Development of a Multi-Strategy Theory Revision System

by

Abu Wasif

Department of Computer Science and Engineering

Bangladesh University of Engineering and Technology

December, 2002

Submitted to
Bangladesh University of Engineering and Technology

In partial fulfillment of the requirements for
M.Sc. Engineering (Computer Science and Engineering)

DEVELOPMENT OF A MULTI-STRATEGY THEORY REVISION SYSTEM

A Thesis submitted by

ABU WASIF
Student No. 100005023P

For the partial fulfillment of the degree of
M.Sc. Engineering (Computer Science and Engineering)
Examination held on December 30, 2002

Approved as to style and contents by:

Dr. Chowdhury Mofizur Rahman Professor Department of Computer Science and Engineering B.U.E.T., Dhaka-1000, Bangladesh.	Chairman and Supervisor
Dr. M. Kaykobad Professor Department of Computer Science and Engineering B.U.E.T., Dhaka-1000, Bangladesh.	Member
Dr. Muhammad Masroor Ali Associate Professor Department of Computer Science and Engineering B.U.E.T., Dhaka-1000, Bangladesh.	Member
Dr. Md. Abul Kashem Mia Associate Professor and Head Department of Computer Science and Engineering B.U.E.T., Dhaka-1000, Bangladesh.	Member (Ex-Officio)
Dr. Md. Kamrul Hasan Associate Professor Department of Electrical and Electronic Engineering B.U.E.T., Dhaka-1000, Bangladesh.	Member (External)

Certificate

This is to certify that the work presented in this thesis paper is the outcome of the investigation carried out by the candidate under the supervision of Dr. Chowdhury Mofizur Rahman in the Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology, Dhaka. It is also declared that neither of this thesis nor any part thereof has been submitted or is being concurrently submitted anywhere else for the award of any degree or diploma.

CM Rahman

Signature of the Supervisor
(Dr. Chowdhury Mofizur Rahman)

Wasif

Signature of the Author
(Abu Wasif)

Contents

List of figures

Acknowledgement

First and foremost, I would like to acknowledge my gratitude to Dr. Chowdhury Mofizur Rahman, Professor, Department of Computer Science and Engineering, Bangladesh University of Engineering and Technology. His Constant supervision, scholarly guidance, valuable advice and encouragement have been a great impetus to this thesis work. His vast experience and in-depth knowledge in Artificial Intelligence have helped significantly to achieve a smooth completion of this thesis work.

I would like to thank the members of the graduate committee; Dr. Md. Abul Kashem Mia, Associate Professor and Head of the Department of Computer Science and Engineering, Dr. M. Kaykobad, Professor, Department of Computer Science and Engineering, Dr. Muhammad Masroor Ali, Associate Professor, Department of Computer Science and Engineering, Dr. Md. Kamrul Hasan, Associate Professor, Department of Electrical and Electronic Engineering, for their valuable suggestions.

I would like to acknowledge the all-out co-operation and services rendered by the faculty members and staffs of the Department of Computer Science and Engineering. Finally, to all who contributed in any way, I would like to express my deep appreciation and sincere feelings.

Abstract

This thesis proposes and implements a new Theory Revision System. The Theory Revision problem is defined as the problem of how best to go about revising a knowledge base on the basis of a collection of examples, some of which expose inaccuracies in the original knowledge base. This problem entreats a thorough investigation of the following machine learning field of study: Combining Inductive and Analytical learning. The problem of theory revision has been studied for quite some time and various systems have been proposed. On one hand, there are successful theory revision systems like EITHER and PTR, which combine Inductive and Analytical Learning. On the other hand, there are mention-worthy systems like KBANN, TANGENTPROP, and EBNN that use imperfect domain theories together with given training set of data. The new system is built by incorporating Version Space-based Incremental Probabilistic Evidence Combination method and Integrated Analytical/Empirical method. The proposed system is constructed to maximize preservation of already gained meaningful information. To our knowledge, Version Space-based approach has not been applied for theory revision problem as yet. Experimental results show that the performance of the new system is comparable with other fairly successful systems.

CHAPTER ONE

INTRODUCTION

We provide a brief summary of the problem of the current thesis work in this chapter. We present a justification of theory revision systems and illustrate their position in the spectrum of learning tasks. We conclude the chapter by describing the objective of the thesis and thesis organization.

1.1 The Problem of Theory Revision

The construction of the underlying knowledge base is considered as one of the most difficult problems in the development of expert systems. Research in machine learning attempts to solve the *knowledge acquisition problem* by developing systems that automatically acquire the requisite knowledge from experience [9, 16].

Normal knowledge acquisition can be divided into two phases: an initial phase in which a knowledge engineer extracts a rough set of rules from an expert, and *knowledge base revision*, in which the initial knowledge base is refined to produce a high-performance system [3]. The initial knowledge base is acquired as whole rules, or sets of rules, that are used to represent various concepts in the domain. In contrast, during knowledge base revision, components of the existing rules are modified, in addition to adding and deleting rules, in an effort to improve its ability to reach correct conclusions in its domain.

One of the main problems in building expert systems is that the initial models obtained from Human expert tend to be only approximately correct. They generally make a good first approximation to the real world, but they typically contain inaccuracies that are exposed, when a fact is asserted that does not agree with empirical observation. The *theory revision problem* is the problem of how best to go about revising a knowledge base on the basis of a collection of examples, some of which expose inaccuracies in the original knowledge base. Formally, the theory revision problem is stated as,

Given:

An imperfect domain theory for a set of categories and a set of classified examples each described by a set of observable features.

Find:

The <u>best revision</u> of the domain theory that correctly classifies all of the examples.

Different theory revision systems define the best revision in different ways. The best revision is defined by EITHER [14] as the revision that results in minimum syntactic change. On the other hand, PTR[7] defines the best revision as the revision that outputs the most probable theory with respect to initial prior probabilities and the training set.

1.2 Justifications of Theory Revision System

Theory revision systems are required to revise initial rough models elicited from human experts. Even if the initial model happens to be a refined one, a theory revision system helps it to remain updated with the dynamic real-world situation.

There are two advantages of a theory revision approach to knowledge acquisition as opposed to a purely empirical learning approach.

- By starting with an approximately correct theory, a revision system should be able to achieve high performance with significantly fewer training examples. So, the theory revision approach has a distinct advantage in domains in which training data is scarce or in which a rough theory is available.

- Theory revision systems result in a structured knowledge base that maintains the explanatory structure of the original theory. So, it is more suitable for supplying meaningful explanations for its conclusions, an important aspect of the usefulness of an expert system.

1.3 A Theory Revision System Viewpoint of the Two Learning Paradigms

The two paradigms for machine learning are *inductive learning* and *analytical learning*. These two learning paradigms are based on fundamentally different justifications for learned hypotheses and they have complementary advantages and disadvantages. Combining them offers the possibility of more powerful learning methods.

- In inductive learning, the learner is given a hypothesis space H from which it must select an output hypothesis, and a set of training examples $D = \{\langle x_1, f(x_1)\rangle, \langle x_2, f(x_2)\rangle, \langle x_3, f(x_3)\rangle, \ldots, \langle x_n, f(x_n)\rangle\}$, where $f(x_i)$ is the target value for the instance x_i. The desired output of the learner is a hypothesis h from H that is consistent with these training examples. Learning Systems, like ID3 [16], are representative of inductive learning system.

- In analytical learning, the input to the learner includes the same hypothesis space H and training examples D as for inductive learning. In addition, the learner is provided an additional input: a *domain theory* B consisting of background knowledge that can be used to explain observed training examples. The desired output of the learner is a hypothesis h from H that is consistent with both the training examples D and the domain theory B. Learning Systems, like Prolog-EBG [6], are representative of analytical learning system.

Hypotheses generated from purely analytical learning method carry a *logical* justification. On the other hand, hypotheses generated from purely inductive learning method carry a *statistical* justification. Logical justifications are only as compelling as the assumptions, or prior knowledge, on which they are built. They are suspect or powerless when prior knowledge is incorrect or unavailable. Statistical justifications are only as compelling as the data and statistical assumptions on which they rest. They are suspect or powerless when assumptions about the underlying distributions cannot be trusted or when the data is scarce.

In short, the two approaches work well for different types of problems and by combining them, we can hope to devise a more general learning approach that covers a more broad range of learning tasks.

Figure 1.1: A spectrum of learning tasks

Theory Revision Problem is characterized by imperfect prior knowledge and scarce training data. It can be viewed as

- An approach of combining inductive (statistical inference) and analytical (deductive inference) learning.
- A way of examining the influence of prior knowledge on concept acquisition.

1.4 The Theory Revision Problem: A Practical Example

In this section, we provide a practical problem and a pictorial representation of the theory revision problem.

Training set data format: [X, F(X)]; X = [x1,x2,x3,x4,x5,x6]

Here X is an instance, described by six observable features x1,x2, ..., x6 and F is the target function value for instance X.

Given, Classification theory C and Training set N.

C: $(x2 \wedge x3) \vee x4$

N:

X	F	
011110	1	
110000	0	
100101	0	← False +ve
111011	1	

A positive member of the training set which is incorrectly classified as a negative member by the imperfect domain theory is called a failing positive or a false negative example. Similarly, a negative member of the training set which is incorrectly classified as a positive member by the imperfect domain theory is called a failing negative or a false positive example.

There can be a number of possible revised theories, such as,

C1: $(x2 \wedge x3) \vee (x4 \wedge x1)$

C2: $(x2 \wedge x3) \vee (x4 \wedge x6)$

C3: $(x2 \wedge x3) \vee (x4 \wedge (x2 \vee x3))$

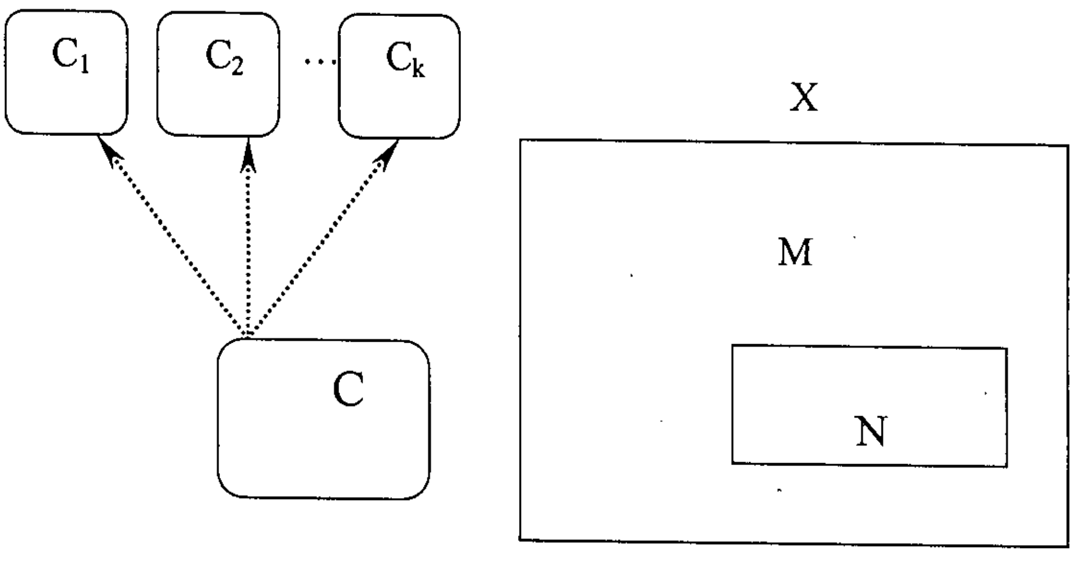

Figure 1.2: The Theory Revision Problem: The choice of C_i

A pictorial representation of the problem of theory revision is shown in figure 1.2. In this figure, X is the total instance space, of which N is a subset that is given as training set for theory revision. C_1, C_2,...C_k are revised theory that are consistent with N. A theory revision system is to choose the best candidate C_i from the set of possible revised theories that are consistent with given training set N.

The problem of theory revision is to find a revised theory C_i so that

- C_i is consistent with the given training set N
- C_i is the best revision with respect to C and N

1.5 The objective of the Thesis

The objective of the current thesis work is to propose and implement a new theory revision system with a new objective of best revision. We define the best revision in the following way: '*The best revision maximizes preservation of already gained meaningful information*'. We construct a new system by incorporating Version Space-based Incremental Probabilistic Evidence Combination method and Integrated Analytical/Empirical method. To our knowledge, Version Space-based approach has not been applied for theory revision problem as yet.

1.6 The Organization of the Thesis

The thesis is organized in the following way:

Chapter one provides a brief summary of the problem of current thesis work along with an overview.

Chapter two describes about the best revision, the existing theory revision systems and relevant systems from the field of study: Combining Inductive and Analytical learning. It provides a brief survey of the learning systems that use imperfect domain theories together with given training set of data.

Chapter three describes the proposed new theory revision system. First, some basic concepts and notations are presented. Next, we formally define the objective of the proposed new theory revision system. Then, we describe the implementation steps of the new theory revision system algorithm.

Chapter four presents the experimental results and comparisons with existing systems.

Chapter five provides concluding remarks and future research directions.

CHAPTER TWO

EXISTING SYSTEMS

In this chapter, we describe about the best revision, the existing theory revision systems and relevant systems from the field of study: Combining Inductive and Analytical learning. It provides a brief survey of the learning systems that use imperfect domain theories together with given training set of data.

2.1 Definition of Best Revision

The theory revision systems attempt to find a (correct and consistent) revised theory as faithful as possible with respect to the original theory. Different theory revision systems define the best revision in different ways and obtain that defined best revision.

Existing theory revision systems with fairly successful performance include EITHER [14] and PTR [7].

- EITHER defines the best revision as the revision that results in minimum syntactic change.
- PTR, first assigns prior probabilities to elements of existing theory, then finds the most probable theory by updating probabilities depending on the given examples and revising the elements with lower probabilities. It defines the best revision as the revision that outputs the most probable theory with respect to initial prior probabilities and the training set.

2.2 Successful Theory Revision Systems: EITHER and PTR

EITHER and PTR are two successful theory revision systems. They are also representative of learning systems that combine analytical and inductive learning.

2.2.1 EITHER

EITHER (Explanation-based and Inductive THeory Extension and Revision) is a modular theory revision system that attempts to integrate analytical methods, i.e., abduction and deduction and empirical methods, i.e., induction. Its modularity stems from the independent subsystems for carrying out deduction, abduction and induction.

2.2.1.1 Induction, Deduction and Abduction

Induction, deduction and abduction are three basic inference processes.

Induction: In supervised learning, the learner is given the correct value of the function for particular input and an example is a pair ⟨x, f(x)⟩, where x is the input and f(x) is the output of the function applied to x. The induction process returns a function h that approximates f, given a collection of examples of f.

Deduction: Deduction is a process that implements the entailment relation. For examples, if there is rule 'if there is smoke, there is fire' and given the fact 'Smoke is true', by deduction, we have, 'Fire is true'.

Abduction: Abduction is a process of reasoning to an explanation according to a known rule. For examples, if we have 'if the morning is hot, humid and cloudy, then it rains in the afternoon', then by abduction from a rainy afternoon, we reason that the morning was hot, humid and cloudy.

2.2.1.2 EITHER Architecture

EITHER architecture is illustrated in figure 2.1. EITHER have abductive, deductive and inductive systems. Each of these reasoning components makes important contributions to the overall goal of the system. EITHER first uses abduction and deduction. Only if these methods fail, EITHER resorts to inductive component to learn new rules.

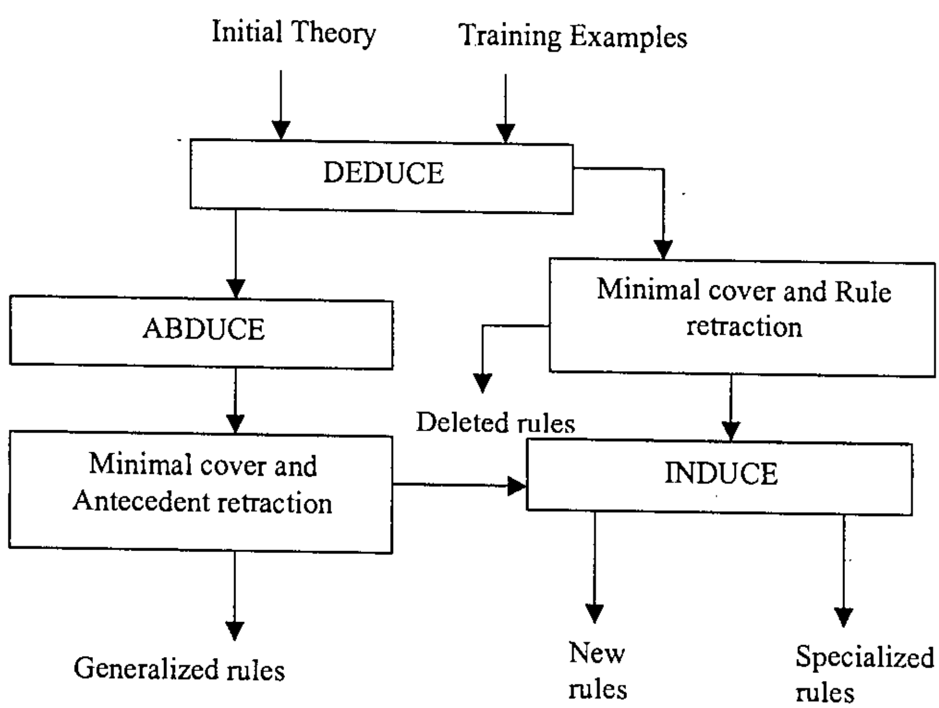

Figure 2.1: EITHER Architecture

Horn-clause deduction is the basic inference engine used to classify examples. EITHER initially uses deduction to identify failing positives and negatives among the training examples. Deduction is also used to assess proposed changes to the theory as part of the generalization and specialization process.

EITHER uses abduction to initially find the incorrect part of an overly specific theory. Abduction identifies sets of assumptions, which would allow a failing positive to become provable. These assumptions identify conflicting antecedents that, if deleted, would properly generalize the theory and correct the failing positive.

EITHER uses its inductive subsystem to learn new rules or to determine which additional antecedents to add to an existing rule. EITHER uses a version of ID3 [16] as its inductive component. An appropriate subset is passed as input to ID3 and the outcome is a decision tree. The output decision tree is then translated to equivalent Horn-Clause rules.

The remaining components of the EITHER constitute generalization and specialization control algorithms, which identify and specify the types of corrections to be made to the theory.

2.2.2 PTR

PTR (Probabilistic Theory Revision) is a theory revision system, which uses a set of training examples to incrementally adjust probabilities, associated with the elements of an imperfect domain theory in order to find the most-probable set of revisions to the theory, which will make it consistent with the provided training set.

PTR is inspired by EITHER [14] and KBANN [18]. Like KBANN, PTR incrementally adjust weights associated with domain theory elements. Like EITHER, all stages of PTR are carried out within the propositional logic framework and the obtained theories are not probabilistic.

PTR requires each element of the given classifier theory to be assigned some a priori probability that it is not flawed and does not require revision. These probabilities might be assigned by an expert or simply chosen by default.

PTR produces a revised theory in the following way. First, PTR translates the given theory into some NAND equations. PTR represents a domain theory as a weighted digraph. The nodes correspond to clause heads or literals representing observable features. The edges correspond to NAND equations of the transformed domain theory. Each edge e is associated with a probability p(e) which corresponds to the expert's confidence that e need not be revised.

Next, faulty edges are determined by processing one training example at a time. The flow of proof through the edges of the digraph is measured. The more an edge contributes to the correct classification of an example, the more its weight is increased, i.e., p(e) is updated to be incremented. The more an edge contributes to the misclassification of an example, the more its weight is decreased, i.e., p(e) is updated to be decremented. If the weight of an edge falls below a predefined revision threshold, then it is marked for revision.

In the next step, a decision is made if a deletion of the edge that has been marked for revision is possible by finding out the set of training examples dependent on the existence of this edge for correct classification. If deletion is found to be impossible, a subtree is computed by an inductive component ID3 [16] and this subtree is added to the child node to which the edge under revision leads.

The steps corresponding to finding faulty edges and revising marked edges are performed repeatedly, by processing training examples one at a time in random order. PTR is proved to converge to a theory that correctly classifies all examples, under certain conditions.

2.3 Other Relevant Systems

Learning Systems like KBANN [18], TANGENTPROP [19] and EBNN [22] combine inductive and analytical learning. They are representative of the Inductive-Analytical approaches to learning. However, these systems are not built for the purpose of producing a revised theory.

Generally, the systems are neural network based systems. From the viewpoint of combining analytical and inductive learning, they are very much successful, but their dependence on the neural network architecture makes them inferior candidates for theory revision systems. This stems from the following general considerations about neural network: Neural networks are distributed representations. Units in neural networks do not typically represent specific propositions. Even if they did, the calculations carried by the network do not treat propositions in any semantically meaningful way. In practical terms, this means that humans can neither construct nor understand neural network representations.

These systems use domain theories to influence the hypothesis space search. More specifically, they use imperfect domain theories either to create the initial hypothesis in the search, or alter the objective of the search.

2.3.1 KBANN

KBANN (Knowledge-Based Artificial Neural Network) is an example of a system that uses the domain theory to initialize the hypothesis. It uses prior knowledge to initialize the hypothesis to perfectly fit the domain theory, then inductively refine this initial hypothesis as needed to fit the training data.

The input to KBANN is a set of training examples and a domain theory consisting of nonrecursive, propositional Horn clauses. It produces a neural network that fits the training examples, biased by the domain theory.

The two stages of the KBANN algorithm are first to create an artificial neural network that perfectly fits the domain theory and second to use the BackPropagation algorithm to refine this initial network to fit the training examples. Although, the initial theory fails to classify all training examples, it forms a useful approximation to the target concept. KBANN uses the domain theory and training examples together to learn the target concept more accurately than it could done from either alone.

In the first stage of the KBANN algorithm, an initial network is constructed that is consistent with the domain theory. In general, the network is constructed by creating a sigmoid threshold unit for each Horn clause in the domain theory. KBANN follows the convention that a sigmoid output value greater than 0.5 is interpreted as *True* and a value below 0.5 as *False*. Each unit is therefore constructed so that its output will be greater than 0.5 just in those cases where the corresponding Horn clause applies. For each antecedent to the Horn clause, an input is created to the corresponding sigmoid unit. The weights of the sigmoid unit are then set so that it computes the logical AND of its inputs. In particular, for each input corresponding to a non-negated antecedent, the weight is set to some positive constant W. For each input corresponding to a negated antecedent, the weight is set to –W. The threshold weight of the unit, w_0 is then set to –(n-.5)W, where n is the number of non-negated antecedents. Each sigmoid unit input is connected to the appropriate network input or to the output of another sigmoid unit, to mirror the graph of dependencies among the corresponding attributes in the domain theory.

The second stage of KBANN uses the training examples and the BackPropagation algorithm to refine the initial network weights. Since the domain theory and training data are inconsistent, this step alters the initial network weights. The result is a network biased by the original domain theory, whose weights are refined inductively based on training data.

There is significant difference between the hypothesis search of KBANN and BackPropagation. KBANN uses the prior knowledge to initialize the weights of the neural network. It is likely to converge to a hypothesis that generalizes beyond the data in a way that is more similar to domain theory predictions. KBANN uses a

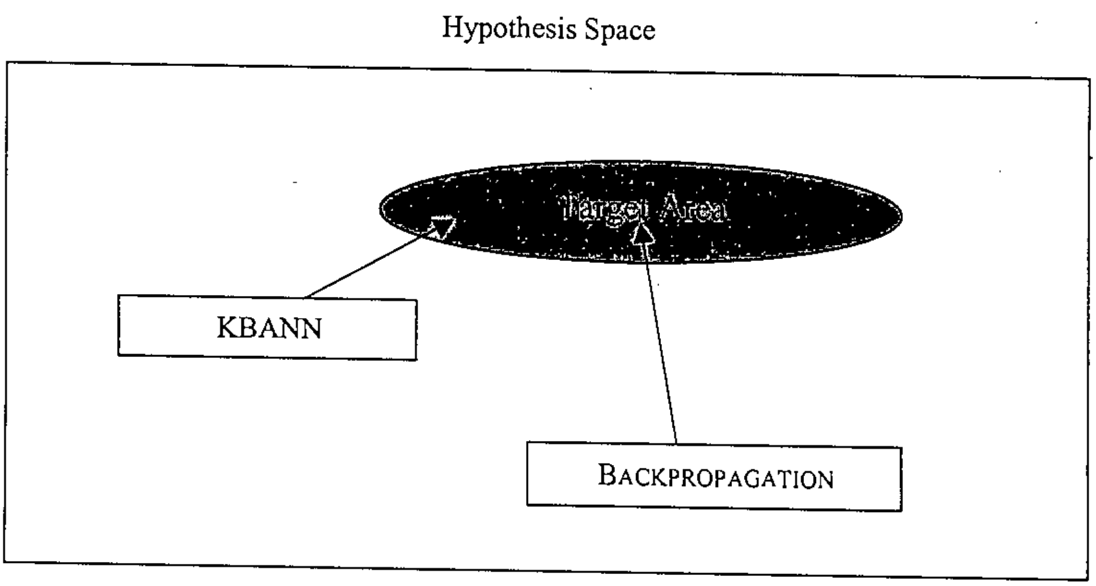

Figure 2.2 : KBANN Hypothesis space search

domain-specific theory to bias generalization, whereas BACKPROPAGATION uses a domain independent syntactic bias toward small weight values.

KBANN illustrates the initialize-the-hypothesis approach to combining analytical and inductive learning. Other examples of this approach include Fu [2], Gallant [4], Bradshaw [1], Lacher [8].

2.3.2 TANGENTPROP:

TANGENTPROP accommodates prior knowledge expressed as derivatives of the target function with respect to transformations of its inputs. In some domains, such as image processing, this is a natural way to express prior knowledge. TANGENTPROP uses this knowledge by altering the objective function minimized by gradient descent search through the space of possible hypotheses.

Let us consider a learning task involving an instance space X and target function f. Generally, each training example consists of a pair $\langle x_i, f(x_i)\rangle$ that describes some instance x_i and its training value $f(x_i)$. The TangentProp algorithm assumes various training derivatives of the target function are also provided. For example, if each instance x_i is described by a single real value, then each training example may be of the form $\left\langle x_i, f(x_i), \frac{\partial f(x)}{\partial x}\Big|_{x_i}\right\rangle$

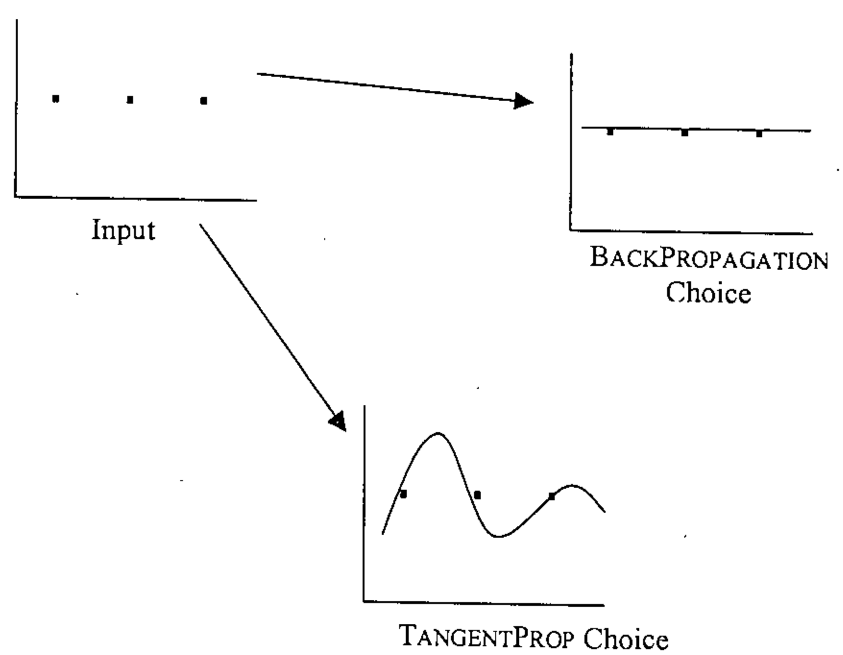

Figure 2.3: Fitting values and derivatives with TANGENTPROP

Figure 2.3 illustrates the benefits of providing training derivatives as well as training values during learning. BACKPROPAGATION can be expected to hypothesize a smooth function. However, by learning from both the training values and the training derivatives, TANGENTPROP has a better chance to correctly generalize from the sparse data.

TANGENTPROP incorporates prior knowledge to influence the hypothesis search by altering the objective function to be minimized by gradient descent. This corresponds to altering the goal of the hypothesis search. Alternatively, as it tries to fit the training derivatives of the target function, it can be viewed as a way of synthesizing additional training data in the neighborhood of the observed training data.

2.3.3 EBNN

EBNN (**E**xplanation-**B**ased **N**eural **N**etwork Learning) uses the domain theory to alter the objective in searching the hypothesis space of possible weights for an artificial neural network. It uses a domain theory consisting of previously learned neural networks to perform a neural network analogous to symbolic explanation-based learning.

As in symbolic explanation-based learning, the domain theory is used to explain individual examples, yielding information about the relevance of different example features. With this neural network representation, however, information about the relevance is expressed in the form of derivatives of the target function value with respect to instance features. The network hypothesis is trained using a variant of the TANGENTPROP algorithm, in which the error to be minimized includes both the error in network output values and the error in network derivatives obtained from explanations.

CHAPTER THREE

PROPOSED NEW THEORY REVISION SYSTEM

In this chapter, we describe the proposed new theory revision system. First, some basic concepts and notations are presented. Next, we formally define the objective of the proposed new theory revision system. Then, we describe the implementation steps of the new theory revision system. We conclude the chapter by presenting the algorithm of the new system.

3.1 Preliminaries

In this section, we present some basic terms and concepts. Subsection 3.1.1 and 3.1.2 present concepts of instance space and hypothesis space. Subsection 3.1.3 and 3.1.4 present concepts of inductive bias and version space.

3.1.1 A Concept Learning Task

Much of learning involves acquiring general concepts from specific training examples. Each such concept can be viewed as describing some subset of objects or events defined over a larger set. A practical example, we may consider acquisition of concept of a bird from a subset of animals that constitute the taxonomic category Birds. In this way, a concept-learning task is considered as the acquisition of definition of a general category given a sample of positive and negative training examples of a category.

Let us consider the task of learning the target concept "Days on which a person M enjoys his favorite sport". Table 3.1 describes a set of example days, each represented by a set of attributes. The attribute *EnjoySport* indicates whether or not person M enjoys his favorite sport on this particular day. The task is to learn to predict the value of *EnjoySport* for an arbitrary day, based on the values of its other attributes.

Let us consider a simple hypothesis representation scheme so that each hypothesis consists of a conjunction of constraints on the instance attributes. In particular, let each hypothesis be a vector of six constraints, specifying the values of the attributes *Sky, AirTemp, Humidity, Wind, Water* and *Forecast*. For each attribute, the hypothesis will either

- Indicate by a "?" that any value is acceptable for this attribute,
- Specify a single required value (e.g. *Rainy*) for the attribute, or
- Indicate by a "∅" that no value is acceptable.

Example	*Sky*	*AirTemp*	*Humidity*	*Wind*	*Water*	*Forecast*	**EnjoySport**
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Table 3.1: Positive and negative examples for the target concept *EnjoySport*

If some instance x satisfies all the constraints of hypothesis h, then h classifies x as a positive example (h(x)=1). To illustrate, the hypothesis that M enjoys his favorite sport only on cold days with high humidity (independent of the values of the other attributes) is represented by the expression

$$\langle ?, \text{Cold}, \text{High}, ?, ?, ? \rangle$$

The most general hypothesis that 'everyday is a positive example' is represented by

$$\langle ?, ?, ?, ?, ?, ? \rangle$$

and the most specific possible hypothesis that 'no day is a positive example' is represented by

$$\langle \varnothing, \varnothing, \varnothing, \varnothing, \varnothing \rangle$$

3.1.2 Notation for Concept Learning Problems

In general, any concept learning task can be described by the set of instances over which the target function is defined, the target function, the set of candidate hypotheses considered by the learner, and the set of available training examples.

Instance Space: The Instance space is constituted by the set of items over which the concept is defined. It is generally denoted by X. In the current example, X consists of the members of the set of all possible days, each represented by the attributes *Sky, AirTemp, Humidity, Wind, Water* and *Forecast.*

Target Concept: The concept or function to be learned is called the target concept, which is generally denoted by c. c can be any Boolean-valued function defined over the instances X; that is, $c:X \rightarrow \{0, 1\}$. In the current example, the target concept corresponds to the value of the attribute EnjoySport (i.e., $c(x) = 1$ if EnjoySport = Yes, and $c(x) = 0$ if EnjoySport = No).

Hypothesis Space: Given a set of training examples of the target concept c, the problem faced by the learner is to hypothesize, or estimate c. The hypothesis space is constituted by the set of all possible hypotheses that the learner may consider regarding the identity of the target concept. It is generally denoted by H and it is determined by the human designer's choice of hypothesis representation. In general, each hypothesis h in hypothesis space H represents a Boolean-valued function defined over X; that is, h: $X \rightarrow \{0, 1\}$. The goal of the learner is to find a hypothesis h such that $h(x) = c(x)$ for all x in X.

3.1.3 Inductive Bias and Version Space

Concept learning can be viewed as the task of searching through a large space of hypotheses implicitly defined by the hypothesis representation. The goal of this search performed by the learner is to find the hypothesis that best fits the training examples.

Inductive Bias: Let us consider a concept learning algorithm L for the set of instances X. Let c be an arbitrary concept defined over X, and let $D_c = \{\langle x, c(x)\rangle\}$ be an arbitrary set of training examples of c. Let $L(x_i, D_c)$ denote the classification assigned to the instance x_i by L after training on the data D_c. The *inductive bias* of L is any minimal set of assertions B such that for any target concept c and corresponding training examples D_c, the following holds,

$$(\forall x_i \in X)[(B \wedge D_c \wedge x_i) \mapsto L(x_i, D_c)]$$

Inductive Bias can be alternatively defined as basis for choosing one generalization over another. Biases improve the predictive power of the induced theory [10] and make induction tractable [11].

Version Space: The version space $VS_{H,D}$, with respect to hypothesis space H and training examples D, is the subset of hypotheses from H consistent with the training examples in D,

$$VS_{H,D} \equiv \{h \in H \mid Consistent(h, D)\}$$

Version space is characterized by two boundary sets S and G. The specific boundary S is the set of maximally specific members of H consistent with D, and the general boundary G is the set of maximally general members of H consistent with D. Recent researches on boundary set characterization and representation enable efficient instance retraction [20, 21].

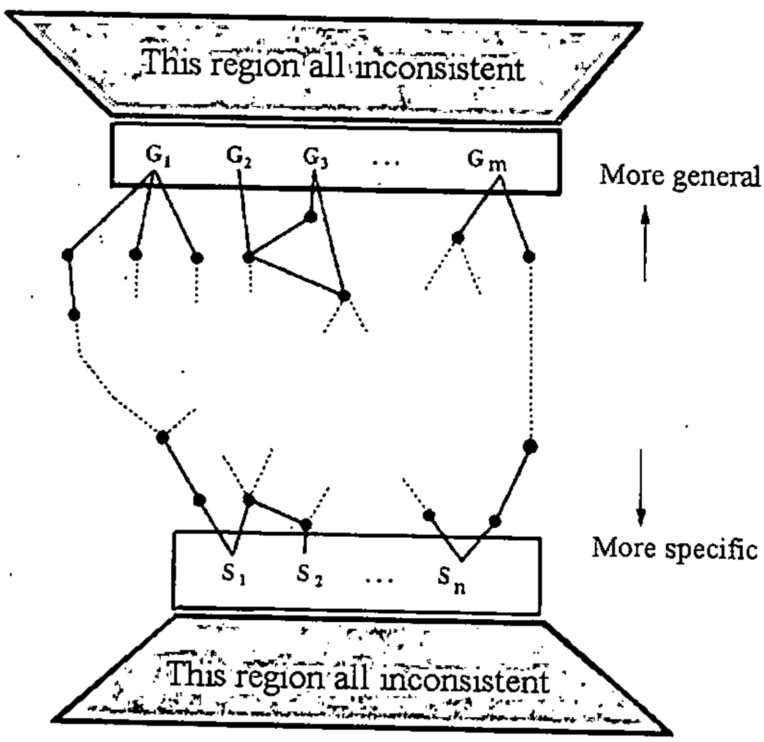

Figure 3.1: Version Space and two boundary sets S and G

Version spaces are viewed as

- Provider of a good deal of insight into the logical structure of hypothesis space.
- Non-practical in most real-world learning because of assumption of consistent training data and being noise-sensitive.

3.1.4 Version Space Update

As concept learning is viewed as a search problem, it is natural that learning algorithms employ different strategies for searching the hypothesis space. Many algorithms for concept learning, organize the search by relying on a very useful structure that exists for any concept-learning problem: a general-to-specific ordering of hypothesis. A hypothesis h_i is more general than h_j, if and only if all positive examples satisfied by h_j are also satisfied by h_i and there is at least one positive example satisfied by h_i that is not satisfied by h_j.

To find all sets of hypotheses consistent with a given set of training examples, it is possible to carry out a version space computation process in the following way. First, we initialize S boundary set to be the most specific or least general hypothesis, i.e. S0 is ⟨∅, ∅, ∅, ∅, ∅⟩. The G boundary set is initialized to be the most general hypothesis, i.e. G0 is ⟨?, ?, ?, ?, ?, ?⟩. Next, we process one training example at a time and update the boundary sets S and G so that any hypothesis found to be inconsistent with the training example under consideration is eliminated from version space. In this way, we process all training examples one at a time from the provided training set.

For example, if we consider the four examples from table 3.1, then we have the following computation of the version space:

Step0: S0: ⟨∅, ∅, ∅, ∅, ∅⟩ and G0: ⟨?, ?, ?, ?, ?, ?⟩

Step1: S1: ⟨Sunny, Warm, Normal, Strong, Warm, Same⟩ and

G1: ⟨?, ?, ?, ?, ?, ?⟩

Step2: S2: ⟨ Sunny, Warm, ?, Strong, Warm, Same ⟩ and

G2: ⟨?, ?, ?, ?, ?, ?⟩

Step3: S3: ⟨ Sunny, Warm, ?, Strong, Warm, Same ⟩ and

G3: {⟨Sunny, ?, ?, ?, ?, ?⟩, ⟨?, Warm, ?, ?, ?, ?⟩, ⟨?, ?, ?, ?, ?, Same⟩}

Step4: S4: ⟨ Sunny, Warm, ?, Strong, ?, ? ⟩ and

G4: {⟨Sunny, ?, ?, ?, ?, ?⟩, ⟨?, Warm, ?, ?, ?, ?⟩ }

At step i, the i–th example from the table 3.1 is processed, and S and G sets are updated. Finally, the version space is characterized by the final boundary sets

S: ⟨ Sunny, Warm, ?, Strong, ?, ? ⟩

and G: {⟨Sunny, ?, ?, ?, ?, ?⟩, ⟨?, Warm, ?, ?, ?, ?⟩ }

3.2 New Theory Revision System Approach

We propose and implement a new theory revision system that incorporates Version Space-based Incremental Probabilistic Evidence Combination method and Integrated Analytical/Empirical method. The Incremental Probabilistic Evidence Combination method is used for guidance in the choice of the best revised theory. The Integrated Analytical/Empirical method is used for minimization of syntactic changes.

3.2.1 New Theory Revision System Description

The Goal: To develop a *new* Theory Revision system.

Given: Classifier theory C and a set of N training examples.

Assumptions: In this regard, wc make the following assumptions,

- The theory revision problem is considered as a case of knowledge base evolution.
- The target concept is contained in the hypothesis space.

The significance of the first assumption is that the given training set N is considered to be unseen until now. If we consider, the total instance space X and M = X – N, then the first assumption states that during the knowledge base construction phase, a subset A of M is used.

The significance of the second assumption is that the chosen representation formalism is considered powerful enough to completely express the target concept.

Strategy: We define the best revision in the following statement:

The best revision maximizes preservation of already gained meaningful information.

Let A be the already seen portion of the instance space X. The significance of this strategy is to produce a revised theory that is consistent with A and N, by determining A. In other words, the focus is to preserve the meaningful information gained during construction phase.

Steps: The steps of the new theory revision system is as follows,

Given C and N,

1. Compute a classifier C_n based on given classifier theory C and set of training data N.

2. Compute A = Intersection (C_n, C) – N.

3. Compute a minimum syntactic revision of C so that the given set N is covered correctly and coverage of the computed set A is unchanged.

The implementation of these three steps is described in the following section. The implementation of the step 1 is described in section 3.3 and the step 3 is described in subsection 3.4. The implementation of step 2 does not require elaboration. However, instead of a classifier C_n, we compute a VS (Version Space), characterized by boundary sets S and G in step 1. The corresponding change in step 2 is provided in section 3.5.

3.3 Implementation of step 1

Step1 requires that a classifier C_n based on given classifier theory C and set of training data N, is to be computed. By *incremental probabilistic evidence combination* method, we compute a VS (Version Space), characterized by S and G.

3.3.1 VS computation by Probabilistic Evidence Combination Method

We first introduce the use of a noise model that provides the conditional probabilities to be used by the probabilistic evidence combination method.

The motivation for this lies in the observation that the inability of the given hypothesis C to be consistent with given training set N indicates faulty bias. This fault can be characterized by noise. From practical viewpoint, the faulty bias is responsible for C to be inconsistent with members of N. However, from viewpoint of C, the bias is not noisy or faulty but the members of N seem to be corrupted by noise.

Classifier learning from noisy data can be viewed as problem of reasoning under uncertainty and knowledge of the noise process can be applied to compute a posteriori probabilities over the hypothesis space [13].

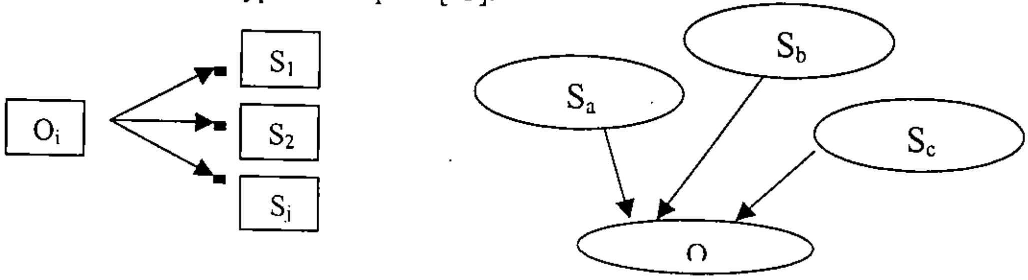

Figure 3.2 : Noise model (Observed data from N and Supposed data)

The computation is done in the following way. Let an individual member O_i (observed data from N) be corrupted and the supposed data for it be $S_1, S_2, \ldots, S_j$. The initial probabilities $P(O_i|S_j)$ is provided by the selected noise model appropriate for the problem domain. Let us consider the case that a corruption up to k-bits is possible due to noise. We initially determine a set M that consists of supposed data for each members of the provided training set of examples. Next, we compute the *maximum a posteriori* hypothesis with respect to M.

The calculation of *maximum a posteriori* (MAP) hypothesis is found from the following derivation:

Maximize,

$$
\begin{aligned}
& P(Hi|O) \\
&= \{P(O|Hi)\, P(Hi)\} / \{P(O)\} \\
&\approx P(O|Hi) \\
&= \Sigma\, P(OSj|Hi) \\
&= \Sigma\, P(O|SjHi)P(Sj|Hi) \\
&= \Sigma\, P(O|Sj)P(Sj) \\
&\approx \Sigma\, P(O|Sj),\ Hi \in VS(S_j)
\end{aligned}
$$

The computation steps for *maximum a posteriori* (MAP) hypothesis with respect to M are:

1. Maintain in parallel all VSs consistent with at least one supposed example from set M.
2. Initialize VS_0 to the VS with the maximum probability.

3. Incrementally update the VS_i by merging it with the next most probable VS. Compute the a posteriori probability of the new VS by summing the probabilities of the supposed examples it is consistent with.
   Carry out step 3 repeatedly while it does not result in a collapse of the VS.
4. Finally, return the VS with maximum a posteriori probability.

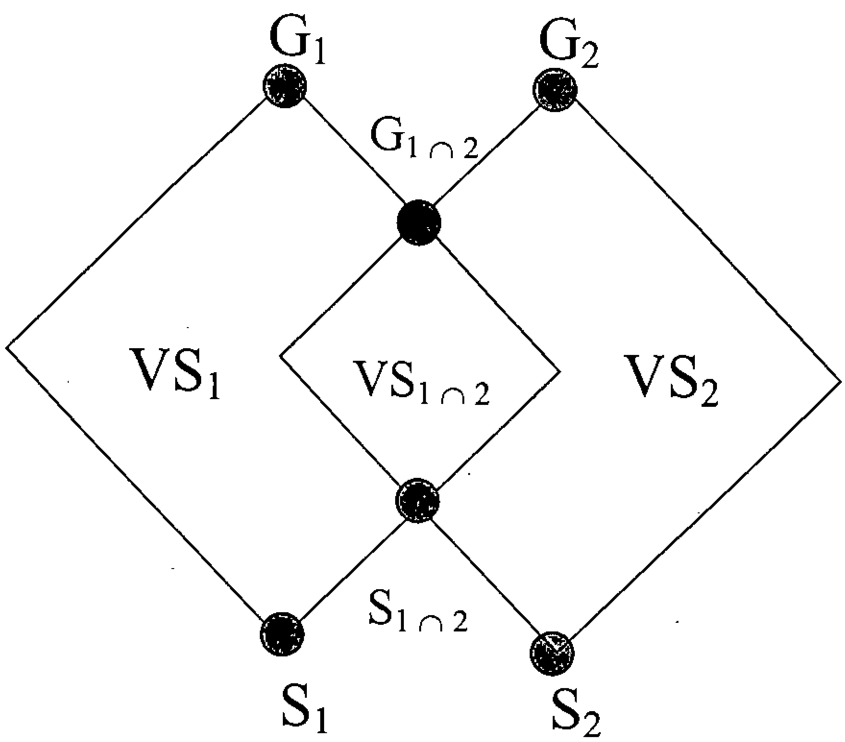

Figure 3.3: Version Space Merging

The computation for step 3 is illustrated by figure 3.3. We compute boundary sets of the new version space by merging two version spaces and the a posteriori probability of it is also computed. The boundary sets are computed by applying the intersection operator and the a posteriori probability is computed by applying the summation operator.

3.4 Implementation of step 3

Implementation of Step 3 is carried out by Integrated Analytical/Empirical method.

3.4.1 Integrated Analytical/Empirical Method

The objective is to produce a minimum syntactic revision of given classification theory C so that the resultant theory is consistent with the given training set N and also the coverage of the computed set A remains intact.

Two non-interfering correction steps are taken to accomplish this goal, viz. theory generalization and theory specialization. They are described in the next two subsections.

3.4.1.1 Theory Generalization

First, we use a greedy covering algorithm [5] to find the *minimum antecedent cover*. Next, for each rule in the cover, the following three operators are used for generalization of a rule:

- Antecedent retraction
- Antecedent generalization
- Inductive rule addition

They are tried successively in the order given.

<u>Antecedent retraction</u>: For each rule, the first step is to remove its conflicting antecedents. This deletion is permanent, unless it creates false positives.

<u>Antecedent generalization</u>: If the first operator results in an over-generalization, then we attempt to generalize the conflicting antecedents just enough to cover the failing positive examples of the rule. For discrete antecedents, disjuncts are added for

the values present in the failing positive examples. For binary antecedents, the antecedent is removed.

Inductive Rule Addition: If both the first and second operator results in an over-generalization, then the inductive component is used to learn entirely new rules for the consequent of the given rule. The set of positive and negative examples is determined in the following way. The positive examples are simply the failing positives for the rule.. The negative examples are obtained by removing all the antecedents from the rule and collecting any new failing negative examples that are created by this action. As the inductive component, C4.5 [17] is used.

3.4.1.2 Theory Specialization

First, we use a greedy covering algorithm [5] to find the *minimum rule cover.*

Next, for each rule in the cover the following two operators are used for specialization of a rule.

- Rule retraction
- Inductive antecedent addition

They are tried successively in the order given.

Rule retraction: For each rule, the first step is to remove it. If it does not create false negatives, this deletion is permanent.

Inductive Rule Addition: If the first operator fails, then the inductive component is used to add new antecedents to the rule. The set of positive and negative examples is determined in the following way. The positive examples are those that become failing positives when the rule is removed. The negative examples are the failing negative examples that use the rule in an erroneous proof. As the inductive component, C4.5 [17] is used.

3.5 The New Theory Revision System Algorithm

We conclude the chapter by presenting the algorithm of the new theory revision system:

Given: The Classifier C and the training set N

Steps:

1. Compute the VS (S and G) for a classifier Cn, based on the given classifier C and given set of data N (n+ and n-).

2. Compute A= Intersection (Cn, C) - N

$$\begin{cases} a+ = \text{intersection } (c+, s+) - n+ \\ a- = \text{intersection } (c-, g-) - n- \end{cases}$$

3. Compute a minimum syntactic revision of C so that N is covered correctly and coverage of A is unchanged.

Here, n+ and n- denote the positive and negative members of N respectively. Similarly, a+ and a- denote the positive and negative members of A respectively.

CHAPTER FOUR

EXPRIMENTAL RESULTS

The new theory revision system is tested on four domain theories to determine its ability to revise expert rule bases using training data. We show that our system is workable and also establish that the new system is comparable with other fairly successful systems like EITHER [14] and PTR [7].

4.1 Balloon Database Results

We tested the new theory revision system on the four data sets of Balloon database to show that the system is workable. The Balloon database has been used in [15] to investigate the influence of prior knowledge on concept acquisition.

<u>Results</u>: We tested the new theory revision system on the Balloon database to show that the system is workable.

- Four data sets of Balloon database were used. There are four data sets representing different conditions of an experiment. There are four binary attributes in this domain and the number of instances in each data set is 16.
- The number of training examples was varied from 8 to 12, with the training and test examples drawn at random with no overlap. In each test, the classification accuracy is measured using 4 disjoint test examples.
- We report 100% success with the new system in this domain. In all cases, the new system outputs the correct concept. For example, for the fourth data set of the database,

  The initial theory: (yellow and large) or (adult and child)

  The target theory: (yellow and small) or (adult and stretched).

  The target theory is correctly found from the initial theory.

We establish the workability of the new system by this result.

4.2 Lens Data set Results

We tested the new theory revision system on the Lens data set to show that the system is workable.

- There are four attributes in this domain and the number of instances in the data set is 24.
- The number of training examples was 6 to 10, with the training and test examples drawn at random with no overlap.
- The initial theory is described in figure 4.1 and its accuracy is 3/24
- The final theory is described in figure 4.2 and its accuracy is 2/24

We establish the workability of the new system by this result.

```
none:-tear-rate=1.
hard:-tear-rate=2, astigmatic=2.
soft:-tear-rate=2, astigmatic=1.
```

Figure 4.1: Initial theory (Lens data set)

```
none:-tear-rate=1.
none:-tear-rate=2, astigmatic=2, spectacle=2.
hard:-tear-rate=2, astigmatic=2, spectacle=1.
soft:-tear-rate=2, astigmatic=1.
```

Figure 4.2: Final theory (Lens data set)

4.3 Post-Operative Data set Results

We tested the new theory revision system on the Post-operative data set to show that the system is workable.

- There are nine attributes in this domain and the number of instances in the data set is 47.
- The number of training examples was 6 to 15, with the training and test examples drawn at random with no overlap.
- The initial theory is described in figure 4.3 and its accuracy is 10/47
- The final theory is described in figure 4.4 and its accuracy is 6/47

We establish the workability of the new system by this result.

```
S:-COMFORT=low.
S:-BP-STBL=stable, CORE-STBL=unstable.
S:-BP-STBL=stable, CORE-STBL=stable.
S:-BP-STBL=stable, L-SURF=mid, L-CORE=low.
S:-BP-STBL=unstable, COMFORT=mid-low.

A:-BP-STBL=mod-stable.
A:-BP-STBL=stable, CORE-STBL=stable, L-SURF=low.
A:-BP-STBL=stable, CORE-STBL=stable, L-CORE=mid.
A:-BP-STBL=unstable, (L-SURF=high; L-SURF=low).
A:-BP-STBL=unstable, COMFORT=mid-high.
```

Figure 4.3: Initial theory (Post-operative data set)

```
S:-COMFORT=low.
S:-BP-STBL=stable, CORE-STBL=unstable.
S:-BP-STBL=stable, CORE-STBL=stable, L-SURF=high.
S:-BP-STBL=stable, CORE-STBL=stable, L-SURF=mid ,L-CORE=low.
S:-BP-STBL=unstable, L-SURF=low.
S:-BP-STBL=unstable, L-SURF=mid, COMFORT=mid-low.

A:-BP-STBL=mod-stable.
A:-BP-STBL=stable, CORE-STBL=stable, L-SURF=low.
A:-BP-STBL=stable, CORE-STBL=stable, L-SURF=mid, L-CORE=mid.
A:-BP-STBL=unstable, L-SURF=high.
A:-BP-STBL=unstable, L-SURF=mid, COMFORT=mid-high.
```

Figure 4.4: Final theory (Post-operative data set)

4.4 DNA Promoter Recognition Data set Results

The new theory revision system is tested on a theory for recognizing biological concepts in DNA sequences. The original theory is described in [12]. It contains 11 rules with a total of 76 propositional symbols. The purpose of the theory is to recognize promoters in strings of nucleotides. A promoter is a genetic region, which initiates the first step in the expression of an adjacent gene.

The input features are 57 sequential DNA nucleotides. There are 106 examples in the data set, consisting of 53 positive and 53 negative examples. The initial imperfect theory classified none of the positive examples and all of the negative examples.

Figure 4.5 shows a subset of the promoter data set and figure 4.6 describes the given initial promoter theory.

i) t, a, c, t, a, g, c, a, a, t, a, c, g, c, t, t, g, c, g, t, t, c, g, g, t, g, g, t, t, a, a, g, t, a, t, g, t, a, t, a, a, t, g, c, g, c, g, g, g, c, t, t, g, t, c, g, t, positive.

ii) t, c, g, a, t, a, a, t, t, a, a, c, t, a, t, t, g, a, c, g, a, a, a, a, g, c, t, g, a, a, a, a, c, c, a, c, t, a, g, a, a, t, g, c, g, c, c, t, c, c, g, t, g, g, t, a, g, positive

iii) a, a, c, t, c, a, a, g, g, c, t, g, a, t, a, c, g, g, c, g, a, g, a, c, t, t, g, c, g, a, g, c, c, t, t, g, t, c, c, t, t, g, c, g, g, t, a, c, a, c, a, g, c, a, g, c, g, negative

iv) t, t, a, c, t, g, t, g, a, a, c, a, t, t, a, t, t, c, g, t, c, t, c, c, g, c, g, a, c, t, a, c, g, a, t, g, a, g, a, t, g, c, c, t, g, a, g, t, g, c, t, t, c, c, g, t, t, negative

Figure 4.5: A subset of the promoter data set.

```
positive:- (X;Y), Z.
X :- p-37=c, p-36=t, p-35=t, p-34=g, p-33=a, p-32=c.
X :- p-36=t, p-35=t, p-34=g, p-32=c, p-31=a.
X :- p-36=t, p-35=t, p-34=g, p-33=a, p-32=c, p-31=a.
X :- p-36=t, p-35=t, p-34=g, p-33=a, p-32=c.

Y :- p-14=t, p-13=a, p-12=t, p-11=a, p-10=a, p-9=t.
Y :- p-13=t, p-12=a, p-10=a, p-8=t.
Y :- p-13=t, p-12=a, p-11=t, p-10=a, p-9=a, p-8=t.
Y :- p-12=t, p-11=a, p-7=t.

Z :- p-47=c, p-46=a, p-45=a, p-43=t, p-42=t, p-40=a, p-39=c,
     p-22=g, p-18=t, p-16=c, p-8=g, p-7=c, p-6=g, p-5=c, p-4=c, p-2=c, p-1=c.
Z :- p-45=a, p-44=a, p-41=a.
Z :- p-49=a, p-44=t, p-27=t, p-22=a, p-18=t, p-16=t, p-15=g, p-1=a.
Z :- p-45=a, p-41=a, p-28=t, p-27=t, p-23=t, p-21=a, p-20=a, p-17=t, p-15=t, p-4=t.
```

Figure 4.6: Initial Promoter Theory

Positive:- p-35=t, p-34 =g
Positive:- p-35=t, p-34=t, p-45 = a
Positive:- (p-36=t; p-36=g), p-34=g
Positive:- p-35=t, p-34=c, p-47=g
Positive:- p-36=t, p-34=t, (p-33=a; p-33=c)
Positive:- p-36=c, p-33=a, p-31= a

Figure 4.7: Final Promoter theory

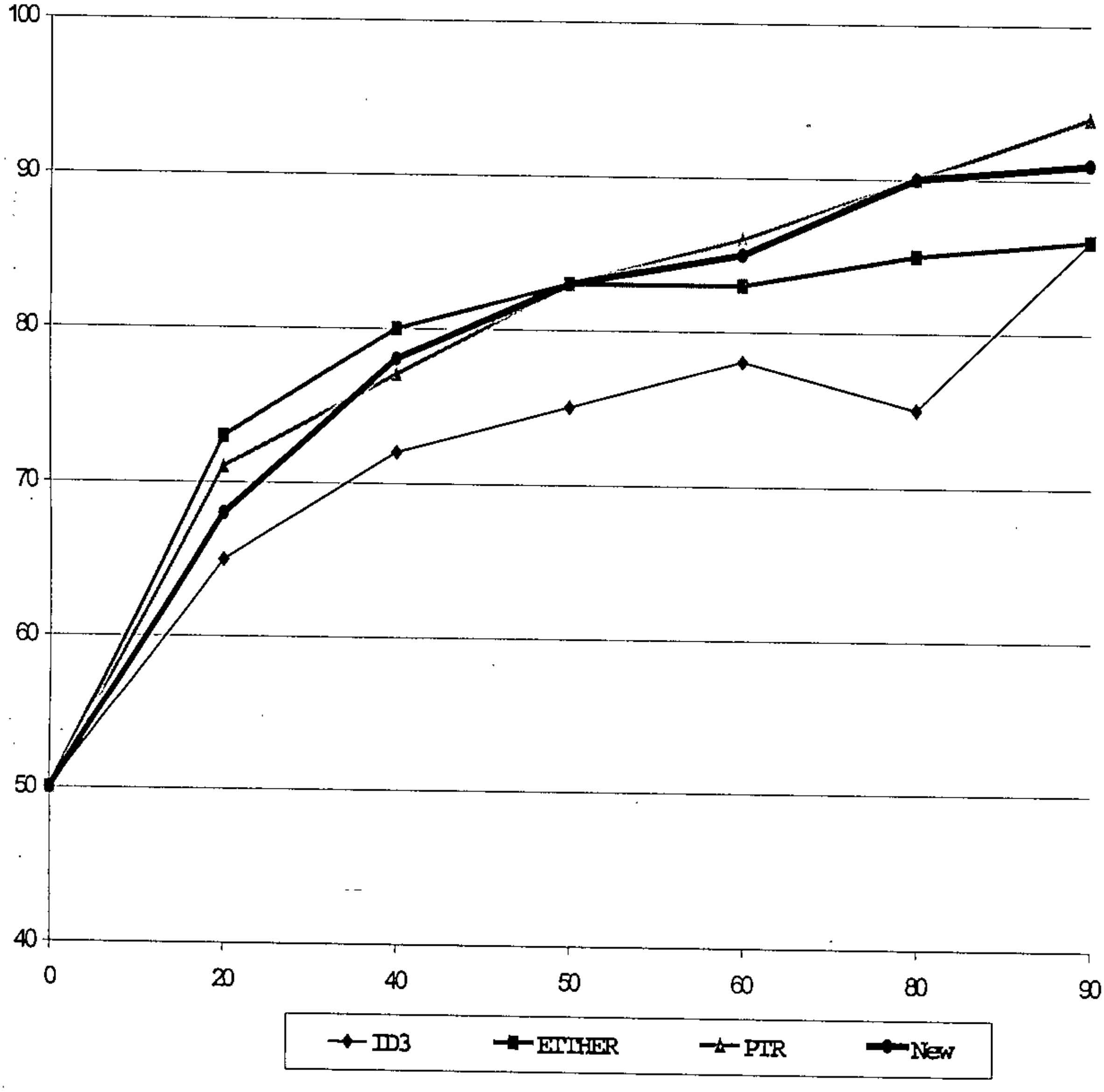

Figure 4.8: Results for DNA Promoter Recognition

Figure 4.8 shows the learning curves for the new theory revision system, EITHER [14], PTR [7] and ID3 [16]. For the new system, the number of training examples was varied from 1 to 90, with the training and test examples drawn at random with no overlap. In each test, the classification accuracy is measured using 16 disjoint test examples. The results were averaged over 20 training/test division.

It is found that the performance of the new system is comparable with the existing successful theory revision system like EITHER and PTR. As the size of the training set increases, the performance of the new system also grows and it is better than EITHER for large training set. The performance of ID3, which is a purely inductive system, shows that the performances of systems using prior knowledge, like EITHER and the new system, are superior. Additional examples are required by ID3 to match the performance of EITHER.

It is found that PTR performs better than the new system in this domain. PTR is inspired by KBANN [18], which is especially successful for this promoter domain. The performance of KBANN in the domain is reported to be better than EITHER, PTR, ID3 and the new system.

There is a set of 13 out of the 106 examples, each of which contain information substantially different than that in the rest of the examples. It is reported in [7] that training on all 93 of these examples and testing on the 13 special examples results in less than 40% accuracy, but using ten-way cross-validation on the 93 examples results in more than 99% accuracy. This explains the dip in the performance for a training set of size 60 for EITHER and the new system in figure 4. The similar performance fall for ID3 takes place for a training set of size 80.

CHAPTER FIVE

CONCLUSION

We have developed a new theory revision system. Experimental results show that the new system is comparable with other fairly successful systems like EITHER and PTR.

The new system is built by incorporating Version Space-based Incremental Probabilistic Evidence Combination method and Integrated Analytical/Empirical method. To our knowledge, Version Space-based approach has not been applied for theory revision problem before.

There is similarity between TANGENTPROP [19], PTR [7] and our system. Each of these systems utilize the provided classification information of the training set to predict classification information of a part of instance space, which is not given. We claim our system to be more robust. TANGENTPROP cannot characterize those neighbors that do not coincide with the training derivative of the given function. On the other hand, the performance of PTR is dependent on the initial probabilities assigned, presumably from a human expert.

There are several ways of extending the current system. The current system is built for Propositional logic system. A future direction of research is to extend the strategy incorporated in the current system to a Theory Revision system for First-Order logic domain. Another way of extension will be to provide a degree of belief measure for the actual observance of the determined subset A, out of the total instance space X, during the construction phase of the given theory.

References

[1] Bradshaw, G., Fozzard, R. and Cice, L., *A connectionist expert system that really works*, Advances in Neural Information Processing, (1989) SanMateo, CA: Morgan Kaufmann, pp. 474-492 .

[2] Fu, L.M., *Knowledge-based Connectionism for revising domain theories*, IEEE Transactions on Systems, Man and Cybernatics, **23** (1), (1993), pp. 173-182.

[3] Ginsberg, A., *Theory reduction, theory revision, and retranslation*, Proceedings AAAI-90, Boston, MA (1990), pp. 777-782.

[4] Gallant, S.I., *Connectionist Expert Systems*, CACM, **31** (2), (1988), pp. 152-169.

[5] Johnson, D. S., *Approximation algorithms for combinatorial problems*, Journal of Computaional System Science, 9, (1974), pp. 256-278.

[6] Kedar-Cabelli, S. and McCarty, T., *Explanation-based generalization as resolution theorem proving*, Proceedings of the Fourth International Workshop on Machine Learning, (1987), pp. 383-389.

[7] Koppel, M., Feldman, R. and Segre, A.M., *Bias-Driven Revision of Logical Domain Theories*, Journal of Artificial Intelligence Research 1 (1994), pp. 159-208

[8] Lacher, R., Hruška, S. and Kuncicky, D., *BackPropagation Learning in expert networks*, Technical Report of Dept. of Computer Science, TR91-015, Florida Satte University, Tallahassee, (1991).

[9] Michaski, R.S., Mozetic, I., Hong J. and Larvac,, N., *The multi-pupose incremental learning system AQ15 and its testing application to three medical domain*, Proceedings AAAI-86, Philadelphia, PA (1986), pp. 1041-1045.

[10] Mitchell, T.M., *The Need for Bias in Learning Generalizations*, Readings in Machine Learning, Morgan Kaufmann (1980), pp.184-191.

[11] Muggleton, S. and Raedt, L. De, *Inductive Logic Programming: Theory and Methods*, Journal of Logic Programming. **19**, (1994), pp. 629-679.

[12] Murphy, P.M. and Aha, D.W., *UCI Repository of Machine Learning Databases*, Irvine, CA: Department of Information and Computer Science, University of California at Irvine.

[13] Norton, S. W. and Hirsh, H., Classifier Learning from Noisy Data as Probabilistic Evidence Combination, AAAI-92 Proceedinngs, (1992).

[14] Ourston, D. and Mooney, R., *Theory Refinement Combining Analytical and Empirical Method*, Artificial Intelligence, **66** (1994), pp. 273-309.

[15] Pazzani, M. and Brunk, C., *Detecting and Correcting Errors in Rule-Based Expert Systems: An Integration of Empirical and Explanation-based Learning*, Knowledge Acquisition, 3(2), (1991), pp. 157-173.

[16] Quinlan J. R., *Induction of Decision Trees*, Machine Leaning, **1** (1986), pp. 81-106.

[17] Quinlan, J .R. C4.5: *Programs for Machine Learning*. Morgan Kaufmann Publishers. San Mateo, California, 1992.

[18] Shavlik, J. and Towell, G. *An approach to combining explanation-based and neural learning algorithms*, Connection Science **1**(3) (1989), pp. 233-255.

[19] Simard, P.S., Victorri B., LeCun Y. and Denker J., *TangentProp-A formalism for specifying selected invariances in an adaptive network.* Advances in Neural Information Processing Systems, San Mateo, CA: Morgan Kaufmann (1992).

[20] Smirnov, E.N., Van den Herik, H.J., and Sprinkhuizen-Kuyper, I.G., *An Adaptable Version-Space Representation*, BNAIC-2001 Proceedings of the 13th Dutch-Belgian Artificial Intelligence Conference, (2001), pp. 233-240.

[21] Smirnov, E.N., Van den Herik, H.J., and Sprinkhuizen-Kuyper, I.G., *Adaptable Boundary Sets*, AI & M 29-2002, Seventh International Symposium on Artificial Intelligence and Mathematics, AMAI, (2002).

[22] Thrun, S., *Explanation Based Neural Network Learning: A Lifelong Appraoch.*, Boston, Kluwer Academic Publishers, (1996).