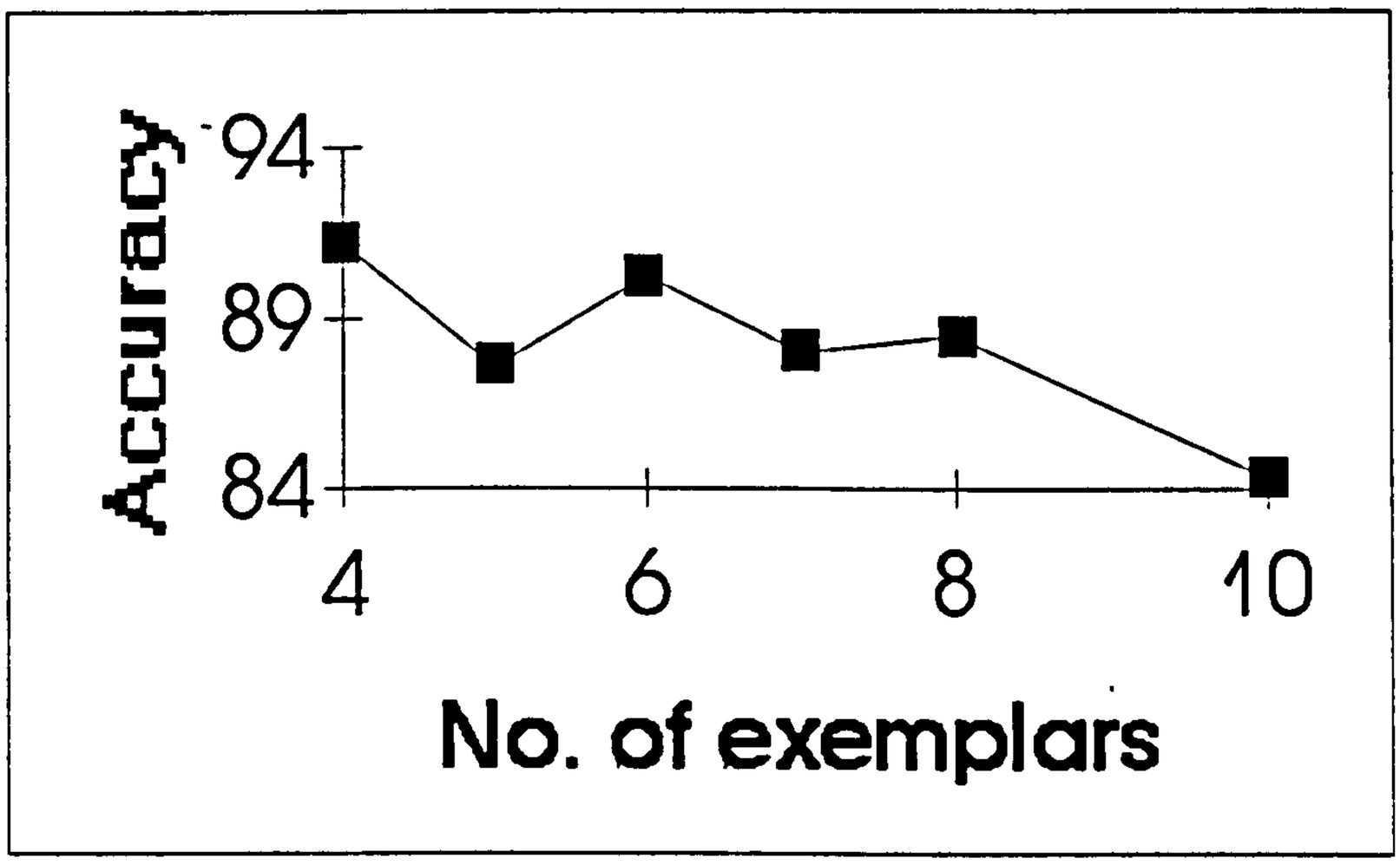


Figure 4.2: Relation between accuracy and exemplars in votes.

4.2.2 Zoo dataset

Table 4.3 presents the average results together with the standard deviations obtained for the zoo dataset when the experiment described in Section 4.1 was carried out. The results for each of the 20 trails are given in Appendix B.

The overall accuracy obtained for this dataset was 92%. This, together with

Table 4.3: Averages results for the zoo dataset.

No.	Category	Training cases	Testing cases	Exemplars	Accuracy ±95% conf. int.
1	class-1	28.4	12.6	1.35	98%±1.7%
2	class-2	14.16	5.35	1	99%±1.4%
3	class-3	3.45	1.55	1.65	16%±12.3%
4	class-4	9.3	3.7	1	100%±0%
5	class-5	2.9	1.1	1	77%±19.3%
6	class-6	5.35	2.65	1	100%±9.8%
7	class-7	7.3	2.7	1.75	80%±9.6%

Table 4.4: Common exemplars in some categories of zoo dataset.

Feature	class-1		class-2	class-6
	dolphin	cheetah	lark	housefly
hair		Y		Y
feathers			Y	
eggs			Y	Y
milk	Y	Y		
airborne			Y	Y
aquatic	Y			
predator	Y	Y		
toothed	Y	Y		
backbone	Y	Y	Y	
breathes	Y	Y	Y	Y
venomous				
fins	Y			
legs		4	2	6
tail	Y	Y	Y	
domestic				
catsize	Y	Y		

the results given in Table 4.3 show that the model has worked very well for this dataset. Most of the classes have about one exemplar representing a type of animal. Since most people have intuitive exemplars for types of animals it is worth giving Table 4.4 which shows a selection of the categories, their exemplars, and features found in some trials.

In this dataset, the overall compression ratio is also very good and more than 87%.

The number of exemplars retained in each of the 20 trails varies only between 7 and 10 exemplars. For this dataset, it is therefore not possible to detect any variation of the accuracy with respect to the number of exemplars retained. However, an interesting difference in accuracy occurs between class-3, which has a low accuracy of 16% and class-5 which has an accuracy of 77% and both classes have about 3 training cases on average. This merits further analysis and so consider Table 4.5 which presents all the cases in both classes. Class-3 consists of five

Table 4.5: Cases in classes: class-3 and class-5.

Feature	class-3					class-5			
	pit-viper	seas-nake	slow-worm	tor-toise	tua-tara	frog A	frog B	newt	toad
hair									
feathers									
eggs	Y		Y	Y	Y	Y	Y	Y	Y
milk									
airborne									
aquatic		Y				Y	Y	Y	Y
predator	Y	Y	Y		Y	Y	Y	Y	
toothed	Y	Y	Y		Y	Y	Y	Y	Y
backbone	Y	Y	Y	Y	Y	Y	Y	Y	Y
breathes	Y		Y	Y	Y	Y	Y	Y	Y
venomous	Y	Y					Y		
fins									
legs				4	4	4	4	4	4
tail	Y	Y	Y	Y	Y			Y	
domestic									
catsize				Y					

relatively different animals: pitviper, seasnake, slowworm, tortoise, and tuatara, while class-5 consists of fairly similar animals: frog, poissionous frog, newt, and toad. Since, class-3 is very polymorphic and only a few cases have been observed, the exemplars representing that category are weak and hence the accuracy of class-3 is low. However, although there are only a few cases in class-5, they are similar and the exemplars are therefore more representative of the category. Hence, the accuracy for class-5 is significantly better.

4.2.3 Audiology dataset

Table 4.6 presents the average results together with the standard deviations obtained for each category of the audiology dataset when the experiment described in Section 4.1 was carried out. The original dataset includes a category named *cochlear_unknown* which appears to consist of all cases that the experts failed to

Table 4.6: Averages results for the audiology dataset.

No.	Category	Training cases	Testing cases	Exemplars	Accuracy ±95% conf. int.
1	mix_coch_age_fix	0.70	0.30	0.7	0%±0%
2	mix_coch_age_ot_med	2.90	1.10	2.9	0%±0%
3	cochlear_age	32.25	13.75	9.9	78%±8%
4	normal_ear	14.30	5.70	7.1	52%±11%
5	cochlear_poss_noise	10.85	5.15	7.3	33%±10%
6	coch_age_and_noise	12.40	5.60	4.85	46%±17%
7	acoustic_neuroma	0.80	0.20	0.8	0%±0%
8	mix_coch_unk_ser_om	2.20	0.80	1.05	44%±20%
9	cond_discontinuity	1.35	0.65	1.35	0%±0%
10	retrococh_unknown	1.70	0.30	1.7	0%±0%
11	conductive_fixation	4.40	1.60	1	100%±0%
12	bells_palsy	0.70	0.30	0.7	0%±0%
13	coch_noi_and_herd	1.55	0.45	1.55	0%±0%
14	mix_coch_unk_fix	3.70	1.30	1	62%±19%
15	mix_poss_noise_om	1.55	0.45	1	100%±0%
16	otitis_media	2.95	1.05	2.95	0%±0%
17	possible_menieres	5.25	2.75	4.6	15%±10%
18	poss_brain_disord	2.65	1.35	1.85	74%±17%
19	coch_age_p_p_men	0.60	0.40	0.6	0%±0%
20	mix_coch_age_s_om	1.50	0.50	1.5	0%±0%
21	mix_coch_unk_dis	1.65	0.35	1.65	0%±0%
22	mix_poss_central_om	0.70	0.30	0.7	0%±0%
23	poss_central	0.65	0.35	0.65	0%±0%

classify. Consequently, in the initial experiments, it resulted in many exemplars and therefore required substantial computation time since probabilistic propagation is computationally expensive. Hence, to enable 20 random trials to be carried out in reasonable time, this category was omitted from these experiments. The results for each of the 20 trails are given in Appendix B.

The overall accuracy obtained for this dataset was 50%. This is, of course, very low! A deeper analysis of these results is necessary in order to understand why the overall accuracy is low. First, consider Fig. 4.3 which displays the number of training cases and the accuracy with respect to the categories. This figure

confirms that those categories with low accuracies also have only a few training cases. For this dataset, 18 of 23 categories have less than 5 training cases. Not surprisingly, the accuracy for these categories is virtually zero.

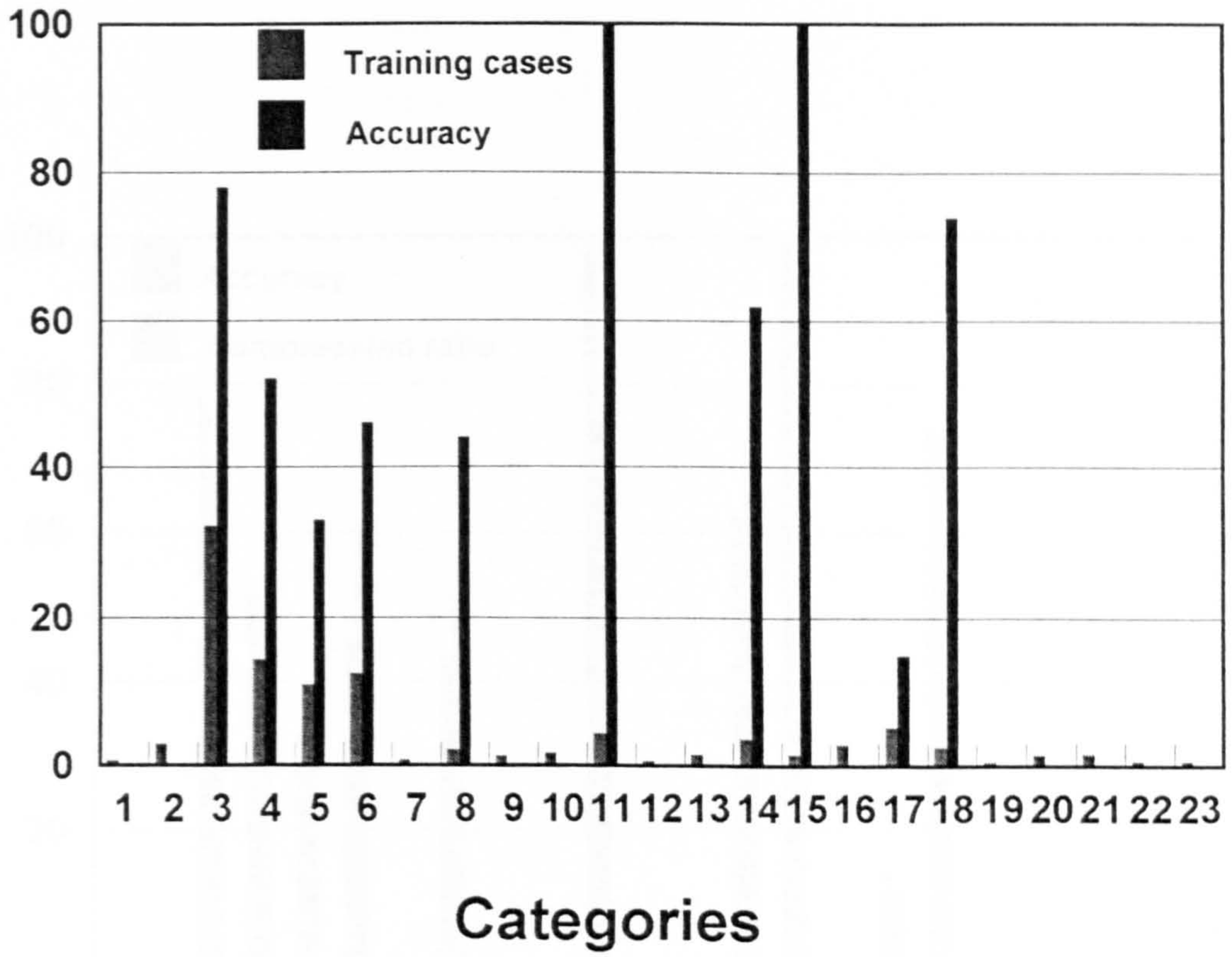


Figure 4.3: Training cases and accuracy for the audiology dataset.

Now consider Figure 4.4, which displays the accuracy and compression ratio for each category. This figure shows that the accuracy is low (i.e. less than 20%) for categories where the compression ratio for a category is low (i.e. less than 20%). Thus, for example, the compression ratio for the *possible_menieres* category (number 17 in the figure) is just 16% and has an accuracy of 15%. In contrast the *conductive_fixation* category (number 11 in the figure) has a compression ratio of 77% and 100% accuracy. Low compression ratios are indicative of categories that have not observed enough cases to cover the category. In this dataset, a closer look

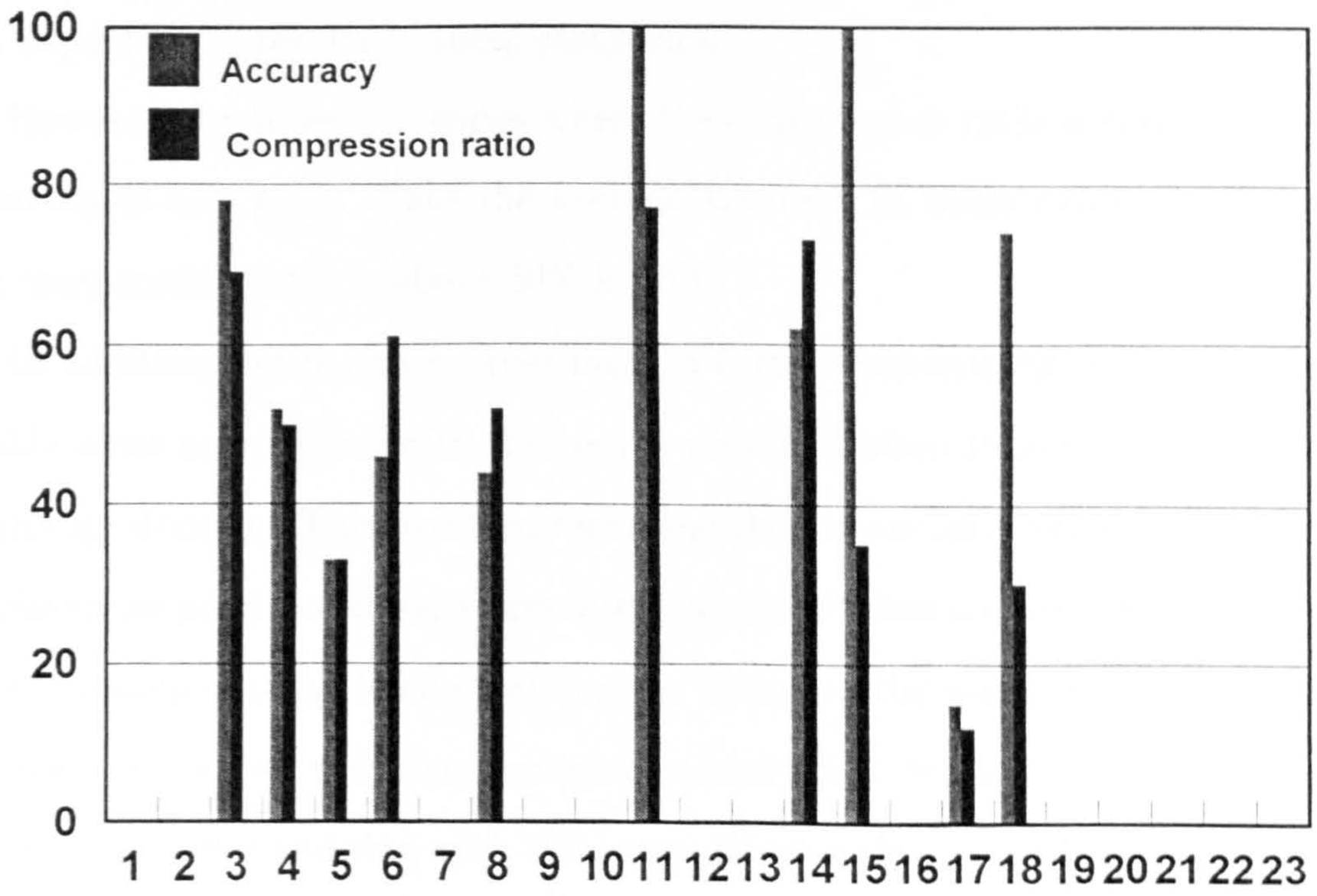


Figure 4.4: Accuracy and compression ratios for the audiology dataset.

shows that the exemplars retained in the categories with low compression ratios are sufficiently different and therefore necessary to retain. In general, categories that are polymorphic will have lower compression ratios than categories that are not polymorphic. For this dataset, there are not enough training cases to conclude whether the categories with low compression ratios are naturally more polymorphic than the other categories. All that can be concluded is that these two factors, namely lack of training cases and polymorphism, mean that one can not expect good results in these categories.

However, for those categories where the compression ratio is good, the overall accuracy is also good. Thus the average accuracy of those categories for which the compression ratio is above 50% is 66%.

In addition to the above experiment, a further experiment was carried out to enable some comparison with the results obtained when Protos was tested in the audiology domain. This involved repeating the experiment conducted by Bareiss as closely as possible. Bareiss presented cases to Protos and used experts to train Protos incrementally. While training, he observed the number of the exemplars retained and noted the accuracy incrementally (i.e. while training) at intervals of 50 cases. After training with 200 cases, Bareiss also tested the accuracy on 26 further cases. Table 4.7 presents the results obtained in that experiment [Bareiss 1989, p90]. The second column of this table, labelled '1st. Correct' refers to the accuracy when the first category proposed by Protos was checked by an expert. Since Protos does not adopt probabilities, it is difficult to utilise thresholds to determine if a classification is geniune or whether it is unknown.

A similar experiment was conducted but without the additional help of an expert and the number of exemplars were recorded at intervals of 50 cases. For this experiment, the *cochlear_unknown* category was included since Bareiss included it in his experiments and only one trial was carried out. However, the accuracy was

Table 4.7: Results reported by Bareiss.

Cases	1st. Correct
1-50	55.0
51-100	72.7
101-150	59.1
151-200	44.7
1-200	57.7
test	92.3

Table 4.8: The incremental learning with audiology dataset.

Cases	Categories	Exemplars	Accuracy	Unclassified
1-50	12	34	46%	50%
1-100	20	64	62%	19%
1-150	24	78	65%	4%
1-200	24	84	65%	4%

not recorded since expert help was not utilised, and the results obtained would not have been comparable[1].

The results obtained are summarised in Table 4.8. The table shows the incremental behaviour on the same 26 cases used in Bareiss's experiment. The accuracy column records the number of cases that are succesfully classified. The accuracy column is a little pesimistic since it interprets those cases that are not classified (i.e. those below the threshold of 0.75) as incorrect. The column labelled 'unclassified' gives the percentage of cases not classified. The accuracy column shows the model improves incrementally as more training cases are observed. The final accuracy, of 65%, is not as high as Bareiss obtained. There may be several reasons for this difference.

- In the results obtained with Protos (Table 4.7) there is a noticeable increase in the accuracy from the incremental tests performed (i.e. from 44.7% to 72.7%) to the final accuracy on the 26 cases (92.3%). This may be a result of the additional help provided by the expert.

[1]Bareiss was approached for the data collected, but unfortunately, the experiment was conducted 10 years ago and he no longer has the required information.

- The experiment performed is just one trial and the results may not be significantly accurate. For example in the earlier experiment where 20 trials were performed, the results varied from an accuracy of 64% to 38% for the same dataset.

- Some of the categories appear to be joint categories, which in the dataset, are treated as separate categories. So for example, there are categories labelled *coch_age_and_noise* and *cochlear_age*. It is unclear whether the experts had been told to treat these categories as disjoint categories or how the results have been interpreted when a more general category has been proposed by Protos. Appendix E of Bareiss (1989) includes a questionnaire that asks experts to rank the possible categories, suggesting that this information may have been utilised, but one can not be certain without having the actual data.

The first two reasons are possible but there is little that can be done to investigate them without the availability of the Protos model for the audiology dataset. The third possibility was investigated by examining a number of the cases that were wrongly classified. When this was done, it became apparent that a number of test cases that were treated as incorrect classifications were classified into related categories and the classification could be justified in terms of the observed cases. As an example, consider the 4 cases shown in Table 4.9 that are labelled T3, P43, P192, P139. The exemplars P192 and P139 represent the category *coch_age_and_noise*, while the exemplar P43 represents the category *cochlear_age* and T3 is the test case that is classified to be in category *coch_age_and_noise* by the expert. The model however, suggests the category *cochlear_age*. From Table 4.9, it is apparent that more features of the exemplar P43 are present (7 out of 9) than those of P192 (9 out of 13) or of P139 (6 out of 11). Given the similarity, and that the proposed category is not disjoint from the expected category,

Table 4.9: Features in test case T3 and exemplars P43, P192, and P139.

Feature	T3	P43	P192	P139
age_gt_60	Y	Y	Y	Y
mod_sn_gt_3k	Y			
history(noise)	Y		Y	Y
bone(unmeasured)	Y		Y	
air(normal)	Y			
speech(good)	Y			
tymp(a)	Y	Y	Y	Y
static(normal)	Y	Y	Y	Y
ar_u(normal)	Y	Y	Y	Y
ar_c(normal)	Y	Y	Y	Y
o_ar_u(normal)	Y	Y	Y	
o_ar_c(normal)	Y	Y	Y	
speech(normal)		Y		
air(mild)		Y	Y	
notch_4k			Y	
m_m_sn_gt_1k			Y	
speech(unmesurated)			Y	
speech(very-poor)				Y
or_ar_c(elevated)				Y
or_ar_u(elevated)				Y
air(moderate)				Y
boneAbnormal				Y

there is some doubt about recording this as an incorrect classification. There are 2 cases like this one and if this is taken into account then the accuracy would be 73%. This is still short of the 92% accuracy reported for the Protos' experiment. Hence, without having access to the Protos model for audiology, one can only hypothesise that the additional help of an expert resulted in a significant improvement to the Protos results.

The number of exemplars retained is more comparable than the accuracies and Table 4.10 gives the exemplars retained by Protos and the probabilistic exemplar based model (PEMB). As the table shows, in general, the model retains fewer exemplars than Protos.

Table 4.10: Exemplars retained by Protos and PEMB for audiology dataset.

No.	Category	Protos	PEBM
1	mix_coch_age_fix	1	1
2	mix_coch_age_ot_med	3	4
3	cochlear_age	20	13
4	normal_ear	16	6
5	cochlear_poss_noise	9	8
6	coch_age_and_noise	8	5
7	acoustic_neuroma	1	1
8	mix_coch_unk_ser_om	3	1
9	cond_discontinuity	2	2
10	retrococh_unknown	2	2
11	conductive_fixation	1	1
12	bells_palsy	1	1
13	coch_noi_and_herd	2	2
14	mix_coch_unk_fix	3	1
15	mix_poss_noise_om	2	1
16	otitis_media	4	4
17	possible_menieres	6	7
18	poss_brain_disord	2	2
19	coch_age_p_p_men	1	1
20	mix_coch_age_s_om	1	2
21	mix_coch_unk_dis	1	2
22	mix_poss_central_om	1	1
23	poss_central	1	1
24	cochlear_unknown	28	15

4.2.4 Conclusions

This chapter has presented an empirical evaluation of the model by testing it on three different datasets. The main aims of the model are:

- not to store all the cases but to learn prototypical exemplars,
- to learn models that are accurate and
- to learn incrementally.

The results show that the model performs well with respect to these aims. For the votes and the zoo datasets, the compression ratio is above 85% and the overall acuracy is above 89%. The compresion ratio for the audilogy dataset was 46.5% and the accuracy was much lower at 50%. A closer analysis of the audiology results shows that there are several categories where there are only a few training cases and the accuracies of these categories is therefore low. The model cannot, of course, be confident about an exemplar until it represents a reasonable number of cases. In these experiments, the compression ratio gives an indication of the number of cases represented by the exemplars. In the case of the audiology dataset, there is strong correlation between low compression ratios and low accuracies within categories.

An attempt was also made to repeat an experiment that was used to test the Protos system. Although the accuracy results are not comparable, due to lack of information about the original data, the results of the number of the exemplars retained is comparable. The results obtained show that the model developed in this thesis retains fewer exemplars in each category for a similar experiment.

In addition, the accuracy of the model was observed as more training cases were presented. In general, the accuracy increases and the rate of improvement reduces as more cases are observed.

In conclusion, the experiments show that the probabilistic exemplar based system learns models that have high accuracy by retaining only a few of the cases, provided there are sufficient cases to cover the variability of the categories. That is, categories that are very polymorphic require more training cases than categories that are not particularly polymorphic. In cases where a category does not have sufficient exemplars, the compression ratio is low, and can therefore be used as a measure of the extent to which the exemplars cover a category.

4.3 Summary

This chapter has carried out an empirical evaluation of the probabilistic exemplar based model. The model was implemented and tested on three datasets. The experiments involved training the model using 70% of a dataset and testing by using the remaining 30%. Twenty random trials were carried out for each dataset and the average accuracy and the number of exemplars retained calculated.

For two of the datasets, namely votes and zoo, high accuracies were obtained with the retention of only a few exemplars. The results for the third dataset, audiology, were significantly poorer in that the overall accuracy was 50% although the compression ratio was 46.5%. A more detailed analysis of these results showed that a number of categories in this dataset had only a few training cases. Consequently, the accuracies in these categories were low and contribute to the low overall accuracy for this dataset.

An attempt was made to repeat the experiment conducted to evaluate Protos so that the results could be compared with the model developed in this thesis. Although the experiment could not be repeated satisfactorily, since the original expert data were unavailable, the number of exemplars retained should be comparable. The model developed compares favorably in that it retains fewer exemplars per category.

The main conclusion of the evaluation was that the probabilistic exemplar based system works well when there are sufficient cases to cover the variability of the categories..

Chapter 5

RELATED WORK

This thesis has developed a probabilistic exemplar based model. Chapter 3 presented the theory and Chapter 4 presented an empirical evaluation of the model. This chapter aims to contrast the model with related work.

The probabilistic exemplar based model addresses problems and issues in the areas of case based reasoning (CBR), machine learning, and probabilistic classification. Hence, the developed model needs to be contrasted with research in these three areas. Since each of these areas is broad in its own right, some care must be taken to select the systems that should be compared. To facilitate this, important systems in each of the broad categories were identified. Figure 5.1 shows the three areas together with a selection of important systems.

The following sections select some systems from each of these areas and contrasts them with the model developed in this thesis. The thesis motivated and developed the probabilistic exemplar based model with respect to (i) the representation used or memory organisation, (ii) the classification process, and (iii) the learning process. Hence, in contrasting the different systems, first each system will be summarised with respect to these three references. Then, the main differences will be summarised, again with respect to these reference points.

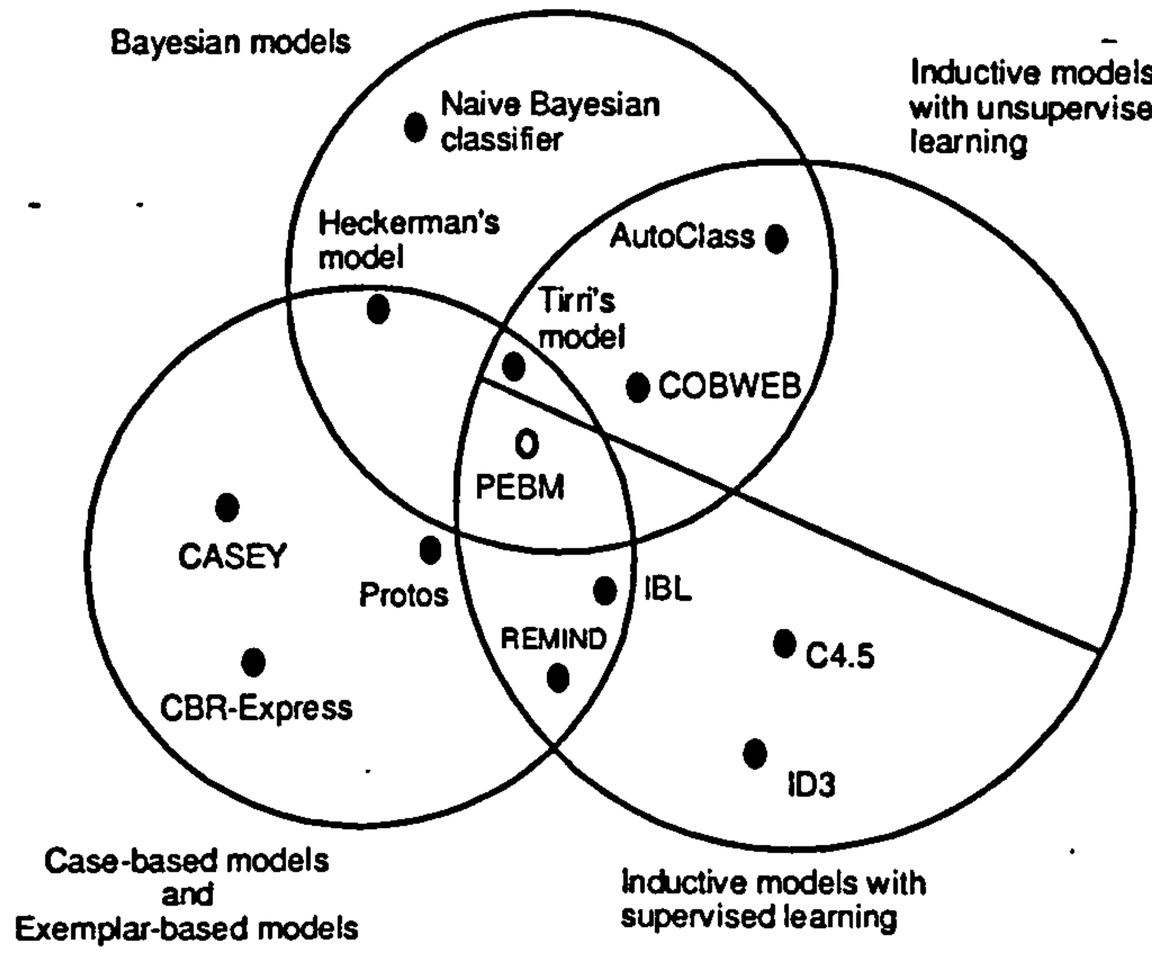


Figure 5.1: Classification of related work.

5.1 Case Based and Exemplar Based Models

The category of case based reasoning models includes a range of systems. There are some commercial tools, such as CBR express and ReMind which provide relatively standard facilities for creating indexes, and using simple retrieval algorithms [Althoff et al. 1995]. A part from following the basic CBR cycle, the relationship between these commercial tools and the model developed in this thesis is not very interesting. This category also includes more advanced systems such as CASEY, and Protos, which aim to address the issues tackled in this thesis. Hence, this section contrasts the developed model with these systems.

5.1.1 The CASEY system

CASEY [Koton 1988] is one of a number of systems that have emerged from Schank's original dynamic memory model [Schank 1982]. These systems, such as MEDIATOR [Kolodner & Simpson 1989] and JULIA [Hinrichs 1989], utilised a

representation known as discrimination network. In this representation, a network is used to partition the cases, based on the semantic similarity of the cases. Each internal node of the network can be viewed as a question that divides the cases stored in its children, where each child represents cases that correspond to one of the possible answers to the question. Each leaf node contains the cases which have the properties obtained by tracing the path from the node to the root of the network.

Given this representation, case retrieval is carried out by starting with the root node and following a path determined by the answers contained in the new case. All the cases below the final node of this path represent the similar cases that need to be considered in more detail. These cases are then evaluated by using a similarity metric and the nearest neighbour is selected as the most similar.

CASEY always learns from a problem solving case. When a new case is classified by a leaf node in the discrimination network then, CASEY stores the new case if it is significantly different from the store case. If the new case is very similar (i.e. most of the features match) then it simply updates the importance of the features used in the similarity metric. If the new case cannot be classified or it is wrongly classified, CASEY needs to reorganise the discrimination network so that the new case is included.

The main differences between CASEY and the probabilistic exemplar based model (PEBM) can be summarised with respect to the three references points as follows.

Representation. The representation adopted by CASEY is more hierarchical than the one adopted in this thesis. The representation does not explicitly address noisy information or represent uncertainty. Both representations aim to cluster regions of similar cases. CASEY clusters cases in terms of the possible values of the features of the case while the model developed in

this thesis, PEBM, clusters the cases using exemplars which are represented using a Bayesian network.

Classification. In CASEY, classification is based on following a path from the root towards a leaf node based on the answers contained in the new case. The new case is then compared to all the cases below the final node of this path by using a similarity metric. One significant disadvantage of this approach is that it is unclear how missing values are handled. That is, if an answer to a question is missing, then all possible values of the missing feature need to be considered and may result in many more cases that need to be compared with the new case. For polymorphic cases, a feature is not just missing, it is not present, and it is unclear how discrimination networks cope with this problem. In contrast, the representation adopted by PEBM enables missing values or polymorphic cases to be supported and classification is carried out by using probabilistic propagation.

Learning. The learning process used by CASEY is limited in that it does not learn the initial hierarchy. Also, the similarity metric needs to be defined. However, both may change as new cases are classified. This means that the hierarchical representation used by CASEY is appropriate when the semantic structure of a domain is known in advance so that the important features can be used as discrimination questions. However, if the semantic structure is not known, then identifying a suitable structure is difficult. In contrast, PEBM learns incrementally, from the data.

5.1.2 The Protos system

Protos [Bareiss 1989] is a system that has the closest relationship with the model developed in this thesis. The Protos system, which actually inspired this research project, integrates a method based on exemplars for concept representation, a

method to classify, and a method of learning, as a solution to the category formation problem.

Protos organises its memory in a semantic network called a category structure which represents concepts using exemplars. The model includes four types of indices: remindings, censors, prototypicality, and difference links. A reminding is a feature that is associated with categories or exemplars that can be expected to be relevant for a new case containing the feature. A censor is a negative reminding that excludes a category or exemplar when a feature is present. Prototypicality is the importance that each exemplar has in the category. Difference links point from one exemplar to another exemplar that should also be considered when searching for similar cases.

These indices are used as follows when a new case needs to be classified. First the remindings are used to propose categories that should be investigated. Then, the strongest exemplars in the category, which are determined by the prototypicality indices, are considered and matched with the new case. This matching process relates features in the exemplar with features in the new case. Identical features are given a weight that is the importance of the feature for the category. Features that can be related by an explanation are given a weight that is computed by using heuristics based on the qualifiers used to relate the features. If a suitable exemplar is not found, difference links are followed to investigate other exemplars. Eventually, a similar exemplar is found or no suitable exemplar is available.

Protos learns in various ways. When a new case is not classified, or wrongly classified, then Protos interacts with the expert to acquire new information that can modify the semantic structure. For example, it learns how the features of the new exemplar contribute to the classification of new exemplars in the category through expert explanations. Protos also learns remindings by analysing expert

explanations of the relevance of the case features to the category. Protos empirically estimates prototypicality ratings, which are used for intracategory indexing. Protos also learns failure indices in response to problem solving failures.

The main differences between the model developed in this thesis and Protos can be summarised as follows.

Representation. In Protos, an exemplar is represented by a case while in PEBM, an exemplar is represented by a Bayesian network, where the random variables of the network are determined by the prototypical case. Protos makes heavy use of indices while PEBM only uses the relationship between categories and exemplars. Protos has no explicit way of representing joint categories. That is, although an exemplar can be duplicated in two different categories, Protos can not conclude that a case is in both categories.

Classification. Protos relies heavily on the indices to retrieve similar cases. That is, remindings and difference links are used to identify potential categories and exemplars. The prototypicality ratings are used to rank the exemplars and a matching process is used to measure the similarity of an exemplar with a new case. All this is done with heuristics. The heuristics have evolved as a result of one application and are difficult to justify with any theory. In contrast, PEBM classifies using probabilistic propagation and therefore has better theoretical foundations. There is also less reliance on the use of indices. That is, remindings, censors, or difference links are not used. Unlike Protos, the measure of similarity is not heuristic but a probability of similarity.

Learning. In the learning phase, Protos learns the importance of its features and indices by explanation, while the proposed model learns directly from the data. Also, Protos retains those cases that are not correctly classified

as new exemplars. Cases that are correctly classified result in an increase of an exemplar's prototypicality but are discarded. In contrast, the proposed model attempts to use the notion of prototypicality to determine if a new case, that is correctly classified, would make a better exemplar.

Protos determines the prototypicality of an exemplar by the number of cases that it represents. The proposed model uses a measure of prototypicality based on the concepts of focality and peripherality identified by Biberman (1995) as characteristics of prototypicality.

5.2 Inductive Learning Models

Research in the area of inductive learning models can be subdivided into systems that perform supervised learning and systems that perform unsupervised learning. Supervised learning systems are trained with examples where a class is known, whereas unsupervised learning systems aim to identify clusters without a known class.

Examples of systems that perform supervised learning include tree induction systems such as ID3 [Quinlan 1996] and C4.5 [Quinlan 1992]. These systems aim to produce decision trees by using evaluation functions to select the nodes of the decision tree from the available attributes. As such, they are not similar to PEBM and are not described further in this section.

Examples of systems that perform unsupervised learning include COBWEB [Fisher 1990], CLASSIT [Gennari et al. 1990] and AutoClass [Cheeseman et al. 1990]. From these systems, the COBWEB system is the most interesting since it determines clusters incrementally. Hence, this section outlines COBWEB and contrasts it with PEBM.

COBWEB system

The main aim of COBWEB is to identify clusters so that they can be used as a way of summarising and explaining data. Since the class is not identified, COBWEB has to use a utility function to measure the quality of a cluster.

It organises its memory as a hierarchy of clusters where the terminal nodes are instances and the non terminal nodes represent clusters. Each parent node represents a cluster that is the union of the clusters represented by its children. Each cluster is described by listing the features and their conditional probabilities given the cluster. Fisher (1996) gave the example shown in Fig. 5.2 which has three variables: size, shape and colour.

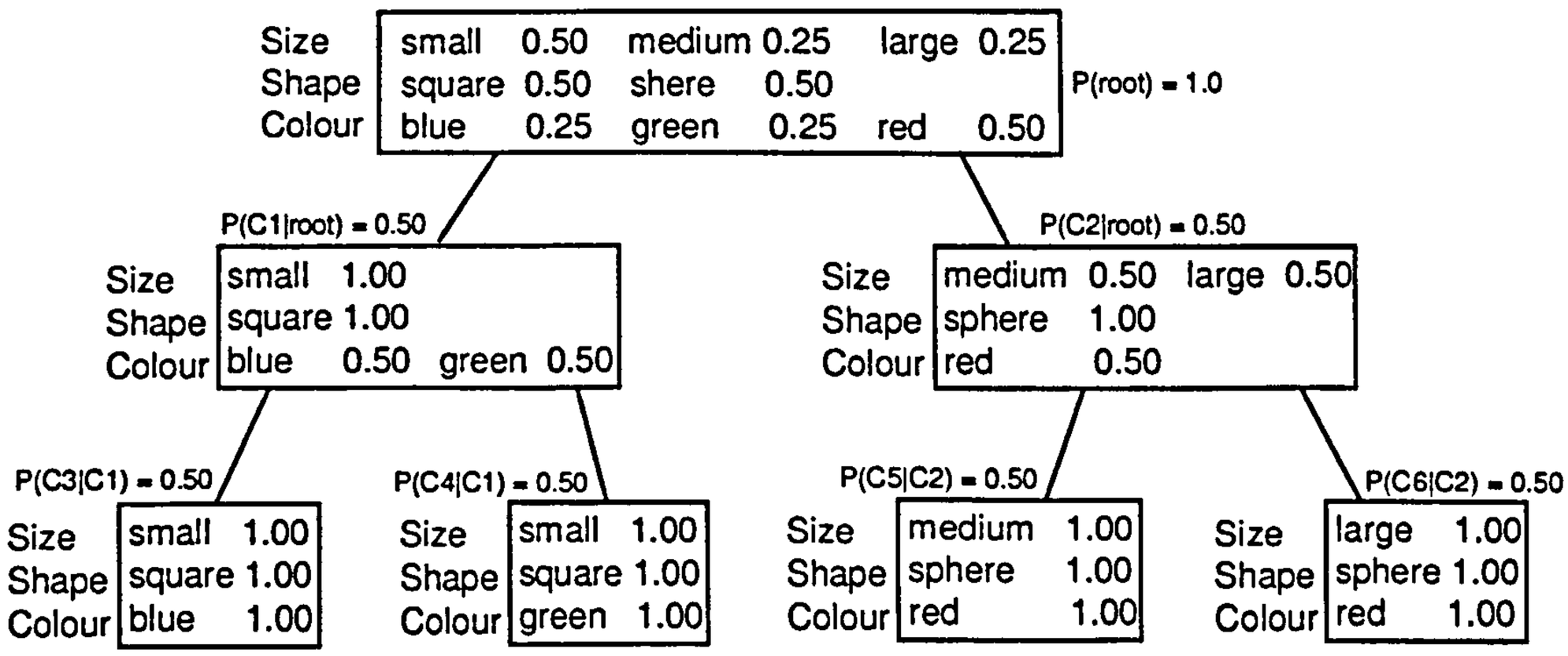


Figure 5.2: A probabilistic categorisation tree.

Given this representation, classification is carried out as follows. A new example is placed in the root cluster and the utility function is evaluated. If this improves the utility then the example is placed in the child node that improves the utility most. This process is repeated until a leaf is reached or placing the example in a cluster reduces the utility. If the utility reduces, then a separate

cluster is created at that level instead.

In addition to learning by creating separate clusters, COBWEB also has operators for merging two clusters and for dividing categories. These operators can also be used to improve the utility.

The differences between COBWEB and PEBM can be summarised as follows.

Representation. The main difference between the representations used by COBWEB and PEBM is that COBWEB does not utilise categories. However, within a category in PEBM the situation is similar to COBWEB in that clustering is required and unsupervised learning is used.

COBWEB uses a hierarchical representation that is able to represent finer regions than the one adopted by PEBM. PEBM uses a Bayesian network that represents the regions. In an exemplar based model one level of regions appears to be natural but it would be interesting to find applications where multiple levels of clustering is required.

Classification. COBWEB classifies a new example by finding the best home cluster for it in a top down manner. That is, the cluster that results in the largest improvement in utility when the example is placed in the cluster is identified starting with the root cluster and specialising until the finest cluster is found. In contrast, PEBM uses probabilistic propagation to determine the probability of an exemplar representing the example.

Learning. Learning in COBWEB is achieved by subdividing regions, introducing new regions, or merging regions so as to optimise a utility function. In PEBM, learning is achieved by growing regions around exemplars and by retaining exemplars that best represent a region.

5.3 Bayesian Probabilistic Approaches

The model developed in the thesis makes significant use of Bayesian networks. Hence, it is reasonable to ask if just the use of Bayesian classification models on their own are adequate. Hence, this section includes a summary of the most common Bayesian classifier, known as the *naive Bayesian classifier.*

Like the work described in this thesis, Tirri and Myllymäki's work utilises Bayesian networks in the area of CBR. Hence, this section also contrasts the PEBM model with their research.

The section concludes with a summary of other systems that have utilised CBR and Bayesian networks.

5.3.1 The naive Bayesian classifier

The naive Bayesian classifier is a probabilistic classification model which takes the form shown in Fig. 5.3, where C_i denotes the categories and f_j denotes the features. Notice that the categories in this representation are assumed to be independent and the features are assumed to be independent given a category.

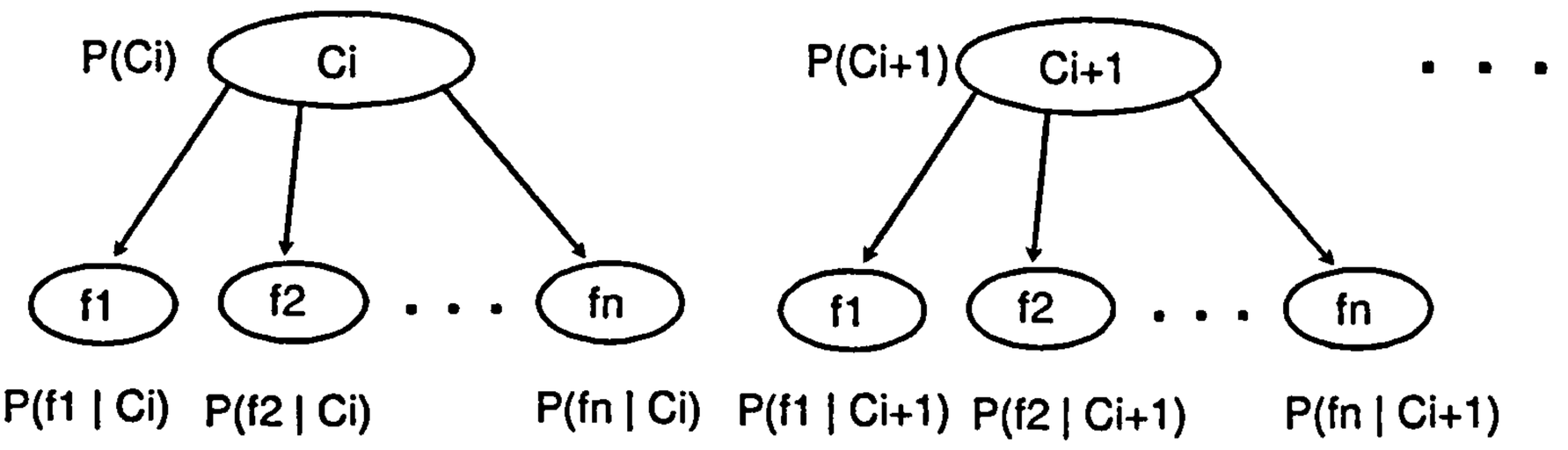


Figure 5.3: The naive Bayesian classifier.

In order to classify a new case I, the model simply applies Bayes' theorem for

each category as follows.

$$P(C_i \mid I) = \frac{P(C_i)\, P(I \mid C_i)}{P(I)}$$

Since I is the conjunction of features f_j then the equation above can be expanded as follows.

$$P(C_i \mid I) = \frac{P(C_i)\, P(\bigwedge f_j \mid C_i)}{\sum_k P(\bigwedge f_j \mid C_k)\, P(C_k)}$$

As can been seen in this equation, the classifier needs to know the prior probabilities of the concepts and the conditional probabilities of the attributes to be able to compute the posterior probability of the categories given the new instance. After calculating these probabilities for each description, this model classifies the new instance in the concept with the highest probability.

As the classifier assumes that the attributes are independent, then the probability of the conjunction of the features given a category can be computed by the product of the conditional probabilities of the features as follows.

$$P(\bigwedge f_j \mid C_i) = \prod_j P(f_j \mid C_K)$$

This classifier contrasts with the proposed model in the following ways.

Representation. The naive Bayesian classifier aims to predict a category using features while PEBM aims to predict exemplars using features. This is a significant difference that can be illustrated with the following example. For example, the category bird has a dominant feature *flies*. Given an ostrich the probability of it being in the category would not be high. In contrast, the probability of it being an exemplar in the birds category would be high.

Classification. Both approaches to classification use Bayes' rule. However, an important difference is that the naive Bayesian classifier assumes that the

categories are independent given a feature but this assumption cannot be made for exemplars. That is, given a feature, it cannot be assumed that the exemplars are independent. This means that the naive Bayesian classifier is much more efficient than PEBM when classifying.

Learning. In terms of learning, the main difference is that the naive Bayesian classifier needs all the data in advance, while PEBM learns incrementally.

5.3.2 Tirri and Myllymäkis' model

Myllymäki & Tirri (1994) have developed a model that integrates Bayesian reasoning and CBR in a connectionist network for case matching and adaptation.

This model represents the case base by a Bayesian network as shown in Fig. 5.4 (a). The cases (upper nodes) are represented as binary random variables. The attributes (low nodes) are represented as random variables that can have n possible values. The model represents a case c_k as:

$$c_k = (P_k(a_{11}), \cdots, P_k(a_{1n_1}), \cdots, (P_k(a_{m1}), \cdots, P_k(a_{mn_m}))$$

where $P_k(a_{i1}), \cdots, P_k(a_{in_i})$ expresses the probability distribution for the values of attribute A_i when the case c_k is in question.

Since this representation can contain values between 0 and 1, the Myllymäki and Tirris model regards a case as a "prototypical" representation of a class of similar cases.

The input to the Bayesian network is given by defining an initial probability distribution for each attribute value of a case, c_0:

$$c_0 = (P_0(a_{11}), \cdots, P_0(a_{1n_1}), \cdots, P_0(a_{m1}), \cdots, P_0(a_{mn_m}))$$

Given an input case c_o, the similarity to C_k is determined by computing $P(C_k = 1 \mid C_0 = c_0)$

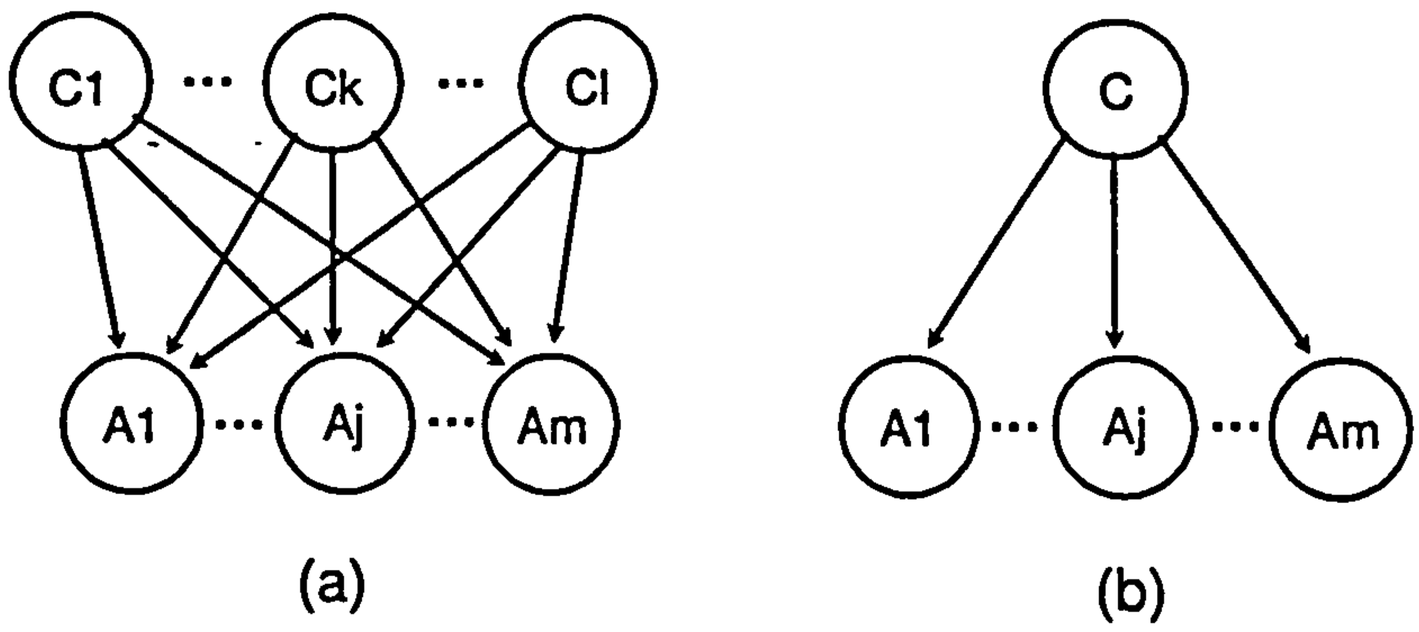


Figure 5.4: Case base as a (a) multiply connected and (b) tree.

In order to reduce the complexity of the algorithm for propagating probabilities in a multiply connected network, Fig. 5.4 (a), the model assumes that the cases $c_1, \cdots, c_l$ are mutually exclusive values of a single random variable C and all the variables A_i are conditionally independent given the value of the variable C. These assumptions enable the multiply connected network to be transformed to a tree as shown in Fig. 5.4 (b) and therefore allow the use of a more efficient propagation algorithm.

The main differences between this model and PEBM model are as follows.

Representation. The initial representation proposed in Myllymäki and Tirris' model (shown in Fig. 5.4 (a)) is similar to the one adopted in this thesis. However, the one actually used by Myllymäki and Tirri (shown in Fig. 5.4 (b)) is simpler and assumes that the cases are mutually exclusive. The extent to which this assumption holds is unclear since a new case can be expected to be similar to a number of previous cases. However, the assumption would hold if mutually exclusive prototypes could be found. In more recent work [Tirri et al. 1996*a*, Tirri et al. 1996*b*], they aim to find prototypes by using a statistical clustering technique known as finite mixture

models which could be used as the cases for their earlier work.

Classification. Since the Bayesian networks adopted in Myllymäki and Tirris' model are simpler, classification is more efficient than in PEBM. The overall efficiency, however, depends on the number of cases they retain. Thus, without prototypes, (i.e. as their model is proposed) they need to adopt parallel computation methods to cope with the number of cases stored.

Learning. The kind of learning performed in Myllymäki and Tirris' model is limited to estimating the probabilities. The subsequent work that aims to identify prototypes learns by using unsupervised clustering techniques which require all the cases in advance. In contrast, PEBM learns exemplar incrementally.

In adition to Myllymäki and Tirris' work there are a number of other researchers who have used Bayesian networks and CBR. A brief summary of the main aspects of other work is as follows.[1]

- Breese & Heckerman (1995) have integrated Bayesian networks and CBR for diagnostic purposes. They used a three-layer Bayes net to link the causes of cases (called, *issues*) with observable *symptoms*. Then, when some evidence is available, it is propagated in the networks to identify the most probable cases. The most probable cases are then used as a basis for diagnosing the fault and determining a cost-effective solution.

- Chang & Harrison (1995) used a Bayesian approach to guide retrieval and indexing as part of an experimental testbed that includes several techniques and allowed a user to experiment with different instance selection algorithms. The instance selection schemes have similar goals to exemplar selection but are not based on notions of protypicality as in PEBM.

[1]The reader is referred to the paper by Aha & Chang (1996) for a more detailed account of these systems.

- Aha & Chang (1996) used a Bayesian network and CBR to work cooperatively on multiagent planning tasks. The Bayesian networks are used to characterize action selection, whereas CBR is used to determine how to implement actions. Unlike the proposed model, their model does not aim to utilize Bayesian networks for CBR, but instead combines their mutual strengths to solve a particular task.

5.4 Summary

This chapter has contrasted the model developed in this thesis with other research in the areas of CBR, machine learning, and Bayesian classification. Systems in each of these areas were summarised and the main differences identified and discussed.

Each system was described in terms of the representation used, classification approach and the learning process. Then each system was contrasted with the model developed in this thesis, again, in terms of representation, classification, and learning.

In terms of representation, PEBM is the only model that uses Bayesian networks to represent exemplars. Tirri and Myllymakis' model uses Bayesian networks but the focus is different in that they represent cases. The representations used by COBWEB and CASEY are interesting in that they allow multiple levels of clusters (or regions) to be represented whereas in PEBM, only two levels are represented.

In terms of classification, both PEBM, Tirri and Myllymakis' model use probabilistic propagation methods. However, the simplified model adopted by Tirri and Myllymakis' model enables them to adopt a simpler propagation algorithm than PEBM. Protos classifies by using its indices and a heuristic matching process, while COBWEB classifies by finding a home for the new case that maximises

models which could be used as the cases for their earlier work.

Classification. Since the Bayesian networks adopted in Myllymäki and Tirris' model are simpler, classification is more efficient than in PEBM. The overall efficiency, however, depends on the number of cases they retain. Thus, without prototypes, (i.e. as their model is proposed) they need to adopt parallel computation methods to cope with the number of cases stored.

Learning. The kind of learning performed in Myllymäki and Tirris' model is limited to estimating the probabilities. The subsequent work that aims to identify prototypes learns by using unsupervised clustering techniques which require all the cases in advance. In contrast, PEBM learns exemplar incrementally.

In adition to Myllymäki and Tirris' work there are a number of other researchers who have used Bayesian networks and CBR. A brief summary of the main aspects of other work is as follows.[1]

- Breese & Heckerman (1995) have integrated Bayesian networks and CBR for diagnostic purposes. They used a three-layer Bayes net to link the causes of cases (called, *issues*) with observable *symptoms*. Then, when some evidence is available, it is propagated in the networks to identify the most probable cases. The most probable cases are then used as a basis for diagnosing the fault and determining a cost-effective solution.

- Chang & Harrison (1995) used a Bayesian approach to guide retrieval and indexing as part of an experimental testbed that includes several techniques and allowed a user to experiment with different instance selection algorithms. The instance selection schemes have similar goals to exemplar selection but are not based on notions of protypicality as in PEBM.

[1]The reader is referred to the paper by Aha & Chang (1996) for a more detailed account of these systems.

- Aha & Chang (1996) used a Bayesian network and CBR to work cooperatively on multiagent planning tasks. The Bayesian networks are used to characterize action selection, whereas CBR is used to determine how to implement actions. Unlike the proposed model, their model does not aim to utilize Bayesian networks for CBR, but instead combines their mutual strengths to solve a particular task.

5.4 Summary

This chapter has contrasted the model developed in this thesis with other research in the areas of CBR, machine learning, and Bayesian classification. Systems in each of these areas were summarised and the main differences identified and discussed.

Each system was described in terms of the representation used, classification approach and the learning process. Then each system was contrasted with the model developed in this thesis, again, in terms of representation, classification, and learning.

In terms of representation, PEBM is the only model that uses Bayesian networks to represent exemplars. Tirri and Myllymakis' model uses Bayesian networks but the focus is different in that they represent cases. The representations used by COBWEB and CASEY are interesting in that they allow multiple levels of clusters (or regions) to be represented whereas in PEBM, only two levels are represented.

In terms of classification, both PEBM, Tirri and Myllymakis' model use probabilistic propagation methods. However, the simplified model adopted by Tirri and Myllymakis' model enables them to adopt a simpler propagation algorithm than PEBM. Protos classifies by using its indices and a heuristic matching process, while COBWEB classifies by finding a home for the new case that maximises

a utility function.

All the models adopt and require different learning processes. Tirri and Myllymakis' model only needs to learn the probabilities from all the data and is not incremental. Protos learns primarily from user provided explanations and the use of heuristics. COBWEB learns by considering the effect of creating new clusters, merging clusters, and partitioning clusters, on an evaluation function and aims to optimise its value. In contrast to all these models, PEBM learns by retaining examplars on the basis of a measure of prototypicality.

Chapter 6

CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

Case based reasoning (CBR) has become an active area for research that aims to solve new problems by adapting the solution of similar problems encountered in the past. A central research problem in CBR is the organisation of the cases. Most current CBR systems have the following characteristics:

1. They store all the past cases but partition the cases in order to make retrieval more efficient.

2. They adopt a fixed format for the cases and often require all the features in advance.

3. They often require all the cases in advance and are not incremental.

4. They do not handle noisy data and do not explicitly handle uncertainty.

One approach to these problems is to develop exemplar based models, where only prototypical cases are stored. However, before an exemplar based model can

be developed, the following questions need to be answered:

- How can an exemplar based model be represented?
- Given a new case, which exemplar, if any, represents it?
- What makes a good exemplar?
- How can an exemplar based model be learned incrementally?

This thesis has attempted to answer these questions by developing and evaluating an exemplar based model whose foundations are based on Bayesian networks. The following subsections summarise the model developed, the empirical evaluation, and contrasts the model with the related systems. The chapter concludes with directions for future work.

6.1.1 The model

The first of the above questions, that of finding a suitable representation, has to cater for weak domains where: (i) the categories are difficult to define by necessary and sufficient features, (ii) the categories can be non-disjoint, and (iii) there is uncertainty in how the categories are represented by cases.

This implies the need for a representation that is capable of representing uncertainty. Hence, the representation adopted consists of a two layered Bayesian network where the nodes in the lower level consist of the features, and the nodes in the upper level consist of the exemplars. The arcs of the network represent the strengths of the dependencies. Categories are then represented as collections of exemplars.

Figure 6.1 illustrates how the exemplar based model shown on the left of the figure is represented by the network on the right of the figure.

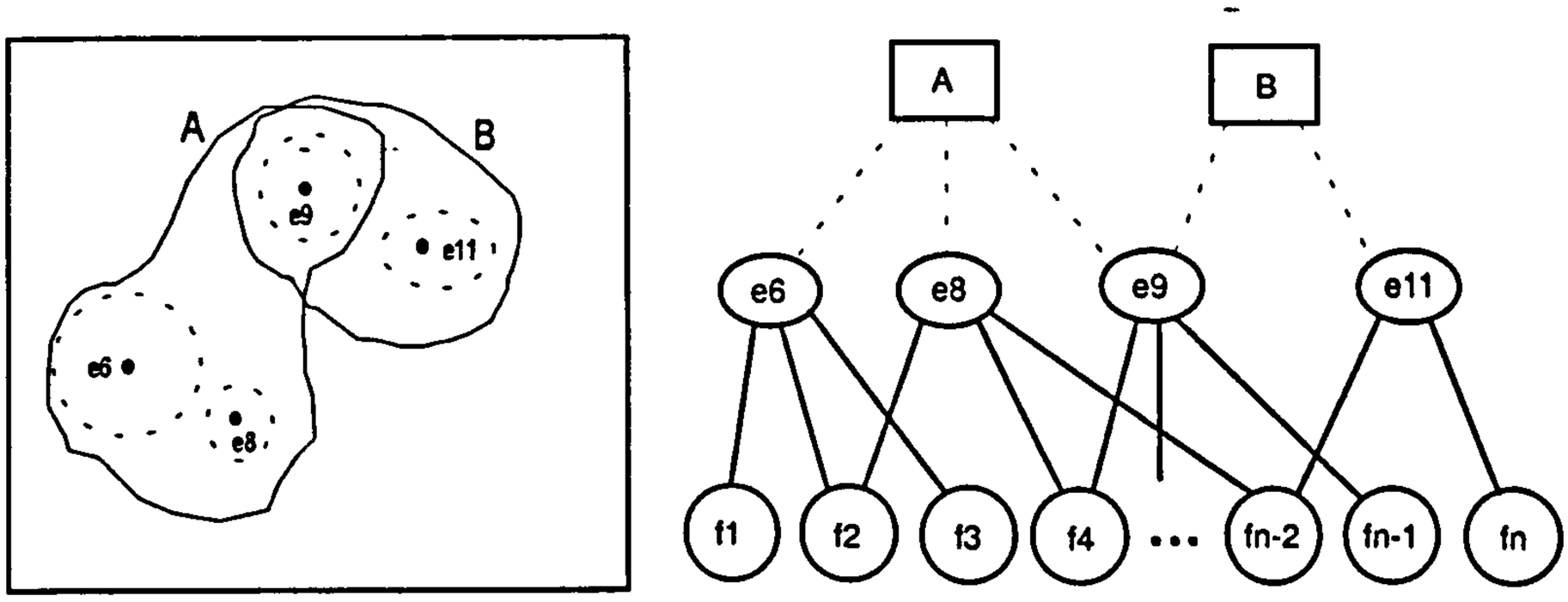


Figure 6.1: Exemplar based model and its representation.

Given this representation, the question of finding an exemplar that represents a new case can be addressed by using probabilistic methods. That is, given a new case with features $f_1, f_2, \cdots, f_q$ the probability

$$P(e \mid f_1, f_2, \cdots, f_q)$$

is the degree to which the new case is represented by the exemplar e. This can be computed by probabilistic propagation methods.

Since probabilistics propagation methods can be expensive, a ranking scheme is used to order the categories according to the most promising exemplars. Then, categories are investigated by applying probabilistic propagation within categories until a suitable exemplar is found (i.e. where the probability is above a threshold).

The third question, what makes a good exemplar, is addressed by utilizing an observation by Rösch & Mervis (1975) who argued that a case is prototypical if it has high family resemblance within the category (focality) and low family resemblance to other categories (peripherality).

Given the availability of the probability of an exemplar e_i representing a region within a category C, this notion of prototypicality can be formalised by

$$Prototypicality(e_i, C) = Focality(e_i) - Peripherality(e_i, C)$$

where the focality and peripherality measures are computed by probabilistic propagation.

Given this measure for prototypicality, if a new case is more prototypical than an existing exemplar, then it replaces that exemplar. Hence, incremental learning can take place by repeated application of this criteria as new cases are observed. However, how can the probabilistic dependencies be estimated incrementally as the model evolves? This is done by the introduction of an additional exemplar, called a *virtual exemplar* (Ve) as shown in Fig. 6.2.

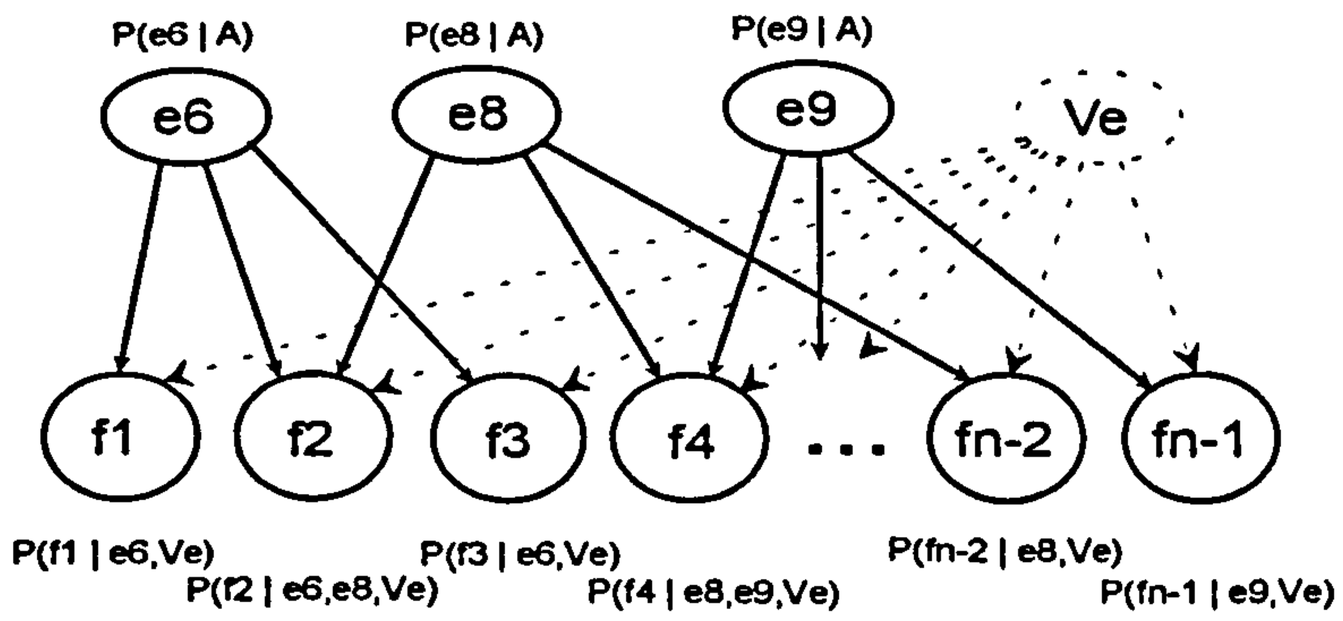


Figure 6.2: Virtual exemplar.

This virtual exemplar can be viewed as representing all the cases that have not yet been observed. The introduction of a virtual exemplar, Ve, requires estimating the strengths of the dependencies $P(f \mid Ve)$. This is done by observing that the strengths should be highest, initially, when there are no cases and should decay as more cases are observed. This observation leads to the adoption of the

following function for estimating the dependencies:

$$P(f \mid Ve) = \lambda e^{-\lambda * \alpha * n}$$

or

$$0.1 \ if \ P(f \mid Ve) < 0.1$$

where n is the number of cases in a category and α is a scaling parameter that determines the rate of decay.

6.1.2 A contrast with related systems

The developed model is related to work in three areas: CBR, machine learning and Bayesian classification models. The thesis therefore compared the model with systems in these three categories. First, a number of systems were identified and classfied in these areas and then the most related and interesting systems were contrasted.

In the area of CBR, CASEY and Protos were contrasted with the model developed in this thesis (PEBM). The representation used by CASEY is more hierarchical but does not handle polymorphic cases and it is unclear how it retrieves cases when features are missing. The kind of learning it does is also limited in comparison to the other models. The representation used by Protos is similar to PEBM in that exemplars are used to define categories. The notion of exemplar is, however, very different in that cases denote exemplars, whereas in PEBM, exemplars are represented by Bayesian networks. The classification process used by Protos is dependent on the use of indices called remindings, censors, and difference links. In contrast, classification in PEBM is achieved by probabilistic propagation. The learning mechanisms are also very different since Protos relies heavily on heuristics that learn from user provided explanations, while PEBM learns from data. The most significant difference, however, is that PEBM has foundations in probabilistic reasoning, whereas Protos appears to be

based primarily on heuristics.

The model was also compared with COBWEB, an important system in the area of inductive-learning models. Although COBWEB performs unsupervised learning, the comparison revealed an interesting aspect of the developed model. Overall, PEBM is a supervised learning model. However, within a category, where exemplars need to be learned, it performs unsupervised learning. The way PEBM performs learning is quite different. In COBWEB, learning is performed by introducing new regions (or clusters), partitioning regions, or merging reasons so as to optimise a utility function. In PEBM, learning is performed by growing regions around exemplars and retaining the most prototypical exemplars that represent a region.

Since the model uses Bayesian networks, an obvious question to ask is: how does it compare with Bayesian classification models? PEBM was therefore compared with the naive Bayesian classifier, a model well studied in the literature. The naive Bayesian classifier operates at the level of categories only and is therefore unable to make finer distinctions within categories of the kind that the exemplar based model can make. In terms of learning, the naive model requires all the data in advance, and is not incremental.

In the category of research that utilises Bayesian networks and CBR, Tirri and Myllymakis' work is the closest to this thesis. They first proposed a representation that is very similar to the one adopted in this thesis but with the exception that their upper level nodes are random variables that represent cases and not prototypes. Given the potentially large number of cases, standard propagation methods would not be practical. Hence, they assume that the cases are mutually exclusive in order to simplify the network to a tree. The extent to which this assumption holds or the effects of violating the assumption are unclear since a new case can be expected to be similar to a number of previous cases. In contrast,

PEBM does not make this assumption and uses exemplars which aim to represent regions of similar cases. This difference is also reflected in the requirements for learning, since their model only estimates the probabilities from all the cases, while PEBM identifies prototypes incrementally.

6.1.3 A summary of empirical results

Given the theoretical model of the Bayesian exemplar based model, the empirical evaluation aimed to examine how well the model worked on real datasets and whether it had the desired characteristics. In particular, the empirical evaluation tested the extent of compression achieved when only exemplars are stored and whether accurate results could be obtained when only a few of the cases were retained as exemplars.

The model was implemented in the C language and tested on three datasets available in the public domain and known as the: votes, zoo and audiology datasets. The experiments involved training the model with 70% of the data and testing with the remaining 30%. This was repeated with twenty trials and the average accuracy and the number of exemplars retained recorded. A measure, called the *compression ratio*, was used as an indication of the number of cases represented by the exemplars in a category.

For the votes and the zoo datasets, the compression ratio was above 85% and the overall accuracy was above of 89%. The compression ratio for the audiology dataset was 46.5% and the accuracy was much lower at 50%. A closer analysis of the results showed that there were several categories where only a few training cases were available and the accuracies of these categories were therefore low. The model cannot, of course, be confident about an exemplar representing a case until it represents a reasonable number of cases. Those categories that had low accuracies also had a small number of training cases.

In the case of the audiology dataset, an attempt was also made to repeat an experiment that was used to test the Protos system. The results obtained showed that the model developed in this thesis retains fewer exemplars in each category for a similar experiment.

In summary, the experiments showed that the probabilistic exemplar based system learns models that have high accuracy by retaining only a few of the cases, when there are sufficient cases to cover the variability of the categories. That is, categories that are very polymorphic require more training cases than categories that are not particularly polymorphic. In cases where a category does not have sufficient exemplars, the compression ratio is low, and can therefore be used as a measure of the extent to which the exemplars cover a category.

In conclusion, this thesis has developed a new exemplar based model whose foundations are established with Bayesian theory and which has produced good results on some test datasets.

6.2 Future Work

There are a number of areas where further research and development can be carried out. These include the following.

- The model currently adopts the propagation algorithm developed by Lauritzen & Spiegelhalter (1988). This algorithm is not efficient and better algorithms need to be found or developed for the special kind of Bayesian networks adopted in this thesis.

- The model currently only uses the information that it learns from the data. In some applications, background knowledge, such as generalisation, hierarchies may be available. Therefore, the model could be extended to utilise such background knowledge.

- The implementation of the model needs to be evaluated on a wider range of datasets and the results compared with other approaches. This may require the implementation of other approaches if they are not available.

Appendix A

Illustration of the model

In order to provide a better understanding of the example presented in Chapter 3, the training data, results of the training phase, test data, and results of the testing phase are shown in this appendix.

University dataset

In the example, the model was trained with the following training cases.

Exemplar	Category	Features
1. L.Pintos	TEACHER	attention(total) dressing(formal) money(few) age(adult)
2. L.Pineda	TEACHER	attention(total) money(few) study(very-much) age(adult)
3. W.Philips	TEACHER	age(old) attention(sleeping) money(much) study(very-much)
4. J.Gomez	TEACHER	age(old) attention(sleeping) money(sufficient) study(very-much)
5. A.Smith	TEACHER STUDENT	age(old) dressing(formal) money(few) attention(middle) study(very-much)
6. G.Leon	TEACHER	attention(sleeping) money(much) study(very-much)
7. B.Wild	TEACHER	age(old) attention(sleeping) money(sufficient) study(very-much)

Exemplar	Category	Features
8. P.Ibar	STUDENT	age(adult) attention(middle) dressing(informal) study(normal)
9. R.Abaco	STUDENT	age(adult) dressing(informal) study(normal)
10. P.DeBuen	TEACHER STUDENT	age(old) dressing(formal) attention(middle) study(very-much) has(computer)
11. S.Santana	TEACHER STUDENT	age(adult) dressing(formal) attention(middle) study(very-much) has(computer)
12. E.Zage	STUDENT	age(adult) attention(middle) dressing(informal)
13. E.Plaza	TEACHER	age(old) attention(sleeping) money(much) dressing(formal)
14. L.Garcia	STUDENT	money(few) attention(middle) age(adult) dressing(informal) study(few)

Exemplar	Category	Features
15. F.Patlan	TEACHER	age(old) attention(sleeping) dressing(formal) money(much)
16. P.Wolf	STUDENT	money(few) attention(middle) age(adult) study(normal)
17. C.Pinan	TEACHER	attention(total) money(few) dressing(formal) study(very-much) age(adult)

The results in the training phase show the exemplars retained. These results were obtained using the threshold value of 0.6 and values of 0.4 and 0.1 for the parameters λ and α respectively[1]

The results of this training process are presented as follows.

1. The training case is presented.

   This description shows the name of the new case *nc*, the categories that it represents and the features that represent it.

2. The probability of the exemplars given a new case is ranked.

   For all exemplars e_i that represent the category C_j, the probabilities $P(e_i \mid nc)$ are computed.

[1]Given that the model normally retains the early cases as exemplars, low values were needed in order to obtain a small exemplar based model suitable for illustrative purposes.

If one exemplar has a probability greater than or equal to a threshold value the new training case is classified by the exemplar.

3. The prototypicality measure is presented.

If the new training case was classified by some exemplar, which is named an old exemplar, then the prototypicality measure of both the old exemplar and the new training case are computed. This prototypicality measure determines which exemplar, between the old exemplar and the new case, will represent the category.

Training phase

Exemplar	Category	Features
1. L.Pintos	TEACHER	attention(total) dressing(formal) money(few) age(adult)

Results:

The new case is not classified by an exemplar.

Probability = 0.00

The new training case is added as a new exemplar.

Exemplar	Category	Features
2. L.Pineda	TEACHER	attention(total) money(few) study(very-much) age(adult)

Results:

The new case is classified by the exemplar: L.Pintos with probability 0.85.

The prototypicality measure is:

$$\text{Prototypicality(L.Pintos)} = 0.97 - 0.00 = 0.97$$

$$\text{Prototypicality(L.Pineda)} = 0.97 - 0.00 = 0.97$$

The exemplar L.Pintos is the exemplar selected.

Exemplar	Category	Features
3. W.Philips	TEACHER	age(old) attention(sleeping) money(much) study(very-much)

Results:

The new case is not classified by an exemplar.

$$\text{Probability} = 0.02$$

The new training case is added as a new exemplar.

Exemplar	Category	Features
4. J.Gomez	TEACHER	age(old) attention(sleeping) money(sufficient) study(very-much)

Results:

The new case is classified by the exemplar: W.Philips with probability 0.71.

The prototypicality measure is:

Prototypicality(W.Philips) = 0.93 − 0.01 = 0.92

Prototypicality(J.Gomez) = 0.93 − 0.01 = 0.92

The exemplar W.Philips is the exemplar selected.

Exemplar	Category	Features
5. A.Smith	TEACHER	age(old)
	STUDENT	dressing(formal)
		money(few)
		attention(middle)
		study(very-much)

Results:

The new case is not classified by an exemplar.

Probability = 0.43

The new training case is added as a new exemplar.

Exemplar	Category	Features
6. G.Leon	TEACHER	attention(sleeping)
		money(much)
		study(very-much)

Results:

The new case is classified by the exemplar: W.Philips with probability 0.71.

The prototypicality measure is:

Prototypicality(W.Philips) = 0.96 − 0.02 = 0.94

Prototypicality(G.Leon) = 0.80 − 0.01 = 0.79

The exemplar W.Philips is the exemplar selected.

Exemplar	Category	Features
7. B.Wild	TEACHER	age(old) attention(sleeping) money(sufficient) study(very-much)

Results:

The new case is classified by the exemplar: W.Philips with probability 0.87.

The prototypicality measure is:

Prototypicality(W.Philips) = 0.97 − 0.02 = 0.95

Prototypicality(B.Wild) = 0.97 − 0.02 = 0.95

The exemplar W.Philips is the exemplar selected.

Exemplar	Category	Features
8. P.Ibar	STUDENT	age(adult) attention(middle) dressing(informal) study(normal)

Results:

The new case is not classified by an exemplar.

Probability = 0.05

The new training case is added as a new exemplar.

Exemplar	Category	Features
9. R.Abaco	STUDENT	age(adult) dressing(informal) study(normal)

Results:

The new case is classified by the exemplar: P.Ibar with probability 0.77.

The prototypicality measure is:

Prototypicality(P.Ibar) = 0.99 − 0.00 = 0.99

Prototypicality(R.Abaco) = 0.95 − 0.00 = 0.95

The exemplar P.Ibar is the exemplar selected.

Exemplar	Category	Features
10. P.DeBuen	TEACHER STUDENT	age(old) dressing(formal) attention(middle) study(very-much) has(computer)

Results:

The new case is classified by the exemplar: A.Smith with probability 0.72.

The prototypicality measure is:

Prototypicality(A.Smith) = 0.99 − 0.00 = 0.99

Prototypicality(P.DeBuen) = 0.99 − 0.00 = 0.99

The exemplar A.Smith is the exemplar selected.

Exemplar	Category	Features
11. S.Santana	TEACHER STUDENT	age(adult) dressing(formal) attention(middle) study(very-much) has(computer)

Results:

The new case is classified by the exemplar: A.Smith with probability 0.62.

The prototypicality measure is:

Prototypicality(A.Smith) = 0.98 − 0.00 = 0.98

Prototypicality(S.Santana) = 0.98 − 0.00 = 0.98

The exemplar A.Smith is the exemplar selected.

Exemplar	Category	Features
12. E.Zage	STUDENT	age(adult) attention(middle) dressing(informal)

Results:

The new case is classified by the exemplar: P.Ibar with probability 0.71.

The prototypicality measure is:

Prototypicality(P.Ibar) = 0.99 − 0.00 = 0.99

Prototypicality(E.Zage) = 0.95 − 0.00 = 0.95

The exemplar P.Ibar is the exemplar selected.

Exemplar	Category	Features
13. E.Plaza	TEACHER	age(old)
		attention(sleeping)
		money(much)
		dressing(formal)

Results:

The new case is classified by the exemplar: W.Philips with probability 0.72.

The prototypicality measure is:

Prototypicality(W.Philips) = 0.97 − 0.01 = 0.96

Prototypicality(E.Plaza) = 0.88 − 0.01 = 0.87

The exemplar W.Philips is the exemplar selected.

Exemplar	Category	Features
14. L.Garcia	STUDENT	money(few)
		attention(middle)
		age(adult)
		dressing(informal)
		study(few)

Results:

The new case is classified by the exemplar: P.Ibar with probability 0.87.

The prototypicality measure is:

Prototypicality(P.Ibar) = 0.98 − 0.00 = 0.98

Prototypicality(L.Garcia) = 0.99 − 0.00 = 0.99

The exemplar L.Garcia is the exemplar selected.

Exemplar	Category	Features
15. F.Patlan	TEACHER	age(old) attention(sleeping) dressing(formal) money(much)

Results:

The new case is classified by the exemplar: W.Philips with probability 0.87.

The prototypicality measure is:

$$\text{Prototypicality(W.Philips)} = 0.97 - 0.01 = 0.96$$

$$\text{Prototypicality(F.Patlan)} = 0.94 - 0.01 = 0.93$$

The exemplar W.Philips is the exemplar selected.

Exemplar	Category	Features
16. P.Wolf	STUDENT	money(few) attention(middle) age(adult) study(normal)

Results:

The new case is classified by the exemplar: L.Garcia with probability 0.64.

The prototypicality measure is:

$$\text{Prototypicality(L.Garcia)} = 0.99 - 0.00 = 0.99$$

$$\text{Prototypicality(P.Wolf)} = 0.97 - 0.00 = 0.97$$

The exemplar L.Garcia is the exemplar selected.

Exemplar	Category	Features
17. C.Pinan	TEACHER	attention(total) money(few) dressing(formal) study(very-much) age(adult)

Results:

The new case is classified by the exemplar: L.Pintos with probability 0.94.

The prototypicality measure is:

Prototypicality(L.Pintos) = 0.95 − 0.01 = 0.94

Prototypicality(C.Pinan) = 0.99 − 0.04 = 0.95

The exemplar C.Pinan is the exemplar selected.

At the end of training phase, the probabilistic exemplar based model held three exemplars for the category TEACHER and two exemplars for the STUDENT category. Notice that the exemplar A.Smith is an exemplar that represents a teacher and a student at the same time.

Testing phase

This simple model was tested with the following three test cases. The test cases have the categories that they represent. In order to determine if the classification was well done, these categories are used in the evaluation.

The test case are the following.

Exemplar	Category	Features
1. L.Paz	STUDENT	has(computer) dressing(informal) study(few) money(few) attention(total)
2. J.Perez	TEACHER STUDENT	age(old) dressing(formal) money(few) study(very-much)
3. A.Lara	TEACHER	age(old) study(few) money(much) attention(sleeping)

The first test case was not well classified. The probabilities in the exemplars of the TEACHER category are:

Category: TEACHER

Exemplar: C.Pinan Prob: 0.13

Exemplar: W.Philips Prob: 0.00

Exemplar: A.Smith Prob: 0.00

The probabilities in the exemplars of the STUDENT category are:

Category: STUDENT

Exemplar: L.Garcia Prob: 0.37

Exemplar: A.Smith Prob: 0.00

The second test case is classified by the exemplar A.Smith with probability = 0.85 in STUDENT category. So the test case is well classified since the exemplar A.Smith represents the TEACHER and STUDENT categories that the test case. The probabilities in all the exemplars of the STUDENT category are:

Category: STUDENT	Ranking of probabilities in exemplars
Exemplar: A.Smith	Prob: 0.85
Exemplar: L.Garcia	Prob: 0.02

The third test case is classified by the exemplar W.Philips with probability = 0.93 in TEACHER category. So the test case is well classified since the exemplar W.Philips represents the TEACHER category that the test case. The probabilities in all the exemplars of the THEACHER category are:

Category: TEACHER

Ranking of probabilities in exemplars	
Exemplar: W.Philips	Prob: 0.93
Exemplar: A.Smith	Prob: 0.01
Exemplar: C.Pinan	Prob: 0.00

Appendix B

Results in datasets

This appendix presents the results of the empirical trials for each of the datasets.

Results in 20 trials of votes dataset

No.	Training cases	Testing cases	Exemplars	Classified	Accuracy
1	304	130	6	118	91%
2	304	130	4	122	94%
3	304	130	8	120	92%
4	304	130	5	116	89%
5	305	129	6	117	91%
6	304	130	8	111	85%
7	304	130	4	118	91%
8	305	129	5	108	84%
9	304	130	10	117	90%
10	305	129	6	117	91%
11	304	130	4	123	95%
12	304	130	10	104	80%
13	304	130	10	108	83%
14	305	129	5	115	89%
15	304	130	4	114	88%
16	304	130	6	115	88%
17	304	130	4	114	88%
18	305	129	7	114	88%
19	304	130	5	110	85%
20	304	130	5	108	91%

Results in 20 trials of zoo dataset

No.	Training cases	Testing cases	Exemplars	Classified	Accuracy
1	72	29	9	26	90%
2	72	29	9	27	93%
3	71	30	9	29	97%
4	71	30	8	26	87%
5	71	30	9	27	90%
6	71	30	8	29	97%
7	72	29	7	23	79%
8	71	30	8	28	93%
9	71	30	9	29	97%
10	72	29	9	29	100%
11	72	29	9	27	93%
12	72	29	10	27	93%
13	71	30	10	30	100%
14	72	29	9	28	97%
15	71	30	9	27	90%
16	71	30	8	26	87%
17	71	30	7	25	83%
18	71	30	9	27	90%
19	71	30	10	28	93%
20	72	29	9	26	90%

Results in 20 trials of audiology dataset

No.	Training cases	Testing cases	Exemplars	Classified	Accuracy
1	107	45	55	29	64%
2	107	45	58	19	42%
3	107	45	52	17	38%
4	107	45	49	25	56%
5	108	44	59	21	48%
6	107	45	57	19	42%
7	107	45	62	26	58%
8	108	44	52	24	55%
9	108	44	66	26	59%
10	107	45	60	23	51%
11	108	44	61	23	52%
12	107	45	56	24	53%
13	107	45	57	22	49%
14	108	44	49	19	43%
15	108	44	59	23	52%
16	107	45	58	22	49%
17	107	45	63	24	53%
18	107	45	64	22	49%
19	107	45	57	23	51%
20	107	45	54	20	44%

Bibliography

Aamodt, A. & Plaza, E. (1994), 'Case-based reasoning: foundation issues, methodological variations, and system approaches', *AICom Artificial Intelligence Communications.*

Acorn, T. & Walden, S. (1992), Smart: support management automated reasoning technology for compaq customer service, *in* 'Proc. of the Fourth Annual Conference on Innovative Applications of Artificial Intelligence', San Jose, CA, U.S.A.: AAAI Press.

Aha, D. (1991), Case-based learning algorithms, *in* 'Proc. of the DARPA Case-Based Reasoning Workshop', Washington, DC, U.S.A.: Morgan Kaufmann, pp. 147–158.

Aha, D. & Bankert, R. (1994), Feature selection for case-based classification of cloud types: an empirical comparison, *in* D. Aha, ed., 'Case-Based Reasoning: Papers from the 1994 Workshop (TR WS9401)', Menlo Park, CA, U.S.A.: AAAI Press.

Aha, D. & Chang, L. (1996), Cooperative bayesian and case-based reasoning for solving multiagent planning tasks, Technical Report AIC-96-005, Navy Center for Applied Research in AI Naval Research Laboratory, Washington, DC, U.S.A.

Aha, D. & Goldstone, R. (1992), Concept learning and flexible weighting, *in* 'Proc. of the Fourteenth Annual Conference of the Cognitive Science Society', Bloomington, IN, U.S.A.: Lawrence Erlbaum, pp. 534–539.

Allen, B. (1994), 'Case-based reasoning: business applications', *Communication of the ACM* **37**(3), 40–42.

Althoff, K., Auriol, E., Barletta, R. & Manago, M. (1995), 'A review of industrial case-based reasoning tools', *AI Intelligence.*

Bareiss, R. (1989), *Exemplar-based knowledge acquisition. A unified approach to concept representation, classification, and learning*, Academic Press Inc., Harcourt Brace Jovanovich Publishers, San Diego, CA, U.S.A.

Barletta, R. & Hennessy, D. (1989), Case adaptation in autoclave layout design, *in* 'Proc. of workshop on case-based reasoning (DARPA)', Pensacola Beach, FL, U.S.A.: Morgan Kaufmann.

Biberman, Y. (1995), The role of prototypicality in exemplar-based learning, *in* N. Lavrač & S. Wrobel, eds, 'Proc. of Machine Learning: ECML-95, 8th European Conference on Machine Learning', Heraclion, Crete, Greece, pp. 77–91.

Bolton, N. (1977), *Concept formation*, Pergamon Press, Oxford, England.

Breese, J. & Heckerman, D. (1995), Decision-theoretic case-based reasoning, *in* 'Proc. of the Fifth International Workshop on Artificial Intelligence and Statistics', Ft. Lauderdale, U.S.A., pp. 56–63.

Carbonell, J. (1990), Paradigms for machine learning, *in* J. Carbonell, ed., 'Machine Learning: Paradigms and Methods', Cambridge, MA, U.S.A.: MIT/Elsevier Science, pp. 1–10.

Cassirer, E. (1953), *The philosophical of symbolic forms, Vol. 3, Phenomenology of Knowledge*, New Haven: Yale University Press.

Chang, L. & Harrison, P. (1995), A case-based reasoning testbed for experiments in adaptive memory retrieval and indexing, *in* D. Aha & A. Ram, eds, 'Proc. of the AAAI fall Symposium on Adaptation of Knowledge for Reuse', Menlo Park, CA, U.S.A.: AAAI Press.

Chatfield, C. (1978), *Statistics for technology*, Chapman and Hall, London, England.

Cheeseman, P., Kelly, J., Self, M., Stutza, J., Taylor, W. & Freeman, D. (1990), Autoclass: A bayesian classification system, *in* J. W. Shavlik & T. G. Dietterich, eds, 'Readings in Machine Learning', San Mateo, CA, U.S.A.: Morgan Kaufmann, pp. 296–306.

Cheetham, W. & Graf, J. (1997), Case-based reasoning in color matching, *in* D. Leake & E. Plaza, eds, 'Proc. of Case-Based Reasoning Research and Development, Second International Conference on Case-Based Reasoning, ICCBR97', Providence, RI, U.S.A.: Springer, pp. 1–12.

Cooper, G. (1984), NESTOR: a computer-based medical diagnostic aid that integrate causal and probabilistic knowledge, PhD dissertation, rep. no. stan-cs-84-48, Computer Science Department, Stanford University, U.S.A.

Cooper, G. (1990), 'The computational complexity of probabilistic inference using bayesian networks', *Artificial Intelligence* **42**, 393–405.

Dean, T., Allen, J. & Aloimonos, Y. (1995), *Artificial Intelligence theory and practice*, The Benjamin/Cummings Publishing Company, Inc., Redwood City, CA, U.S.A.

Dombal, F. D., Leaper, D., Horrocks, J., Staniland, J. & McCann, A. (1974), 'Human and computer aided diagnosis of abdominal pain: further report with emphasis on performance', *British Medical Journal* **1**, 376 380.

Domingos, P. (1997), 'Context-sensitive feature selection for lazy learners', *Artificial Intelligence Review* **11**, 227–253.

Dougherty, J., Kohavi, R. & Sahami, M. (1995), Supervised and unsupervised discretization of continuous features, *in* A. Prieditis & S. Russell, eds, 'Proc. of the Twelfth International Conference, Machine Learning', San Francisco, CA, U.S.A.: Morgan Kaufmann.

Fisher, D. (1990), Knowledge acquisition via incremental conceptual clustering, *in* J. W. Shavlik & T. G. Dietterich, eds, 'Readings in Machine Learning', San Mateo, CA, U.S.A.: Morgan Kaufmann, pp. 267–283.

Fisher, D. (1996), 'Iterative optimization and simplification of hierarchical clustering', *Artificial Intelligence Research* **4**, 147–179.

Geiger, D. & Pearl, J. (1988), On the logic of casual models, *in* 'Proc. of the Fourth Workshop on Uncertainty in AI', St. Paul, MI, U.S.A., pp. 136–147.

Geiger, D., Verma, T. & Pearl, J. (1989), Identifying independence in bayesian networks, Technical Report R-116, UCLA Cognitive System Laboratory, U.S.A.

Gennari, J., Langley, P. & Fisher, D. (1990), Models of incremental concept formation, *in* J. Carbonell, ed., 'Machine Learning: Paradigms and Methods', Cambridge, MA, U.S.A.: MIT/Elsevier Science, pp. 11–62.

Golumbic, M. (1980), *Computer Science and applied mathematics, algorithm graph theory and perfect graphs*, Academic Press, Inc., New York, NY, U.S.A.

Gorry, G. & Barnett, G. (1968), 'Experience with a model of sequential diagnosis', *Computer and Biomedical Research* **1**, 490–507.

Heckerman, D. (1995), A tutorial on learning with bayesian networks, Technical Report MSR-TR-95-06, Microsoft Research, Advanced Technology Division, Microsoft Corporation, One Microsoft Way, Redmond, WA, U.S.A.

Hinrichs, T. (1989), Strategies for adaptation and recovery in a design problem solver, *in* 'Proc. of workshop on case-based reasoning (DARPA)', Pensacola Beach, FL, U.S.A.: Morgan Kaufmann.

Horvitz, E., Suermondt, H. & Cooper, G. (1989), Bounded conditioning: flexible inference for decisions under scarce resources, *in* 'Proc. of the Fifth Conference on Uncertainty in Artificial Intelligence (UAI-89)', Windsor, ON, U.S.A.: Morgan Kaufmann, pp. 182–193.

Jabbour, K., Vega-Riveros, J., Landsbergen, D. & Meyer, W. (1988), 'Alfa: automated load forecasting assistant', *IEEE Transaction on Power Apparatus and Systems* **3**(3), 908–914.

Jensen, F. (1996), *An introduction to Bayesian networks*, UCL Press, UCL Press Limited, University College London, Gower Street, London, England.

Kitano, H. & Shimazu, H. (1996), The experience-sharing architecture: a case study in corporate-wide case-based software quality control, *in* D. B. Leake, ed., 'Case-Based Reasoning, Experiences, Lessons, & Future Directions', Cambridge, MA, U.S.A.: AAAI Press/The MIT Press, pp. 235–268.

Kolodner, J. (1993), *Case-based reasoning*, Morgan Kaufmann, Palo Alto, CA, U.S.A.

Kolodner, J. (1996), Making the implicit explicit: clarifying the principles of case-based reasoning, *in* D. B. Leake, ed., 'Case-Based Reasoning, Experiences,

Lessons, & Future Directions', Cambridge, MA, U.S.A.: AAAI Press/The MIT Press, pp. 349–370.

Kolodner, J. & Simpson, R. (1989), 'The mediator: analysis of an early case-based problem solver', *Cognitive Science* **13**(4), 507–549.

Koton, P. (1988), Using experience in learning and problem solving, PhD dissertation, mit/lcs/tr-441, (1989), Laboratory of Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts, U.S.A.

Lauritzen, S. & Spiegelhalter, D. (1988), 'Local computations with probabilities on graphical structures and their application to expert systems', *Journal of the Royal Statistical Society series B* **50**(2), 157–224.

Leake, D. (1996), CBR in context: the present and future, *in* D. B. Leake, ed., 'Case-Based Reasoning, Experiences, Lessons, & Future Directions', Cambridge, Massachusetts, U.S.A.: AAAI Press/The MIT Press, pp. 3–30.

Lindgren, B. (1976), *Statistical Theory*, Macmillan Publishing Co. Inc.

Marir, F. & Watson, I. (1994), 'Case-based reasoning: a review', *The Knowledge Engineering Review.*

Mark, W. (1989), Case-based reasoning for autoclave management, *in* 'Proc. of workshop on case-based reasoning (DARPA)', Pensacola Beach, FL, U.S.A.: Morgan Kaufmann.

Mark, W., Simoudis, E. & Hinkle, D. (1996), Case-based reasoning: expectations and results, *in* D. B. Leake, ed., 'Case-Based Reasoning, Experiences, Lessons, & Future Directions', Cambridge, MA, U.S.A.: AAAI Press/The MIT Press, pp. 269–294.

Medin, D. & Schaffer, M. (1978), 'Contex theory of clasification learning', *Psychological Review* **85**, 207–238.

Merz, C. & Murphy, P. (1996), UCI repository of machine learning databases, http://www.ics.uci.edu/~mlearn/mlrepository.html, Technical report, University of California, Irvine, Dept. of Information and Computer Sciences.

Mohri, T. & Tanaka, H. (1994), An optimal weighting criterion of case indexing for both numeric and symbolic attributes, *in* D. Aha, ed., 'Case-Based Reasoning: Papers from the 1994 Workshop (TR WS9401)', Menlo Park, CA.: AAAI Press.

Myllymäki, P. & Tirri, H. (1994), Massively parallel case-based reasoning with probabilistic similarity metrics, *in* K.-D. A. Stefan Wess & M. M. Ritcher, eds, 'Topics in Case-Based Reasoning', Volume 837, Lecture Notes in Artificial Intelligence. Springer Verlag, pp. 144–154.

Neapolitan, R. (1990), *Probabilistic reasoning in expert systems, theory and algorithms*, A Wiley-Interscience Publication, John Wiley & Sons, Inc., U.S.A.

Pearl, J. (1988), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, Palo Alto, CA, U.S.A.

Pearl, J. (1991), *Probabilistic reasoning in intelligent systems: networks of plausible inference*, Morgan Kaufmann, Palo Alto, CA, U.S.A., (Revised 2nd Edition).

Pearl, J., Geiger, D. & Verma, T. (1990), Conditional independence and its representation, *in* G. Shafer & J. Pearl, eds, 'Readings in Uncertain Reasoning', San Mateo, CA, U.S.A.: Morgan Kaufmann, pp. 55–60.

Peng, Y. & Reggia, J. (1994), 'A probabilistic causal model for diagnostic problem solving-parts i and ii', *IEEE Transaction on Systems, Man, and Cybernetics* **SMC17***(2,3), 395406, 146162.*

Porter, B., Bareiss, R. & Holte, R. (1990), 'Concept learning and heuristic classification in weak-theory domains', *Artificial Intelligence, University of Texas, Austin Texas, U.S.A.* (45), 229–263.

Quinlan, J. (1992), *C4.5: A program for machine learning*, Morgan Kauffman.

Quinlan, J. (1996), 'Induction of decision trees', *Machine Learning* **1**, 81–106.

Riesbeck, C. (1988), An interface for case-based knowledge acquisition, *in* J. kolodner, ed., 'Proc. of the DARPA Case-Based Reasoning Workshop', San Francisco, CA, U.S.A.: Morgan Kaufmann, pp. 312–326.

Riesbeck, C. (1996), What next? the future of case-based reasoning in postmoder ai, *in* D. B. Leake, ed., 'Case-Based Reasoning, Experiences, Lessons, & Future Directions', Cambridge, MA, U.S.A.: AAAI Press/The MIT Press, pp. 371–388.

Riesbeck, C. & Schank, R. (1989), *Inside case-based reasoning*, Lawrence Erlbaum Associates, Hillsdale, NJ, U.S.A.

Rosch, E. & Mervis, C. (1975), 'Family resemblance studies in the internal structure of categories', *Cognitive Psychology* (7), 573-605.

Schank, R. (1982), *Dynamic memory: a theory of reminding and learning in computers and people*, Cambridge University Press, U.S.A.

Smith, E. & Medin, D. (1981), *Categories and concepts*, Cambride: Harvard University Press, U.S.A.

Thomas, H., Foil, R. & Dacus, J. (1997), New technology bliss and pain in a large customer service centre, *in* D. Leake & E. Plaza, eds, 'Proc. of Case-Based Reasoning Research and Development, Second International Conference on Case-Based Reasoning, ICCBR97', Providence, RI, U.S.A.: Springer, pp. 166–177.

Tirri, H., Kontkanen, P. & Myllymäki, P. (1996*a*), A bayesian framework for case-based reasoning, *in* 'Proc. of the 3rd European Workshop on Case-Based Reasoning', Lausanne, Switzerland, pp. 413–427.

Tirri, H., Kontkanen, P. & Myllymäki, P. (1996*b*), Probabilistic instance based learning, *in* L. Saitta, ed., 'Proc. of the Thirteenth International Conference on Machine Learning', San Francisco, CA, U.S.A.: Morgan Kaufmann, pp. 507–515.

Turner, R. (1989), Case-based and schema-based reasoning for problem solving, *in* 'Proc. of workshop on case-based reasoning (DARPA)', Pensacola Beach, FL, U.S.A.: Morgan Kaufmann.

Tversky, A. & Gati, I. (1989), Studies of similarity, *in* K. Holyoak & P. Thagard, eds, 'Cognitive Science, Vol. 13', pp. 79–98.

Veloso, M. (1996), Flexible strategy learning: analogical replay of problem solving episodes, *in* D. B. Leake, ed., 'Case-Based Reasoning, Experiences, Lessons, & Future Directions', Cambridge, MA, U.S.A.: AAAI Press/The MIT Press, pp. 137–149.

Watson, I. (1997), *Applying case-based reasoning: techniques for enterprise systems*, Morgan Kaufmann, Palo Alto, CA, U.S.A.

Wettschereck, D. & Aha, D. (1995), Weighting features, *in* M. Veloso & A. Aamodt, eds, 'Proc. of the First International Conference on Case-Based Reasoning (ICCBR95)', Sesimbra, Portugal: Spring-Verlag, pp. 347–358.

Wettschereck, D., Aha, D. & Mohri, T. (1997), 'A review and empirical evaluation of feature-weighting methods for a class of lazy learning algorithms', *Artificial Intelligence Review* **11**, 273–314.

Wittgenstein, L. (1953), *Philosophical investigations*, Oxford, England: Basil Blackwell.