Information and Decision Theoretic Approaches to Problems in Active Diagnosis

by

Gowtham Bellala

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering : Systems)
in The University of Michigan
2012

Doctoral Committee:

Assistant Professor Clayton D. Scott, Chair
Professor Alfred O. Hero III
Professor Susan A. Murphy
Associate Professor Sandeep P. Sadanandarao

To my Parents
Kameswara Rao Bellala and (Late) Sulochana Bellala

ACKNOWLEDGEMENTS

I would like to express my sincerest gratitude to my advisor, Professor Clayton Scott for accepting me as a graduate student, for introducing me to an active research field that has proven to be both stimulating and enjoyable, for his dedication in nurturing me, for his intellectual support and constant motivation, and for giving me the freedom to explore ideas. His guidance and support have been essential not only in the development of this thesis, but also in my overall development as a researcher. I have thoroughly enjoyed working with him and learning from him.

I would also like to thank Professor Suresh Bhavnani for introducing me to the problem of toxic chemical identification, which strongly motivated the work in this thesis. I am also very grateful to have had an opportunity to work with him on several other interesting projects. I have greatly admired his vision, and his in-depth knowledge of the domain problems. It has been a pleasure working with him through these years. I would also like to thank all the members of my Doctoral committee, Prof. Alfred Hero, Prof. Susan Murphy and Prof. Sandeep Pradhan, for their excellent feedback and support.

I would also like to thank my colleagues and lab mates, Jason Stanley, Gyemin Lee, JooSeuk Kim, Takanori Watanabe and Robert Vandermeulen, for many discussions and helpful suggestions.

My heartfelt thanks go to all my roommates and friends, who have been my extended family for the past few years. Particularly, I would like to thank my current and ex-roommates, Kumar Sricharan, Pradeep Muthukrishnan, Karthik Kumar, Tushar Agarwal and Hari Bharadwaj, who has made my stay at Ann Arbor very memorable. I would also like to thank all my friends (particularly Aneesh Reddy, Sudhir Vinjamuri and Veeresh Thummadi), and all my relatives in the US, who made me feel closer at home.

Finally, this thesis would not have been complete without the support of my family. I am extremely grateful to my mom Sulochana Bellala, who has been a great source of inspiration all through my life. Words cannot express my gratitude to her for all the sacrifices she made to help me pursue my dreams. I am forever indebted to her and to my entire family, my dad Kameswara Rao Bellala, my brother Srikanth Bellala and my sister-in-law Ramya Bellala, for their love and support and for putting my happiness before their own.

TABLE OF CONTENTS

DEDICATION ii

ACKNOWLEDGEMENTS iii

LIST OF FIGURES viii

LIST OF TABLES xi

LIST OF APPENDICES xii

ABSTRACT xiii

CHAPTER

I. Introduction 1

II. Background 6

2.1 Special cases of the Object Identification Problem 8
2.1.1 Source Coding 9
2.1.2 Guessing 9
2.2 Other Related Problems 10
2.2.1 Preference Elicitation 10
2.2.2 Adaptive Group Testing 11
2.2.3 Adaptive Conjoint Analysis 12
2.3 Prior Work and their Limitations 13
2.4 Overview of our Approach 16
2.5 Motivating Applications 17
2.5.1 Emergency Response 17
2.5.2 Fault Diagnosis in Computer Networks 18

III. Group Diagnosis 20

3.1 Introduction 20

3.1.1 Notation 22
3.2 Coding-Theoretic Interpretation of Object Identification . . . 23
3.3 Group Identification 26
3.3.1 Relation to Impurity-based Decision Tree Induction 31
3.3.2 A Near-optimal Algorithm 33
3.4 Object Identification under Group Queries 35
3.5 Group Identification under Group Queries 39
3.6 Object Identification under Persistent Noise 42
3.6.1 Constant Noise Rate 45
3.7 Experimental Evaluation 46
3.7.1 Group Identification 47
3.7.2 Object Identification under Group Queries 51
3.7.3 Object Identification under Persistent Noise 53

IV. Diagnosis under Exponential Query costs 57

4.1 Introduction 57
4.2 Object Identification under Exponential Costs 59
4.2.1 Average case scenario 61
4.2.2 Worst case scenario 61
4.3 Group Identification under Exponential Costs 62
4.3.1 Average case scenario 65
4.3.2 Worst case scenario 66
4.4 Experimental Evaluation 67

V. Diagnosis under Persistent Query Noise 69

5.1 Introduction 69
5.2 Data Model 71
5.3 Active Diagnosis under Persistent Noise 73
5.3.1 Entropy-based Active Query Selection 74
5.3.2 Rank-based Active Query Selection 75
5.3.3 Noise Adaptive Active Query Selection 77
5.4 Experimental Evaluation 81

VI. Multiple Fault Diagnosis 85

6.1 Introduction 85
6.2 Data Model 88
6.3 Active Diagnosis under Multiple Faults 90
6.3.1 Entropy-based Active Query Selection 91
6.3.2 AUC-based Active Query Selection 93
6.4 Experimental Evaluation 103

VII. Concluding Remarks 106

7.1 Summary of Contributions 106
7.2 Future Directions 107

APPENDICES 110

BIBLIOGRAPHY 154

LIST OF FIGURES

Figure

2.1 A bipartite diagnosis graph (BDG) corresponding to an object identification problem with 3 objects and 5 queries, where $\Theta = \{\theta_1, \theta_2, \theta_3\}$ and $Q = \{q_1, q_2, q_3, q_4, q_5\}$ with $q_1 = q_3 = \{\theta_1\}$, $q_2 = q_4 = \{\theta_2, \theta_3\}$, and $q_5 = \{\theta_1, \theta_3\}$. 6

2.2 A screen shot of the WISER decision support system. 17

2.3 A toy example demonstrating a system of networked computers along with probe stations and probes. 19

3.1 Toy Example 1 28

3.2 Decision tree constructed using GBS for group identification on toy example 1 28

3.3 Toy Example 2 36

3.4 Decision tree constructed on toy example 2 for object identification under group queries 36

3.5 For the toy example shown in (a) consisting of 2 objects and 3 queries with an $\epsilon = 1$, (b) demonstrates the construction of matrix $\widetilde{\mathbf{B}}$. The probability distribution of the objects in $\widetilde{\mathbf{B}}$ are generated using the noise model described in Section 3.6.1, where only queries q_2 and q_3 are assumed to be prone to error. 43

3.6 Random data model - The query parameters $(\gamma_w(q), \gamma_b(q))$ are restricted to lie in the rectangular space 47

3.7 Expected number of queries required to identify the group of an object using GBS, GISA and modified GISA on random datasets generated using the proposed random data model. Note that GISA and modified GISA achieve almost similar performance on these datasets, with GISA performing slightly better than modified GISA. 48

3.8 Scatter plot of the query parameters in the WISER database 49

3.9 Compares the average query complexity of different algorithms for object identification under group queries in synthetic datasets . . . 52

3.10 Compares the performance of GBS, modified GISA and GISA in identifying the true object in the presence of persistent query noise described in Section 3.6.1 for $p = 0.5$. 54

3.11 (a) Compares the performance of GBS, modified GISA and GISA in identifying the true object in the presence of persistent query noise described in Section 3.6.1 for different values of p (b) Compares the performance of GBS and GISA under persistent noise in the presence of discrepancies between the true value of p, p_{true} and the value used in the algorithm p_{alg} 55

4.1 Experiments to demonstrate the improved performance of λ-GBS over GBS and GBS with uniform prior. The plots in the first column correspond to the WISER database and those in the second column correspond to synthetic data. 67

5.1 A toy bipartite diagnosis graph (BDG) where the circled nodes denote the objects and the square nodes denote queries. 72

5.2 Demonstration of worst case ranking 76

5.3 The first column corresponds to a dataset generated using the ER model, the second column corresponds to a dataset generated using the PA model, the third column corresponds to the WISER database and the last column corresponds to a BRITE network. In all the experiments, the rank-based algorithm has no knowledge of the noise parameters. 82

5.4 Demonstrates the sensitivity of entropy-based query selection to mis specification of the noise parameters 84

6.1 (top) A toy bipartite diagnosis graph, (bottom) A Bayesian network corresponding to the given BDG. 88

6.2 A rank order corresponding to this example is $\mathbf{r} = (3, 1, 2, 4, 5)$. . . . 93

6.3 Demonstrates the different approximations for area under the ROC curve 95

6.4 Demonstrates the competitive performance of the AUC-based query selection under single fault assumption to that of BPEA, while having a computational complexity that is orders less (near quadratic vs. exponential complexity of BPEA). On INET, we only compare AUC+SF with Entropy+SF as BPEA becomes slow and intractable. 102

6.5 Comparison of time complexity of selecting a query using BPEA and AUC+SF. 104

D.1 Demonstrates the ROC curve corresponding to a ranked list of objects, when there is only one object in state 1 151

LIST OF TABLES

Table

3.1 Expected number of queries required to identify the group of an object in the WISER database 50

3.2 Expected number of queries required to identify an object under group queries in the WISER database 53

LIST OF APPENDICES

Appendix

A. Appendix for Group Diagnosis 111

B. Appendix for Diagnosis under Exponential Query costs 127

C. Appendix for Diagnosis under Persistent Query Noise 134

D. Appendix for Multi Fault Diagnosis 141

ABSTRACT

Information and Decision Theoretic Approaches to Problems in Active Diagnosis

by

Gowtham Bellala

Chair: Clayton D. Scott

In applications such as active learning or disease/fault diagnosis, one often encounters the problem of identifying an unknown object while minimizing the number of "yes" or "no" questions (queries) posed about that object. This problem has been commonly referred to as object/entity identification or active diagnosis in the literature. In this thesis, we consider several extensions of this fundamental problem that are motivated by practical considerations in real-world, time-critical identification tasks such as emergency response.

First, we consider the problem where the objects are partitioned into groups, and the goal is to identify only the group to which the object belongs. We then consider the case where the cost of identifying an object grows exponentially in the number of queries. To address these problems we show that a standard algorithm for object identification, known as the splitting algorithm or generalized binary search (GBS), may be viewed as a generalization of Shannon-Fano coding. We then extend this result to the group-based and the exponential cost settings, leading to new, improved algorithms.

We then study the problem of active diagnosis under persistent query noise. Previous work in this area either assumed that the noise is independent or that the underlying query noise distribution is completely known. We make no such assumptions, and introduce an algorithm that returns a ranked list of objects, such that the expected rank of the true object is optimized. Finally, we study the problem of active diagnosis where multiple objects are present, such as in disease/fault diagnosis. Current algorithms in this area have an exponential time complexity making them slow and intractable. We address this issue by proposing an extension of our rank-based approach to the multiple object scenario, where we optimize the area under the ROC curve of the rank-based output. The AUC criterion allows us to make a simplifying assumption that significantly reduces the complexity of active diagnosis (from exponential to near quadratic), with little or no compromise on the performance quality. Further, we demonstrate the performance of the proposed algorithms through extensive experiments on both synthetic and real world datasets.

CHAPTER I

Introduction

In emergency response applications, as well as other time-critical diagnostic tasks, there is a need to rapidly identify a cause by selectively acquiring information from the environment. For example, in the problem of toxic chemical identification, a first responder may question victims of chemical exposure regarding the symptoms they experience. Chemicals that are inconsistent with the reported symptoms may then be eliminated. Because of the importance of this problem, several organizations have constructed extensive evidence-based databases (e.g., WISER[1]) that record toxic chemicals and the acute symptoms which they are known to cause. Unfortunately, many symptoms tend to be nonspecific (e.g., vomiting can be caused by many different chemicals), and it is therefore critical for the first responder to pose these questions in a sequence that leads to chemical identification in as few questions as possible.

This problem has been studied from a mathematical perspective for decades, and has been described variously as query learning (with membership queries) (*Angluin*, 2004), active learning (*Dasgupta*, 2004), active/adaptive diagnosis (*Rish et al.*, 2005) object/entity identification (*Garey*, 1970, 1972), and binary testing (*Garey*, 1972; *Loveland*, 1985). In this thesis we will refer to this problem either as object identification or as active diagnosis. The standard mathematical formulation of object identification is often idealized relative to many real-world diagnostic tasks, in that

[1] `http://wiser.nlm.nih.gov/`

it does not account for time constraints and resulting input errors. In this thesis we investigate algorithms that extend object identification to such more realistic settings by addressing the need for rapid response, and error-tolerant algorithms.

In an object identification problem, there is a set $\Theta = \{\theta_1, \cdots, \theta_M\}$ of M different objects and a set $Q = \{q_1, \cdots, q_N\}$ of N distinct subsets of Θ known as queries. An unknown object θ is generated from this set Θ with a certain *prior* probability distribution $\Pi = (\pi_1, \cdots, \pi_M)$, i.e., $\pi_i = \Pr(\theta = \theta_i)$. The goal is to determine the unknown object $\theta \in \Theta$ through as few queries from Q as possible, where a query $q \in Q$ returns a value 1 if $\theta \in q$, and 0 otherwise. An object identification algorithm thus corresponds to a decision tree, where the internal nodes are queries, and the leaf nodes are objects. Problems of this nature also arise in applications such as computer vision (*Geman and Jedynak*, 1996; *Swain and Stricker*, 1993), image processing (*Korostelev and Kim*, 2000), job scheduling (*Kosaraju et al.*, 1999), pool-based active learning (*Dasgupta*, 2004; *Nowak*, 2008; *Golovin and Krause*, 2010) and the adaptive traveling salesperson problem (*Gupta et al.*, 2010). Algorithms and performance guarantees have been extensively developed in the literature, as described in Chapter II.

In the context of toxic chemical identification, the objects are chemicals, and the queries are symptoms. An object identification algorithm will prompt the first responder with a symptom. Once the presence or absence of that symptom is determined, a new symptom is suggested by the algorithm, and so on, until the chemical is uniquely determined. In this thesis, we consider several variations on this basic object identification framework that are motivated by toxic chemical identification, and are naturally applicable to other time-critical diagnostic tasks. In particular, we can broadly classify our contributions into four main categories - group based active diagnosis, active diagnosis under exponential query costs, active diagnosis under persistent query noise and active diagnosis under multiple unknown objects.

First, we consider the problem of group diagnosis where Θ is partitioned into groups of objects, and it is only necessary to identify the group to which the unknown object θ belongs. This scenario often arises in the problem of toxic chemical identification, where the appropriate response to a toxic chemical may only depend on the class of chemicals to which it belongs (pesticide, corrosive acid, etc.). We also consider other group based settings that naturally arise in real-world diagnostic scenarios, as described in more detail in Chapter III.

In Chapter IV, we study the problem of diagnosis under exponential query costs. We begin by noting that the standard formulation for object identification along with the existing algorithms inherently assume that the cost of identifying an object grows linearly in the number of queries. This often results in requiring a large number of queries for diagnosis, especially for objects with low prior probabilities. However, this is not acceptable in time-critical applications such as emergency response where the cost of additional queries may grow significantly.

To address these two problems, we propose extensions of a standard object identification algorithm known as the splitting algorithm, or generalized binary search (GBS) to these settings. The proposed algorithms are derived in a common coding-theoretic framework, and are based on reinterpretation of GBS as a generalized form of Shannon-Fano coding. For more details, refer to Chapters III and IV.

We then consider the problem of active diagnosis under persistent query noise in Chapter V. Query noise corresponds to errors in the obtained query responses. Though the problem of diagnosis under query noise has been considered in the literature, it has often been assumed that the queries can be re-sampled, such that repeated querying results in independent query responses (*Kääriäinen*, 2006; *Nowak*, 2008, 2009). However, in most diagnosis problems, the query noise persists in that repeated querying results in the same query response. Moreover, the underlying noise distribution is often not known. Unlike the independent noise model where the unknown object θ can be identified with great certainty after sufficiently many queries, in the persistent noise model it may not be possible to identify θ even after all queries are made. Hence, we propose a novel rank-based approach where we output a ranked list of the objects in Θ based on their likelihood of being the unknown object θ. We propose a greedy algorithm to select queries such that the expected rank of this unknown object θ is minimized. Further, we show that the proposed algorithm can be implemented without any knowledge of the underlying query noise distribution.

Finally, in Chapter VI, we consider a more general setting of the above diagnosis problem that arises in applications such as medical diagnosis (*Heckerman*, 1990; *Jaakkola and Jordan*, 1999), fault diagnosis in nuclear plants (*Santoso et al.*, 1999), computer networks (*Rish et al.*, 2005; *Zheng et al.*, 2005), and power-delivery systems (*Yongli et al.*, 2006). In these applications, more than one object is often of interest, i.e., θ could now correspond to a subset of objects from Θ. For example, in a fault diagnosis problem where objects correspond to components and queries to probes or alarm responses, more than one component could be faulty and the goal is to identify all the faulty components. The problem of active diagnosis is now to identify this unknown set θ by obtaining (noisy) responses to as few queries as possible, where the query noise is persistent. In the recent years, this problem has been formulated as an inference problem on a Bayesian network, and the current algorithms for active diagnosis in this setting rely on belief propagation making them slow and intractable.

To address this issue, we propose an extension of our above rank-based algorithm to the multiple object scenario, where we choose queries sequentially such that the area under the ROC curve (AUC) of the rank-based output is maximized. The AUC criterion allows us to make a simplifying assumption that significantly reduces the complexity of active query selection (from the current exponential to near quadratic) in the multiple object scenario, with little or no compromise on the performance quality. In summary, we show that the proposed rank-based framework is a fast, robust, and a reliable approach for active diagnosis in large-scale, real-world diagnosis problems.

We demonstrate the performance of our proposed algorithms through extensive simulations on both synthetic as well as real world datasets. In particular, we demonstrate our results on two real world datasets, the first one is a toxic chemical database used by first responders known as WISER, and the second corresponds to network topologies built using the BRITE (*Medina et al.*, 2001) and the INET (*Winick and Jamin*, 2002) generators that arise in the problem of fault diagnosis in computer networks.

CHAPTER II

Background

In diagnosis problems, there is a set $\Theta = \{\theta_1, \cdots, \theta_M\}$ of M different objects and a set $Q = \{q_1, \cdots, q_N\}$ of N distinct subsets of Θ known as queries. The relation between the objects and the queries can be captured using a bipartite diagnosis graph (BDG) as shown in Figure 2.1. The edges in this graph represent the relation or the interactions between the two entities. For example, in the toxic chemical identification problem, objects correspond to toxic chemicals and queries to symptoms, where an edge indicates that a particular symptom is exhibited by the presence of that toxic chemical. Similarly, in a fault diagnosis problem, objects may correspond to components and queries to alarms, where an edge indicates that a particular component-alarm pair are connected.

The relation between the objects and the queries can also be denoted using a

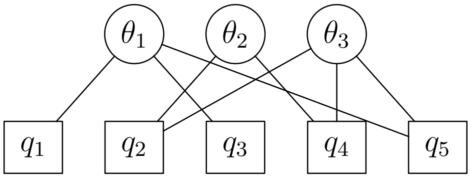


Figure 2.1: A bipartite diagnosis graph (BDG) corresponding to an object identification problem with 3 objects and 5 queries, where $\Theta = \{\theta_1, \theta_2, \theta_3\}$ and $Q = \{q_1, q_2, q_3, q_4, q_5\}$ with $q_1 = q_3 = \{\theta_1\}$, $q_2 = q_4 = \{\theta_2, \theta_3\}$, and $q_5 = \{\theta_1, \theta_3\}$.

binary matrix $\mathbf{B}$, where the rows correspond to different objects and columns to queries, with the binary entries in the matrix corresponding to the presence/absence of edges. The binary matrix corresponding to the bipartite diagnosis graph in Figure 2.1 is given by,

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

In the problem of object identification, there is an unknown object θ generated from this set Θ with a certain prior probability distribution $\Pi = (\pi_1, \ldots, \pi_M)$, where $\pi_i = \Pr(\theta = \theta_i)$. The goal of object identification is to identify this unknown object θ using as few queries from the set Q as possible. In the ideal scenario when there is no noise, a query $q \in Q$ returns a value 1 if $\theta \in q$, and 0 otherwise. In other words, the true responses to the queries correspond to the entries in the binary row vector associated with the unknown object in the matrix $\mathbf{B}$.

As mentioned earlier, problems of this nature arise in several applications such as job scheduling (*Kosaraju et al.*, 1999), image processing (*Korostelev and Kim*, 2000), computer vision (*Geman and Jedynak*, 1996; *Swain and Stricker*, 1993), pool-based active learning (*Dasgupta*, 2004; *Nowak*, 2008; *Golovin and Krause*, 2010) and the adaptive traveling salesperson problem (*Gupta et al.*, 2010). In an active learning setting, objects correspond to classifiers and queries to labels at specific unlabeled data points, with the goal of identifying the best classifier using as few labeled data as possible.

A solution to an object identification problem is a decision tree, where each internal node in the tree corresponds to a query, each leaf node corresponds to a unique object from the set Θ and the optimality is with respect to minimizing the expected depth of the leaf node corresponding to θ. In particular, the expected depth of a tree is given by $\sum_{i=1}^{M} \pi_i d_i$, where d_i corresponds to the depth of object θ_i in the tree. The problem of obtaining an optimal decision tree with the least expected depth has been studied extensively in the literature with *Garey* (1970) proposing a dynamic programming based algorithm. However, this algorithm runs in exponential time in the worst case. Later, *Hyafil and Rivest* (1976) showed that determining an optimal binary decision tree for this problem is NP-complete. Thereafter, various greedy algorithms (*Loveland*, 1985; *Kosaraju et al.*, 1999; *Roy et al.*, 2008) have been proposed to obtain a suboptimal binary decision tree.

Among the various greedy algorithms, the most widely studied algorithm is known as the *splitting algorithm* (*Loveland*, 1985) or *generalized binary search* (GBS) (*Dasgupta*, 2004; *Nowak*, 2008). This algorithm grows a binary decision tree in a top down greedy manner, where at each internal node, it selects a query that most evenly divides the probability mass of the remaining objects (*Loveland*, 1985; *Dasgupta*, 2004). The resulting tree has been shown to be near-optimal (*Loveland*, 1985; *Kosaraju et al.*, 1999; *Dasgupta*, 2004), in that the expected depth of the greedy tree is logarithmically close to that of an optimal tree, i.e.,

$$\mathbb{E}[\text{depth}_{GBS}] \leq O\left(\ln \frac{1}{\min_i \pi_i}\right) \mathbb{E}[\text{depth}_{opt}].$$

In addition, several variants of this problem such as multiway or k-ary splits (instead of binary splits) (*Chakaravarthy et al.*, 2007, 2009; *Cicalese et al.*, 2010) and unequal query costs (*Adler and Heeringa*, 2008; *Golovin and Krause*, 2010; *Gupta et al.*, 2010; *Cicalese et al.*, 2010) have also been studied in the literature.

2.1 Special cases of the Object Identification Problem

In this section, we will discuss two interesting special cases of the object identification problem described above. In these two cases, the problem of object identification reduces to a well-known, and well-studied problem in the literature. Moreover, though the problem of finding an optimal decision tree is NP-complete for a general object identification problem, there exists efficient, polynomial time algorithms to find optimal solutions in both these special cases.

2.1.1 Source Coding

In the special case when the query set Q is *complete* (a query set Q is said to be *complete* if for any $S \subseteq \Theta$ there exists a query $q \in Q$ such that either $q = S$ or $\Theta \setminus q = S$), the problem of object identification reduces to the problem of source coding. Here, the problem of constructing an optimal binary decision tree is equivalent to construction of optimal variable-length binary prefix codes with minimum expected length. This problem has been widely studied in information theory with both *Shannon* (1948) and *Fano* (1961) independently proposing a top-down greedy strategy to construct suboptimal binary prefix codes, popularly known as Shannon-Fano codes. Later, *Huffman* (1952) derived a simple bottom-up algorithm to construct optimal binary prefix codes. A well known lower bound on the expected length of binary prefix codes is given by the Shannon entropy of the probability distribution Π (*Cover and Thomas*, 1991). In fact, the problem of object identification when the query set Q is not *complete* can be considered as "constrained" prefix coding with the same entropy lower bound on the expected depth of the tree. This interpretation of object identification forms the basis of our results in Chapters III and IV.

2.1.2 Guessing

A query set Q is said to be *singleton complete* if Q contains all singleton queries, where a singleton query is a query that responds 1 to only one object, i.e., the query q is of the form $\{\theta_i\}$ for some $i \in \{1, \cdots, M\}$. In the special case when the query set Q is *singleton complete*, the problem of object identification reduces to the well-known problem of guessing. In guessing, the goal is to identify the value taken by a discrete random variable X in one trial of a random experiment by asking questions of the form "Did X take on its i^{th} possible value?" until the answer is "Yes", while minimizing the expected number of guesses required to identify the realization of X. This problem along with its variants have been studied extensively in the literature (*Massey*, 1994; *Arikan*, 1996; *Arikan and Merhav*, 1998; *Merhav and Arikan*, 1999; *Sundaresan*, 2007; *Hanawal and Sundaresan*, 2008). The problem of guessing is often encountered in applications such as cryptography and pattern matching. Given the prior probability distribution Π of the random variable X, the optimal guessing strategy is to guess in the decreasing order of these probabilities. Moreover, GBS when applied to this problem produces this optimal solution.

2.2 Other Related Problems

We now briefly mention other related problems that have been studied in the literature. We will describe interesting similarities between these problems to those we study in this thesis, along with their critical differences.

2.2.1 Preference Elicitation

The problem of preference elicitation arises in combinatorial auctions. It is the process of asking questions about the preferences of bidders so as to best divide some set of goods. The problem can be formalized more generally as follows. Consider a set S of M items that needs to be sold and let $x \in \{0,1\}^M$ denote any subset of items called an "example". Potentially there could be $N = 2^M - 1$ examples. Let there be k bidders where each bidder is associated with a preference function $f_k : \{0,1\}^M \to \mathbb{R}$, where $f_k(x)$ denotes the amount bidder k is willing to pay for example x or the subset of items in x. Now, the objective is to determine a k-way partition $(S_1, S_2, \cdots, S_k)$ of the set of items S such that $f_1(S_1) + f_2(S_2) + \cdots + f_k(S_k)$ is maximized. However, the preference functions of bidders are often unknown and it is not feasible to ask the bidder to provide his valuations for all the examples, which can be exponentially large. Hence, the problem of preference elicitation deals with learning the preference functions using as few queries as possible and then obtaining the best k-way partition, where a query could be of the form "How much are you willing to pay for example x?"

In the context of an object identification problem, it can be thought of as a setting in which there are multiple target objects that can each be queried separately, but where the goal is not so much to learn each target object as it is to produce an "optimal partition". For an extensive survey on preference elicitation methods, refer to (*Chen and Pu*, 2004). Also, for a more detailed analysis on the similarities and differences between preference elicitation and the problem of object identification, refer to (*Blum et al.*, 2004).

2.2.2 Adaptive Group Testing

Traditionally, group testing has been a design problem, where the goal is to construct an optimally efficient set of tests of items such that the test results contain enough information to determine a small subset of items of interest. The problem can be described more generally as follows. Consider a set S of M items, of which d items are defective. Let $\mathcal{D}$ denote the defective set. This defective set must be a member or sample of a given family called the sample space. For example, the sample space could be all subsets of M items of size d. Now, the goal of adaptive group testing is to construct a collection of tests to minimize the number of tests needed to find the defective set.

As a motivating example, consider the problem of identifying defective bulbs in a set of light bulbs. Here, an experiment or a test would be as follows. A batch of light bulbs would be arranged in series and an electrical voltage would be applied at either end. If the lights are on, then the whole tested batch of light bulbs must be good, else there is at least one bulb in the batch that is defective. Here, any subset of items could be selected to perform this experiment. The goal is to determine the experiments and the order in which they should be performed such that all the defective bulbs are identified in as few experiments as possible.

In the context of a diagnosis problem, it is similar to the problem of active diagnosis when multiple objects are present (i.e., θ is some unknown subset of Θ), with the goal of identifying all the unknown objects. However, the key difference is that in a diagnosis setting, the tests are constrained to be those from a fixed set of queries Q, where as in the case of group testing, the test could comprise of any subset of the M items (i.e. $2^M - 1$ possible tests). Moreover, in group testing, it is assumed that the size of the unknown set θ is known a priori. For an extensive study on the problem of adaptive group testing and its algorithms, refer to (*Du and Hwang*, 2000).

2.2.3 Adaptive Conjoint Analysis

Conjoint analysis is a statistical technique used in market research to determine how people value different features that make up an individual product or service. The objective of conjoint analysis is to determine what combination of a limited number of attributes is most influential on respondent choice or decision making. A controlled set of potential products or services is shown to respondents and by analyzing how they make preferences between these products, the implicit valuation of the individual elements making up the product or service can be determined. These implicit valuations then can be used to create market models that estimate market share, revenue and even profitability of new designs.

Here, a product is considered as a bundle of attributes, each with specified levels. For example, the attributes of a laptop computer can be weight, battery life, price, processor speed etc, where the attribute "price" can have three levels – less than

\$1000, \$1000 - \$2000, greater than \$2000. It is normally assumed that each attribute level has a particular value for a customer, which affects how much he/she likes the product. These values are called “utilities”. The goal of conjoint analysis is to estimate these utilities for various customers. As the number of attributes and the levels in each attribute increases, it becomes infeasible to ask the customer for his complete set of utilities.

In adaptive conjoint analysis, the customer is presented with trade-off questions in a sequential manner. For example, a trade-off question could be “Which would you prefer − 2.4GHz Intel **quad** core processor with **2** hours battery life or 2.4GHz Intel **single** core processor with **6** hours battery life?” The goal here is to learn the utility function of the customer in as few trade-off questions as possible.

Comparing to a diagnosis problem, there are some interesting similarities as well as some critical differences. The similarity being that in a diagnosis problem, the goal is to learn the binary value associated with each object (1 if $\theta_i \in \theta$, and 0 else), and in the case of adaptive conjoint analysis, it is the utility associated with each attribute and each level. However, once again the key difference is that there is no restriction on the queries to be made unlike in a diagnosis setting where the queries are restricted to be from the set Q. For more details on adaptive conjoint analysis, refer to (*Johnson*, 1987; *Johnson et al.*, 2003).

2.3 Prior Work and their Limitations

As mentioned in Chapter I, the main contributions of this thesis can be broadly classified into four categories - group-based active diagnosis, active diagnosis under exponential query costs, active diagnosis under persistent query noise, and active diagnosis under multiple unknown objects. In this section, we will briefly describe any prior work in each of these four categories and state their limitations.

The problem of rapid group identification has been simultaneously studied by

Golovin et al. (2010), who like us, also proposed a near-optimal algorithm, which is discussed in more detail in Chapter III. The problem of diagnosis under exponential query costs has been studied earlier in the special case where the query set Q is complete, i.e., in the context of source coding for the design of prefix-free codes (*Campbell*, 1965). In this special case, it has also been shown that an optimal binary decision tree (i.e., optimal binary prefix-free codes) can be obtained using a modified version of the Huffman algorithm (*Hu et al.*, 1979; *Parker*, 1980). However, to the best of our knowledge, there does not exist any optimal or suboptimal algorithm for the general case where the query set Q is not complete. For more details, refer to Chapter IV or *Bellala et al.* (2010).

The problem of rapid object identification in the presence of query noise has been studied in the literature (*Kääriäinen*, 2006; *Nowak*, 2008, 2009) where the query noise is assumed to be independent, such that repeated querying may result in different responses. However, in many diagnosis applications, re-sampling or repeating a query does not change the query response confining an algorithm to non-repeatable queries. The work by *Rényi* (1961) is regarded to be the first to consider this more stringent noise model, also referred to as persistent noise in the literature (*Goldman et al.*, 1990; *Jackson et al.*, 1997; *Hanneke*, 2007). However, his work has focused on the passive setting where the queries are chosen at random. The problem of pool-based active learning under persistent noise has been studied by *Balcan et al.* (2006) and *Hanneke* (2007) in the PAC (Probably Approximately Correct) model. However, they assume that the query set is large enough (possibly infinite) such that it is possible to get arbitrarily close to the optimal classifier, for any given noise level.

In this thesis, we focus on the problem of object identification under persistent query noise where the query set is possibly finite. We address this problem in two parts. First, we consider a restricted noise setting where we limit the number of persistent errors such that unique identification of the unknown object is guaranteed.

Specifically, given an object identification problem **B**, we limit the number of persistent errors to half the minimum Hamming distance between any two object row vectors (refer to Section 3.6 for more details). In the special case when the query set Q is complete, this problem reduces to the problem of designing minimum length k-error correcting codes in communication theory, also referred to as the Rényi-Ulam's problem in the game-theoretic literature (*Pelc*, 2002). However, this problem has not been studied earlier in the general case where the query set Q is not complete.

We then consider a more general noise setting with no restrictions on the number of persistent errors. In this context, *Rish et al.* (2005) proposed an information gain based active diagnosis algorithm. However, this algorithm requires complete knowledge of the underlying query noise distribution, which is often not known. Refer to Chapter V for more details.

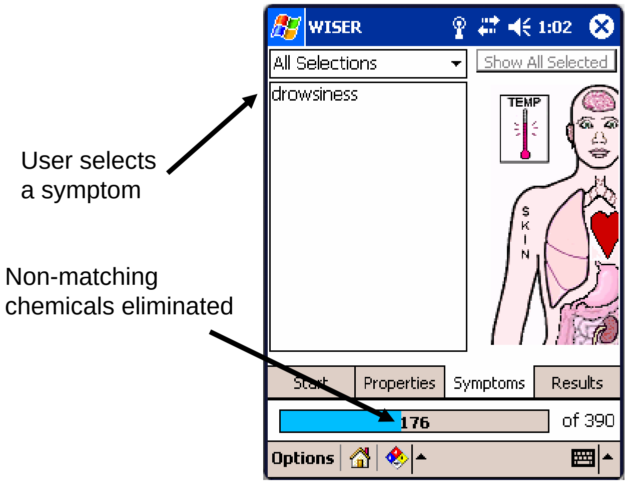
The problem of diagnosis when multiple objects are present has been studied in the recent years, where it has been formulated as an inference problem on a Bayesian network, with the goal of assigning most likely states to unobserved object nodes based on the outcome of the query nodes. In this context, *Zheng et al.* (2005) proposed the use of information gain for active query selection. Further, noting that exact computation of information gain is intractable in the multiple object scenario, they proposed an approximate algorithm based on loopy belief propagation (BP) to estimate the information gain. This algorithm, which they refer to as BPEA (Belief Propagation for Entropy Approximation) requires exactly one run of BP for each query selection. However, BPEA is not scalable as its complexity grows exponentially in the maximum degree of the underlying Bayesian network. More recently, *Cheng et al.* (2010) proposed a speed up to query selection using BPEA by reducing the number of queries to be investigated at each stage. However, the exponential complexity still remains. Refer to Chapter VI for more details.

2.4 Overview of our Approach

In this thesis, we propose algorithms that address the above limitations of the existing approaches. Our algorithms are derived in a common, principled framework. In particular, the proposed algorithms can be broadly classified into two settings as stated below.

- In Chapters III and IV, we consider extensions of the object identification problem where the unknown object θ can be identified with certainty. In particular, we consider the group-based settings, exponential query costs, and object identification under a restricted number of persistent errors. To address these problems, we first present a new interpretation of GBS from a coding-theoretic perspective by viewing the problem of object identification as constrained source coding. Specifically, we present an exact formula for the expected number of queries required to identify an unknown object in terms of Shannon entropy of the prior distribution Π, and show that GBS is a top-down algorithm that greedily minimizes this cost function. We then extend this framework to each of the above cases and derive extensions of GBS. The work in these chapters is based on *Bellala et al.* (2010) and *Bellala et al.* (2011b).

- In Chapters V and VI, we study the problem of object identification under persistent query noise in the single fault (only one unknown object) and multi-fault (multiple unknown objects) settings. In these problems, θ may not be identified even after obtaining responses to all the queries from the set Q. Hence, we modify the goal of active diagnosis to maximize the quality of the obtained estimate for θ while minimizing the number of queries. Specifically, we pose this problem as active diagnosis on a Bayesian network, and propose a novel rank-based approach where the algorithm returns a ranked list of the objects based on their posterior probabilities. We use area under the ROC curve (AUC)



User selects a symptom

Non-matching chemicals eliminated

All Selections

drowsiness

176 of 390

Figure 2.2: A screen shot of the WISER decision support system.

as a criterion to measure the quality of the obtained ranked list, and show how to choose queries actively such that the AUC is maximized. We also show how active query selection using the proposed AUC criterion overcomes the limitations of the existing approaches. The work in these chapters is based on *Bellala et al.* (2011a) and *Bellala et al.* (2011c).

2.5 Motivating Applications

We will now briefly describe two diagnosis applications that have primarily motivated the work in this thesis. In addition, we will be demonstrating the performance of our proposed algorithms on real world databases corresponding to these two applications in the rest of this thesis.

2.5.1 Emergency Response

In a recent study, *Kleindorfer et al.* (2003) reported that hundreds of toxic chemical accidents take place every year in the U.S. In the event of such an accident, the goal of a first responder is to rapidly identify the toxic chemical that may have leaked in to the environment. This rapid identification of the toxic chemical is needed to treat victims, decontaminate site, and issue neighborhood warnings. Owing to the importance of this problem, several organizations such as the NLM (National Library of Medicine) have constructed extensive evidence-based databases that record toxic chemicals and the acute symptoms they are known to cause. In addition, NLM has developed a decision-support system known as WISER (**W**ireless **I**nformation **S**ystem for **E**mergency **R**esponders) to aid first responders in rapid identification of the toxic chemical. The WISER database describes the binary relationship between 402 toxic chemicals and 79 acute symptoms.

Figure 2.2 shows a screen shot of the WISER system. It consists of a drop down menu containing the list of all symptoms. A first responder may question victims of chemical exposure regarding the symptoms they experience, and inputs this information in to the system. Chemicals that are inconsistent with the reported symptoms are then eliminated. Unfortunately, many symptoms tend to be non-specific. For example, *acute dyspnea* (difficulty breathing) can be caused by many different chemicals. Therefore, it is important for a first responder to pose these questions in a sequence that leads to chemical identification in as few symptom queries as possible.

2.5.2 Fault Diagnosis in Computer Networks

In the problem of fault diagnosis in computer networks, the goal is to continuously monitor a system of networked computers for faults, where each computer can be associated with a binary random variable X_i (0 for working and 1 for faulty). It is not possible to test each individual computer directly in a large network. Hence, a common solution is to test a subset of computers with a single test probe Z_j , where a probe can be as simple as a ping request or more sophisticated such as an e-mail message or a webpage-access request (see Figure 2.3). Thus, there is a bipartite diagnosis graph with each query (probe) connected to all the objects (computers) it passes through.

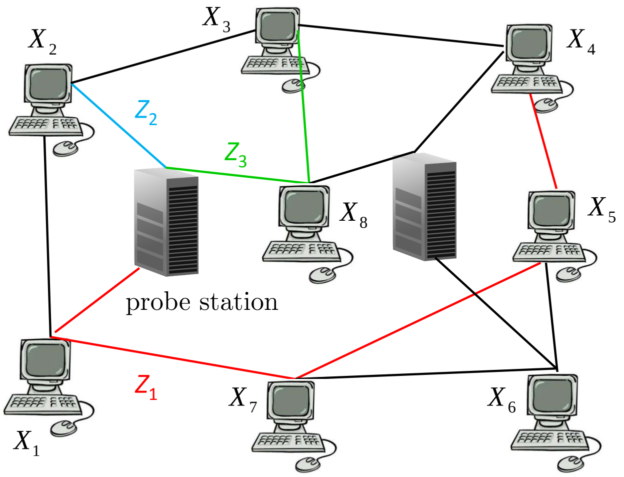


Figure 2.3: A toy example demonstrating a system of networked computers along with probe stations and probes.

In these networks, certain computers are designated as probe stations, which are instrumented to send out probes to test the response of the networked elements. However, the available set of probes is often very large, and hence it is desired to minimize the number of probes required to identify the faulty computers. In our experiments, we use networks generated using the BRITE (*Medina et al.*, 2001) and the INET (*Winick and Jamin*, 2002) generators, which simulate an Internet-like topology at the Autonomous systems level. To generate a BDG of computers and probes from these topologies, we used the approach described by *Rish et al.* (2005). Refer to Appendix C for a brief description on how these networks were generated.

CHAPTER III

Group Diagnosis

3.1 Introduction

In this chapter, we consider variations on the basic object identification framework that are motivated by the problem of toxic chemical identification, and are naturally applicable to other time-critical diagnostic tasks. In particular, we develop theoretical results and new algorithms for what might be described as group-based active diagnosis. The work in this chapter is based on *Bellala, Bhavnani and Scott* (2011b).

First, we consider the case where the object set Θ is partitioned into groups of objects, and it is only necessary to identify the group to which the unknown object belongs. For example, the appropriate response to a toxic chemical may only depend on the class of chemicals to which it belongs (pesticide, corrosive acid, etc.). As our experiments reveal, an active query selection algorithm designed to rapidly identify individual objects is not necessarily efficient for group identification.

Second, we consider the problem where the set Q of queries is partitioned into groups (respiratory symptoms, cardio symptoms, etc.). Instead of suggesting specific symptoms to the user, we design an algorithm that suggests a group of queries, and allows the user the freedom to input information on any query in that group. Although such a system will theoretically be less efficient, it is motivated by the fact that in a practical application, some symptoms will be easier for a given user to understand

and identify. Instead of suggesting a single symptom, which might seem “out of the blue” to the user, suggesting a query group will be less bewildering, and hence lead to a more efficient and accurate outcome. Our experiments demonstrate that the proposed algorithm based on query groups identifies objects in nearly as few queries as a fully active method.

Third, we apply our algorithm for group identification to the problem of object identification under persistent query noise. Persistent query noise occurs when the response to a query is in error, but cannot be re-sampled, as is often assumed in the literature. Such is the case when the presence or absence of a symptom is incorrectly determined, which is more likely in a stressful emergency response scenario. Experiments show our method offers significant gains over algorithms not designed for persistent query noise.

Our algorithms are derived in a common framework, and are based on reinterpretation of a standard object identification algorithm (the splitting algorithm, or generalized binary search) as a generalized form of Shannon-Fano coding. We first establish an exact formula for the expected number of queries required to identify an object using an arbitrary decision tree, and show that the splitting algorithm effectively performs a greedy, top-down optimization of this objective. We then extend this formula to the case of group identification and query groups, and develop analogous greedy algorithms. In the process, we provide a new interpretation of impurity-based decision tree induction for multi-class classification. We also develop a logarithmic approximation bound for group identification, using the notion of submodular functions.

Finally, we demonstrate the performance of our algorithms through experiments on synthetic data as well as the WISER database (version 4.21). WISER, which stands for **W**ireless **I**nformation **S**ystem for **E**mergency **R**esponders, is a decision support system developed by the National Library of Medicine (NLM) for first responders. This database describes the binary relationship between 298 toxic chemicals (corresponding to the number of distinguishable chemicals in this database) and 79 acute symptoms. The symptoms are grouped into 10 categories (e.g., neurological, cardio) as determined by NLM, and the chemicals are grouped into 16 categories (e.g., pesticides, corrosive acids) as determined by a toxicologist and a Hazmat expert.

3.1.1 Notation

We denote an object identification problem by a pair $(\mathbf{B}, \Pi)$ where Π denotes the prior probability distribution on the objects, i.e., $\pi_i = \Pr(\theta = \theta_i)$, and $\mathbf{B}$ is a binary matrix denoting the binary relation between the objects and the queries as described in Chapter II. We assume that the rows of $\mathbf{B}$ are distinct, i.e., we make the assumption of unique identifiability of every object in Θ. This is reasonable since objects that have similar query responses for all queries in Q, i.e., objects that are not distinguishable, can always be grouped into a single meta-object.

A decision tree T constructed on $(\mathbf{B}, \Pi)$ has a query from the set Q at each of its internal nodes with the leaf nodes terminating in the objects from the set Θ. At each internal node in the tree, the object set under consideration is divided into two subsets, corresponding to the objects that respond 0 and 1 to the query, respectively. For a decision tree with L leaves, the leaf nodes are indexed by the set $\mathcal{L} = \{1, \cdots, L\}$ and the internal nodes are indexed by the set $\mathcal{I} = \{L+1, \cdots, 2L-1\}$. At any internal node $a \in \mathcal{I}$, let $l(a), r(a)$ denote the "left" and "right" child nodes, where the set $\Theta_a \subseteq \Theta$ corresponds to the set of objects that reach node 'a', and the sets $\Theta_{l(a)} \subseteq \Theta_a, \Theta_{r(a)} \subseteq \Theta_a$ corresponds to the set of objects that respond 0 and 1 to the query at node 'a', respectively. We denote by $\pi_{\Theta_a} := \sum_{\{i:\theta_i \in \Theta_a\}} \pi_i$, the probability mass of the objects under consideration at any node 'a' in the tree. Also, at any node 'a', the set $Q_a \subseteq Q$ corresponds to the set of queries that have been performed along the path from the root node up to node 'a'.

We denote the Shannon entropy of a vector $\Pi = (\pi_1, \cdots, \pi_M)$ by $H(\Pi) := -\sum_i \pi_i \log_2 \pi_i$ and the Shannon entropy of a proportion $\pi \in [0,1]$ by $H(\pi) := -\pi \log_2 \pi - (1-\pi)\log_2(1-\pi)$, where we use the limit, $\lim\limits_{\pi \to 0} \pi \log_2 \pi = 0$ to define the limiting cases. Finally, given a tree T, we use the random variable $K(T)$ to denote the number of queries required to identify an unknown object θ or the group of an unknown object θ using the given tree.

3.2 Coding-Theoretic Interpretation of Object Identification

Before proceeding to the group-based settings, we first present an exact formula for the standard object identification problem. This result allows us to interpret the splitting algorithm or GBS as generalized Shannon-Fano coding. Furthermore, our proposed algorithms for group-based settings are based on generalizations of this result.

First, we define a parameter called the *reduction factor* on the binary matrix/tree combination that provides a useful quantification on the expected number of queries required to identify an unknown object.

Definition III.1. A *reduction factor* at any internal node 'a' in a decision tree is defined as $\rho_a = \max(\pi_{\Theta_{l(a)}}, \pi_{\Theta_{r(a)}})/\pi_{\Theta_a}$ and the *overall reduction factor* of a tree is defined as $\rho = \max_{a \in \mathcal{I}} \rho_a$.

Note from the above definition that $0.5 \leq \rho_a \leq \rho \leq 1$ and we describe a decision tree with $\rho = 0.5$ to be a perfectly balanced tree.

Given an object identification problem $(\mathbf{B}, \Pi)$, let $\mathcal{T}(\mathbf{B}, \Pi)$ denote the set of decision trees that can uniquely identify all the objects in the set Θ. For any decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi)$, let $\{\rho_a\}_{a \in \mathcal{I}}$ denote the set of reduction factors and let d_i denote the depth of object θ_i in the tree. Then, the expected number of queries required to identify an unknown object using the given tree is equal to

$$\mathbb{E}[K(T)] = \sum_{i=1}^{M} \Pr(\theta = \theta_i)\mathbb{E}[K(T)|\theta = \theta_i] = \sum_{i=1}^{M} \pi_i d_i.$$

Theorem III.2. *The expected number of queries required to identify an unknown object using a tree $T \in \mathcal{T}(\mathbf{B}, \Pi)$ with reduction factors $\{\rho_a\}_{a\in\mathcal{I}}$ is given by*

$$\begin{aligned} \mathbb{E}[K(T)] &= H(\Pi) + \sum_{a\in\mathcal{I}} \pi_{\Theta_a}[1 - H(\rho_a)] \qquad (3.1) \\ &= \frac{H(\Pi)}{\sum_{a\in\mathcal{I}} \tilde{\pi}_{\Theta_a} H(\rho_a)} \end{aligned}$$

where $\tilde{\pi}_{\Theta_a} := \frac{\pi_{\Theta_a}}{\sum_{r\in\mathcal{I}} \pi_{\Theta_r}}$.

Proof. The first equality is a special case of Theorem III.6 below. The second equality follows from the observation $\mathbb{E}[K(T)] = \sum_{i=1}^{M} \pi_i d_i = \sum_{a\in\mathcal{I}} \pi_{\Theta_a}$. Hence replacing π_{Θ_a} with $\tilde{\pi}_{\Theta_a} \cdot \mathbb{E}[K(T)]$ in the first equality leads to the result. □

In the second equality, the term $\sum_{a\in\mathcal{I}} \tilde{\pi}_{\Theta_a} H(\rho_a)$ denotes the average entropy of the reduction factors, weighted by the proportion of times each internal node 'a' is queried in the tree. This theorem re-iterates an earlier observation that the expected number of queries required to identify an unknown object using a tree constructed on $(\mathbf{B}, \Pi)$ (where the query set Q is not necessarily a *complete* set) is bounded below by its entropy $H(\Pi)$. It also follows from the above result that a tree attains this minimum value (i.e., $\mathbb{E}[K(T)] = H(\Pi)$) iff it is perfectly balanced, i.e., the overall reduction factor ρ of the tree is equal to 0.5.

From the first equality, the problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\min_{T\in\mathcal{T}(\mathbf{B},\Pi)} H(\Pi) + \sum_{a\in\mathcal{I}} \pi_{\Theta_a}[1 - H(\rho_a)]. \qquad (3.2)$$

Since Π is fixed, the optimization problem reduces to minimizing $\sum_{a\in\mathcal{I}} \pi_{\Theta_a}[1-H(\rho_a)]$ over the set of trees $\mathcal{T}(\mathbf{B},\Pi)$. Note that the reduction factor ρ_a depends on the query chosen at node 'a' in a tree T. As mentioned earlier, finding a global optimal solution for this optimization problem is NP-complete.

Instead, we may take a top down approach and minimize the objective function by minimizing the term $\pi_{\Theta_a}[1-H(\rho_a)]$ at each internal node, starting from the root node. Since π_{Θ_a} is independent of the query chosen at node 'a', this reduces to minimizing ρ_a (i.e., choosing a split as balanced as possible) at each internal node $a \in \mathcal{I}$. The algorithm can be summarized as shown below.

```
Generalized Binary Search (GBS)
Initialization : Let the leaf set consist of the root node
while some leaf node 'a' has $|\Theta_a| > 1$ do
    for each query $q \in Q \setminus Q_a$ do
        Find $\Theta_{l(a)}$ and $\Theta_{r(a)}$ produced by making a split with query $q$
        Compute the reduction factor $\rho_a$ produced by query $q$
    end
    Choose a query with the smallest reduction factor
    Form child nodes $l(a), r(a)$
end
```

Note that when the query set Q is *complete*, GBS is similar to Shannon-Fano coding (*Shannon*, 1948; *Fano*, 1961). The only difference is that in Shannon-Fano coding, for computational reasons, the queries are restricted to those that are based on thresholding the prior probabilities π_i.

Corollary III.3. *The standard splitting algorithm/GBS is a greedy algorithm to minimize the expected number of queries required to uniquely identify an object.*

Corollary III.4 below follows from Theorem III.2. It states that given a tree T with overall reduction factor $\rho < 1$, the average complexity of identifying an unknown object using this tree is $O(\log_2 M)$. Recently, *Nowak* (2008) showed there are geometric conditions (incoherence and neighborliness) that also bound the worst-case depth of the tree to be $O(\log_2 M)$, assuming a uniform prior on objects. These conditions imply that the reduction factors are close to $\frac{1}{2}$ except possibly near the very bottom of the tree where they could be close to 1. Because ρ_a could be close to 1 for deeper nodes, the upper bound on $\mathbb{E}[K(T)]$ based on the overall reduction factor ρ given below could be very loose in practice.

Corollary III.4. *The expected number of queries required to identify an unknown object using a tree T with overall reduction factor ρ constructed on $(\mathbf{B}, \Pi)$ is bounded above by*

$$\mathbb{E}[K(T)] \leq \frac{H(\Pi)}{H(\rho)} \leq \frac{\log_2 M}{H(\rho)}$$

Proof. Using the second equality in Theorem III.2, we get

$$\mathbb{E}[K(T)] = \frac{H(\Pi)}{\sum_{a \in \mathcal{I}} \tilde{\pi}_{\Theta_a} H(\rho_a)} \leq \frac{H(\Pi)}{H(\rho)} \leq \frac{\log_2 M}{H(\rho)}$$

where the first inequality follows from the definition of ρ, $\rho \geq \rho_a \geq 0.5, \forall a \in \mathcal{I}$ and the last inequality follows from the concavity of the entropy function. □

In the sections that follow, we show how Theorem III.2 and GBS may be generalized, leading to principled strategies for group identification, object identification with group queries and object identification with persistent noise.

3.3 Group Identification

We now move to the problem of group identification, where the goal is not to determine the unknown object $\theta \in \Theta$, rather the group to which the object belongs. Here, in addition to the binary matrix $\mathbf{B}$ and *a priori* probability distribution Π on the objects, the group labels for the objects are also provided, where the groups are assumed to be disjoint. Note that if the groups are overlapping, it can be reduced to the disjoint setting by finding the smallest partition of the objects such that the group labels are constant on each cell of the partition. Then, a group identification algorithm would identify precisely those groups to which the object belongs. For example, in toxic chemical identification, a first responder may only need to know whether a chemical is a pesticide, a corrosive acid, or both. Hence, it could be reasonable to reduce a group identification problem with overlapping groups to that of disjoint groups arising out of its partition. Thus, we devote our attention to the problem of group identification with disjoint groups.

We denote a group identification problem by $(\mathbf{B}, \Pi, \mathbf{y})$, where $\mathbf{y} = (y_1, \cdots, y_M)$ denotes the group labels of the objects, $y_i \in \{1, \cdots, m\}$. Let $\{\Theta^i\}_{i=1}^m$ be a partition of the object set Θ, where Θ^i denotes the set of objects in Θ that belong to group i. It is important to note here that the group identification problem cannot be simply reduced to an object identification problem with groups $\{\Theta^1, \cdots, \Theta^m\}$ as "meta-objects," since the objects within a group need not respond the same to each query. For example, consider the toy example shown in Figure 3.1 where the objects θ_1, θ_2 and θ_3 belonging to group 1 cannot be considered as one single meta-object as these objects respond differently to queries q_1 and q_3.

In this context, we also note that GBS can fail to find a good solution for a group identification problem as it does not take the group labels into consideration while choosing queries. Once again, consider the toy example shown in Figure 3.1 where just one query (query q_2) is sufficient to identify the group of an unknown object, whereas GBS requires 2 queries to identify the group when the unknown object is either θ_2 or θ_4, as shown in Figure 3.2. Hence, we develop a new strategy which accounts for the group labels when choosing the best query at each stage.

Note that when constructing a tree for group identification, a greedy, top-down algorithm terminates splitting when all the objects at the node belong to the same

	q_1	q_2	q_3	Group label, y
θ_1	0	1	1	1
θ_2	1	1	0	1
θ_3	0	1	0	1
θ_4	1	0	0	2

Figure 3.1: Toy Example 1

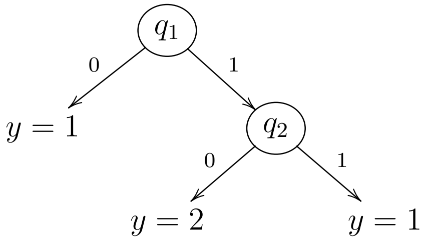


Figure 3.2: Decision tree constructed using GBS for group identification on toy example 1

group. Hence, a tree constructed in this fashion can have multiple objects ending in the same leaf node and multiple leaves ending in the same group.

For a tree with L leaves, we denote by $\mathcal{L}^i \subset \mathcal{L} = \{1, \cdots, L\}$ the set of leaves that terminate in group i. Similar to $\Theta^i \subseteq \Theta$, we denote by $\Theta^i_a \subseteq \Theta_a$ the set of objects that belong to group i at any internal node $a \in \mathcal{I}$ in the tree. Also, in addition to the reduction factors defined in Section 3.2, we define a new set of reduction factors called the group reduction factors at each internal node.

Definition III.5. The group reduction factor of group i at any internal node 'a' in a decision tree is defined as $\rho^i_a = \max(\pi_{\Theta^i_{l(a)}}, \pi_{\Theta^i_{r(a)}})/\pi_{\Theta^i_a}$.

Given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ denote the set of decision trees that can uniquely identify the groups of all objects in the set Θ. For any decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, let ρ_a denote the reduction factor and let $\{\rho^i_a\}_{i=1}^m$ denote the set of group reduction factors at each of its internal nodes. Also, let d_j denote the depth of leaf node $j \in \mathcal{L}$ in the tree. Then the expected number of queries required to identify the group of an unknown object using the given tree is equal to

$$\begin{aligned}\mathbb{E}[K(T)] &= \sum_{i=1}^{m} \Pr(\theta \in \Theta^i)\mathbb{E}[K(T)|\theta \in \Theta^i] \\ &= \sum_{i=1}^{m} \pi_{\Theta^i}\left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} d_j\right]\end{aligned}$$

Theorem III.6. *The expected number of queries required to identify the group of an object using a tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ with reduction factors $\{\rho_a\}_{a \in \mathcal{I}}$ and group reduction factors $\{\rho_a^i\}_{i=1}^m, \forall a \in \mathcal{I}$, is given by*

$$\mathbb{E}[K(T)] = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right] \tag{3.3}$$

where $\Pi_{\mathbf{y}}$ denotes the probability distribution of the object groups induced by the labels $\mathbf{y}$, i.e. $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^m})$.

Proof. Special case of Theorem III.12 below. □

The above theorem states that given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$, the expected number of queries required to identify the group of an unknown object is lower bounded by the entropy of the probability distribution of the groups. It also follows from the above result that this lower bound is achieved iff there exists a perfectly balanced tree (i.e. $\rho = 0.5$) with the group reduction factors equal to 1 at every internal node in the tree. Also, note that Theorem III.2 is a special case of this theorem where each group has size 1 leading to $\rho_a^i = 1$ for all groups at every internal node.

Using Theorem III.6, the problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$

can be formulated as the following optimization problem:

$$\min_{T \in \mathcal{T}(\mathbf{B},\Pi,\mathbf{y})} \textstyle\sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right]. \tag{3.4}$$

Note that here both the reduction factor ρ_a and the group reduction factors $\{\rho_a^i\}_{i=1}^m$ depend on the query chosen at node 'a'. Also, the above optimization problem being a generalized version of the optimization problem in (3.2) is NP-complete. Hence, we propose a suboptimal approach to solve the above optimization problem where we optimize the objective function locally instead of globally. We take a top-down approach and minimize the objective function by minimizing the term $\Delta_a := \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right]$ at each internal node, starting from the root node. The algorithm can be summarized as shown below. We refer to this algorithm as GISA (Group Identification Splitting Algorithm) or GGBS (Group Generalized Binary Search).

```
Group Identification Splitting Algorithm (GISA)
Initialization : Let the leaf set consist of the root node
while some leaf node 'a' has more than one group of objects do
    for each query $q_j \in Q \setminus Q_a$ do
        Compute $\{\rho_a^i\}_{i=1}^m$ and $\rho_a$ produced by making a split with query $q_j$
        Compute the cost $\Delta_a(j)$ of making a split with query $q_j$
    end
    Choose a query with the least cost $\Delta_a$ at node 'a'
    Form child nodes $l(a), r(a)$
end
```

Note that the objective function in this algorithm consists of two terms. The first term $[1 - H(\rho_a)]$ favors queries that evenly distribute the probability mass of the objects at node 'a' to its child nodes (regardless of the group) while the second term $\sum_i \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)$ favors queries that transfer an entire group of objects to one of its child nodes.

3.3.1 Relation to Impurity-based Decision Tree Induction

As a brief digression, in this section we show a connection between the above algorithm and impurity-based decision tree induction. In particular, we show that the above algorithm is equivalent to the decision tree splitting algorithm used in the C4.5 software package (*Quinlan*, 1993). Before establishing this result, we briefly review the multi-class classification setting where impurity-based decision tree induction is popularly used.

In the multi-class classification setting, the input is training data $\mathbf{x}_1, \cdots, \mathbf{x}_M$ sampled from some input space (with an underlying probability distribution) along with their class labels, $y_1, \cdots, y_M$ and the task is to construct a classifier with the least probability of misclassification. Decision tree classifiers are grown by maximizing an impurity-based objective function at every internal node to select the best classifier from a set of base classifiers. These base classifiers can vary from simple axis-orthogonal splits to more complex non-linear classifiers. The impurity-based objective function is

$$I(\Theta_a) - \left[\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} I(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} I(\Theta_{r(a)})\right], \tag{3.5}$$

which represents the decrease in impurity resulting from split 'a'. Here $I(\Theta_a)$ corresponds to the measure of impurity in the input subspace at node 'a' and π_{Θ_a} corresponds to the probability measure of the input subspace at node 'a'.

Among the various impurity functions suggested in the literature (*Kearns and Mansour*, 1995; *Takimoto and Maruoka*, 2003), the entropy measure used in the C4.5 software package (*Quinlan*, 1993) is popular. In the multi-class classification setting with m different class labels, this measure is given by

$$I(\Theta_a) = -\textstyle\sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \tag{3.6}$$

where $\pi_{\Theta_a}, \pi_{\Theta_a^i}$ are empirical probabilities based on the training data.

Similar to a group identification problem, the input here is a binary matrix $\mathbf{B}$ with b_{ij} denoting the binary label produced by base classifier j on training sample i, and a probability distribution Π on the training data along with their class labels $\mathbf{y}$. Specifically, the objects correspond to training data, queries to different base classifiers, and the object groups correspond to the different classes (two classes in case of a binary classification problem). However, unlike a group identification problem where the nodes in a tree are not terminated until all the objects belong to the same group, the leaf nodes here are allowed to contain some impurity in order to avoid overfitting. The following result extends Theorem III.6 to the case of impure leaf nodes.

Theorem III.7. *The expected depth of a leaf node in a decision tree classifier $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ with reduction factors $\{\rho_a\}_{a \in \mathcal{I}}$ and class reduction factors $\{\rho_a^i\}_{i=1}^m, \forall a \in \mathcal{I}$, is given by*

$$\mathbb{E}[K(T)] = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i) \right] - \sum_{a \in \mathcal{L}} \pi_{\Theta_a} I(\Theta_a) \qquad (3.7)$$

where $\Pi_{\mathbf{y}}$ denotes the probability distribution of the classes induced by the class labels $\mathbf{y}$, i.e., $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^m})$ and $I(\Theta_a)$ denotes the impurity in leaf node 'a' given by (3.6).

Proof. The proof is given in Appendix A. □

The only difference compared to Theorem III.6 is the last term, which corresponds to the average impurity in the leaf nodes.

Theorem III.8. *At every internal node in a tree, minimizing the objective function $\Delta_a := 1 - H(\rho_a) + \sum_{i=1}^m \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)$ is equivalent to maximizing $I(\Theta_a) - \Big[\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} I(\Theta_{l(a)})$ $+ \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} I(\Theta_{r(a)})\Big]$ with entropy measure as the impurity function.*

Proof. The proof is given in Appendix A. □

Therefore, greedy optimization of (3.7) at internal nodes corresponds to greedy optimization of impurity. Also, note that optimizing (3.7) at a leaf assigns the majority vote class label. Therefore, we conclude that impurity-based decision tree induction with entropy as the impurity measure amounts to a greedy optimization of the expected depth of a leaf node in the tree. Also, Theorem III.7 allows us to interpret impurity based splitting algorithms for multi-class decision trees in terms of reduction factors, which also appears to be a new insight.

3.3.2 A Near-optimal Algorithm

As mentioned in Chapter II, the splitting algorithm or GBS has been shown to be near-optimal with a logarithmic approximation ratio (*Dasgupta*, 2004; *Nowak*, 2008; *Golovin and Krause*, 2010), i.e.,

$$\mathbb{E}[K(\widehat{T})] \leq O\left(\ln \frac{1}{\pi_{\min}}\right) \mathbb{E}[K(T^*)],$$

where $\pi_{\min} := \min_i \pi_i$ is the minimum prior probability of any object, $\widehat{T}$ is a greedy tree constructed using GBS and T^* is an optimal tree for the given problem.

Recently, *Golovin and Krause* (2010) introduced the notion of adaptive submodularity and strong adaptive monotonicity (refer to Appendix A), and showed that a greedy optimization algorithm with these properties can be near-optimal and achieve a logarithmic approximation ratio, with GBS being a specific instance of this class. Unfortunately, the objective function in GISA, i.e.,

$$H(\rho_a) - \sum_{i=1}^{m} \frac{\pi_a^i}{\pi_a} H(\rho_a^i) \tag{3.8}$$

does not satisfy these properties. We now present a modified version of GISA that can be shown to be adaptive submodular and strong adaptive monotone, and hence can achieve a logarithmic approximation to the optimal solution.

The modified algorithm is to construct a top-down, greedy decision tree where at each internal node, a query that maximizes

$$\pi_{l(a)}\pi_{r(a)} - \sum_{i=1}^{m} \frac{\pi_a^i}{\pi_a} \pi_{l(a)}^i \pi_{r(a)}^i \tag{3.9}$$

is chosen. Essentially, the binary entropy terms $H(\rho_a)$ and $H(\rho_a^i)$ in (3.8) are approximated by the weighted Gini indices, $\pi_a^2(\rho_a(1-\rho_a))$ and $(\pi_a^i)^2\,(\rho_a^i(1-\rho_a^i))$, respectively. Note that in the special case where each group is of size 1, the query selection criterion in (3.9) reduces to $\pi_{l(a)}\pi_{r(a)}$, thereby reducing modified GISA to the standard splitting algorithm.

Given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$, recall that $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ denotes the set of all possible trees that can uniquely identify the group of any object from the set Θ. Then, let T^* denote a tree with the least expected depth, i.e.,

$$T^* \in \operatorname*{arg\,min}_{T \in \mathcal{T}(\mathbf{B},\Pi,\mathbf{y})} \mathbb{E}[K(T)],$$

and let $\widehat{T}$ denote a tree constructed using modified GISA. The following theorem states that the expected depth of $\widehat{T}$ is logarithmically close to that of an optimal tree. In effect, this also provides a near-optimal algorithm for decision tree construction in the classification setting.

Theorem III.9. *Let* $(\mathbf{B}, \Pi, \mathbf{y})$ *denote a group identification problem. For a greedy decision tree* $\widehat{T}$ *constructed using modified GISA, it holds that*

$$\mathbb{E}[K(\widehat{T})] \le \left(2 \ln\left(\frac{1}{\sqrt{3}\pi_{\min}}\right) + 1\right) \mathbb{E}[K(T^*)], \tag{3.10}$$

where $\pi_{\min} := \min\{\pi \in \Pi : \pi > 0\}$ *is the minimum prior probability of any object.*

Proof. The proof is given in Appendix A. □

In addition, if the query costs are unequal, the query selection criterion in modified GISA can be changed to $\arg\max_{q \notin Q_a} \Delta_a(q)/c(q)$, where $\Delta_a(q)$ is as defined in (3.9), and $c(q)$ is the cost of obtaining the response to query q. This simple heuristic can been shown to retain the near-optimal property, i.e.,

$$c(\widehat{T}) \leq \left(2 \ln \left(\frac{1}{\sqrt{3}\pi_{\min}}\right) + 1\right) c(T^*),$$

where $\widehat{T}$ is a greedy tree constructed using the above heuristic, and T^* is a tree with minimum expected cost. The cost of a tree T is defined as $c(T) := \mathbb{E}_\theta[c(T, \theta)]$, where $c(T, \theta_i)$ is the total cost of the queries made along the path from the root node to the leaf node ending in object θ_i.

Golovin et al. (2010) simultaneously studied the problem of group identification, and, like us, used it in the context of object identification with persistent noise (refer Section 3.6). They proposed an extension of the algorithm by *Dasgupta* (2006) for group identification, and showed a logarithmic approximation similar to us. However, their result holds only when the priors π_i are rational. In addition, the bound achieved by modified GISA is marginally tighter than theirs.

3.4 Object Identification under Group Queries

In this section, we return to the problem of object identification. The input is a binary matrix $\mathbf{B}$ denoting the relationship between M objects and N queries, where the queries are grouped *a priori* into n disjoint categories, along with the *a priori* probability distribution Π on the objects. However, unlike the decision trees constructed in the previous two sections where the end user (e.g., a first responder) has to go through a fixed set of questions as dictated by the decision tree, here, the user

	Q^1		Q^2	
	0.5	0.5	0.9	0.1
	q_1	q_2	q_3	q_4
θ_1	0	1	1	0
θ_2	1	0	1	1
θ_3	1	1	0	1

Figure 3.3: Toy Example 2

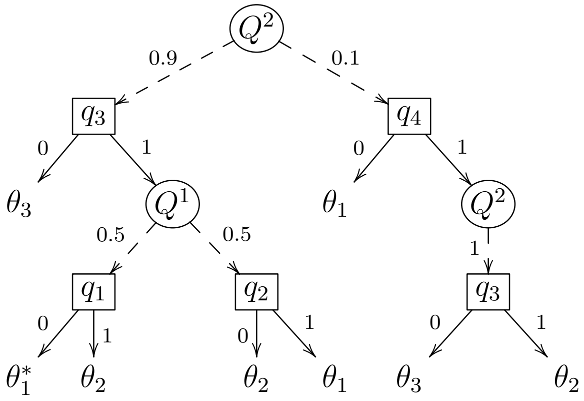


Figure 3.4: Decision tree constructed on toy example 2 for object identification under group queries

is offered more flexibility in choosing the questions at each stage. More specifically, the decision tree suggests a query group from the n groups instead of a single query at each stage, and the user can choose a query to answer from the suggested query group.

A decision tree constructed with a group of queries at each stage has multiple branches at each internal node, corresponding to the size of the query group. Hence, a tree constructed in this fashion has multiple leaves ending in the same object. While traversing this decision tree, the user chooses the path at each internal node by selecting the query to answer from the given list of queries. Figure 3.4 demonstrates a decision tree constructed in this fashion for the toy example shown in Figure 3.3. The circled nodes correspond to the internal nodes, where each internal node is associated with a query group. The numbers associated with a dashed edge correspond to the probability that the user will choose that path over the others. The probability of reaching a node $a \in \mathcal{I}$ in the tree given $\theta \in \Theta_a$ is given by the product of the probabilities on the dashed edges along the path from the root node to that node, for example, the probability of reaching leaf node θ_1^* given $\theta = \theta_1$ in Figure 3.4 is 0.45. The problem now is to select the query categories that will identify the object most efficiently, on average.

In addition to the terminology defined in Sections 3.1.1 and 3.2, we also define $\mathbf{z} = (z_1, \cdots, z_N)$ to be the group labels of the queries, where $z_j \in \{1, \cdots, n\}, \forall j = 1, \cdots, N$. Let $\{Q^i\}_{i=1}^n$ be a partition of the query set Q, where Q^i denotes the set of queries in Q that belong to group i. Similarly, at any node 'a' in a tree, let Q_a^i and $\overline{Q_a^i}$ denote the set of queries in Q_a and $Q \setminus Q_a$ that belong to group i respectively. Let $p_i(q)$ be the *a priori* probability of the user selecting query $q \in Q^i$ at any node with query group i in the tree, where $\sum_{q \in Q^i} p_i(q) = 1$. In addition, at any node 'a' in the tree, the function $p_i(q) = 0, \forall q \in Q_a^i$, since the user would not choose a query which has already been answered, in which case $p_i(q)$ is renormalized. In our experiments we take $p_i(q)$ to be uniform on $\overline{Q_a^i}$. Finally, let $z_a \in \{1, \cdots, n\}$ denote the query group selected at an internal node 'a' in the tree and let $\tilde{p}_a$ denote the probability of reaching that node given $\theta \in \Theta_a$.

We denote an object identification problem with query groups by $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$. Given $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$ denote the set of decision trees that can uniquely identify all the objects in the set Θ with query groups at each internal node. For a decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, let $\{\rho_a(q)\}_{q \in Q^{z_a}}$ denote the reduction factors of all the queries in the query group at each internal node $a \in \mathcal{I}$ in the tree, where the reduction factors are treated as functions with input being a query.

Also, for a tree with L leaves, let $\mathcal{L}^i \subset \mathcal{L} = \{1, \cdots, L\}$ denote the set of leaves terminating in object θ_i and let d_j denote the depth of leaf node $j \in \mathcal{L}$. Then, the expected number of queries required to identify the unknown object using the given tree is equal to

$$\begin{aligned}
\mathbb{E}[K(T)] &= \sum_{i=1}^{M} \Pr(\theta = \theta_i) \mathbb{E}[K(T) | \theta = \theta_i] \\
&= \sum_{i=1}^{M} \pi_i \left[\sum_{j \in \mathcal{L}^i} \tilde{p}_j d_j \right]
\end{aligned}$$

Theorem III.10. *The expected number of queries required to identify an object using a tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$ is given by*

$$\mathbb{E}[K(T)] \quad = \quad H(\Pi) + \sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \left[1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) H(\rho_a(q)) \right] \tag{3.11}$$

Proof. Special case of Theorem III.12 below. □

Note from the above theorem, that given an object identification problem with group queries $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, the expected number of queries required to identify an object is lower bounded by its entropy $H(\Pi)$. Also, this lower bound can be achieved iff the reduction factors of all the queries in a query group at each internal node of the tree is equal to 0.5. In fact, Theorem III.2 is a special case of the above theorem where each query group has just one query.

Given $(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})$, the problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\min_{T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{z}, \mathbf{p})} \textstyle\sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \left[1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) H(\rho_a(q)) \right].$$

Note that here the reduction factors $\rho_a(q), \forall q \in Q^{z_a}$ and the prior probability function $p_{z_a}(q)$ depends on the query group $z_a \in \{1, \cdots, n\}$ chosen at node 'a' in the tree. The above optimization problem being a generalized version of the optimization problem in (3.2) is NP-complete. A greedy top-down local optimization of the above objective function yields a suboptimal solution where we choose a query group that minimizes the term $\Delta_a(j) := \left[1 - \sum_{q \in Q^j} p_j(q) H(\rho_a(q)) \right]$ at each internal node, starting from the root node. We refer to this algorithm summarized below, as GQSA (Group Queries Splitting Algorithm).

Remark III.11. In this section and the one following, we assume that the query groups are disjoint only for the sake of simplicity. However, we do not need this assumption

```
Group Queries Splitting Algorithm (GQSA)
Initialization : Let the leaf set consist of the root node
while some leaf node 'a' has $|\Theta_a| > 1$ do
    for each query group with $\left|\overline{Q_a^j}\right| \geq 1$ do
        Compute the prior probabilities of selecting queries within a group $p_j(q), \forall q \in Q^j$ at node 'a'
        Compute the reduction factors for all the queries in the query group $\{\rho_a(q)\}_{q \in Q^j}$
        Compute the cost $\Delta_a(j)$ of using query group $j$ at node 'a'
    end
    Choose a query group $j$ with the least cost $\Delta_a(j)$ at node 'a'
    Form the left and the right child nodes for all queries with $p_j(q) > 0$ in the query group
end
```

for the results in Theorem III.10, and Theorem III.12 in the next section, to hold. Similarly, we assume that the prior probability of choosing a query from a query group depends only on the group membership. However, one could use a more complex prior distribution that not only depends on the group membership, but also on the previous queries and their responses. The results in Theorems III.10 and III.12 do not change by these generalizations, as long as the prior distribution is normalized and sums to 1 at each internal node in the tree. This can be readily observed from the proof of Theorem III.12 in Appendix A.

3.5 Group Identification under Group Queries

For the sake of completion, we consider here the problem of identifying the group of an unknown object $\theta \in \Theta$ under group queries. The input is a binary matrix $\mathbf{B}$ denoting the relationship between M objects and N queries, where the objects are grouped into m groups and the queries are grouped into n groups. The task is to identify the group of an unknown object through as few queries from Q as possible where, at each stage, the user is offered a query group from which a query is chosen.

As noted in Section 3.3, a decision tree constructed for group identification can have multiple objects terminating in the same leaf node. Also, a decision tree constructed for group identification with a query group at each internal node has multiple leaves terminating in the same group. Hence a decision tree constructed in this section can have multiple objects terminating in the same leaf node and multiple leaves terminating in the same group. Also, we use most of the terminology defined in Sections 3.3 and 3.4 here.

We denote a group identification problem with query groups by $(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$ where $\mathbf{y} = (y_1, \cdots, y_M)$ denotes the group labels on the objects, $\mathbf{z} = (z_1, \cdots, z_N)$ denotes the group labels on the queries and $\mathbf{p} = (p_1(q), \cdots, p_n(q))$ denotes the *a priori* probability functions of selecting queries within query groups. Given a group identification problem under group queries $(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$ denote the set of decision trees that can uniquely identify the groups of all objects in the set Θ with query groups at each internal node. For any decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$, let $\{\rho_a(q)\}_{q \in Q^{z_a}}$ denote the reduction factor set and let $\{\{\rho_a^i(q)\}_{i=1}^m\}_{q \in Q^{z_a}}$ denote the group reduction factor sets at each internal node $a \in \mathcal{I}$ in the tree, where $z_a \in \{1, \cdots, n\}$ denotes the query group selected at that node.

Also, for a tree with L leaves, let $\mathcal{L}^i \subset \mathcal{L} = \{1, \cdots, L\}$ denote the set of leaves terminating in object group i and let $d_j, \tilde{p}_j$ denote the depth of leaf node $j \in \mathcal{L}$ and the probability of reaching that node given $\theta \in \Theta_j$, respectively. Then, the expected number of queries required to identify the group of an unknown object using the given tree is equal to

$$\begin{aligned} \mathbb{E}[K(T)] &= \sum_{i=1}^{m} \Pr(\theta \in \Theta^i) \mathbb{E}[K(T) | \theta \in \Theta^i] \\ &= \sum_{i=1}^{m} \pi_{\Theta^i} \left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} \tilde{p}_j d_j \right] \end{aligned}$$

Theorem III.12. *The expected number of queries required to identify the group of an unknown object using a tree* $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$ *is given by*

$$\begin{aligned} \mathbb{E}[K(T)] \quad = \quad & H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \Bigg\{ 1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) \\ & \left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q)) \right] \Bigg\} \end{aligned} \tag{3.12}$$

where $\Pi_{\mathbf{y}}$ *denotes the probability distribution of the object groups induced by the labels* $\mathbf{y}$*, i.e.* $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^m})$

Proof. The proof is given in Appendix A. □

Note that Theorems III.2, III.6 and III.10 are special cases of the above theorem. This theorem states that, given a group identification problem under group queries $(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})$, the expected number of queries required to identify the group of an object is lower bounded by the entropy of the probability distribution of the object groups $H(\Pi_{\mathbf{y}})$. It also follows from the above theorem that this lower bound can be achieved iff the reduction factors and the group reduction factors of all the queries in a query group at each internal node are equal to 0.5 and 1 respectively.

The problem of finding a decision tree with minimum $\mathbb{E}[K(T)]$ can be formulated as the following optimization problem:

$$\min_{T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y}, \mathbf{z}, \mathbf{p})} \quad \sum_{a \in \mathcal{I}} \tilde{p}_a \pi_{\Theta_a} \left\{ 1 - \sum_{q \in Q^{z_a}} p_{z_a}(q) \left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q)) \right] \right\}.$$

Note that here the reduction factors $\{\rho_a(q)\}_{q \in Q^{z_a}}$, the group reduction factors $\{\rho_a^i(q)\}_{q \in Q^{z_a}}$ for all $i = 1, \cdots, m$, and the prior probability function $p_{z_a}(q)$ depends on the query group $z_a \in \{1, \cdots, n\}$ chosen at node 'a' in the tree. Once again, the above optimization problem being a generalized version of the optimization problem in (3.2) is NP-complete. A greedy top-down optimization of the above objective

```
Group Identification under Group Queries Splitting Algorithm (GIGQSA)
Initialization : Let the leaf set consist of the root node
while some leaf node 'a' has more than one group of objects do
    for each query group with $\left|\overline{Q_a^j}\right| \geq 1$ do
        Compute the prior probabilities of selecting queries within a group, $p_j(q), \forall q \in Q^j$ at node 'a'
        Compute the reduction factors for all the queries in the query group $\{\rho_a(q)\}_{q \in Q^j}$
        Compute the group reduction factors for all the queries in the query group $\{\rho_a^i(q)\}_{q \in Q^j}$, $\forall i = 1, \cdots, m$
        Compute the cost $\Delta_a(j)$ of using query group $j$ at node 'a'
    end
    Choose a query group $j$ with the least cost $\Delta_a(j)$ at node 'a'
    Form the left and the right child nodes for all queries with $p_j(q) > 0$ in the query group
end
```

function yields a suboptimal solution where we choose a query group that minimizes the term $\Delta_a(j) := 1 - \sum_{q \in Q^j} p_j(q) \left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q)) \right]$ at each internal node, starting from the root node. We refer to this algorithm summarized above, as GIGQSA (Group Identification under Group Queries Splitting Algorithm).

3.6 Object Identification under Persistent Noise

We now consider the problem of rapidly identifying an unknown object $\theta \in \Theta$ in the presence of persistent query noise, and relate this problem to group identification. Query noise refers to errors in the query responses, i.e., the observed query response is different from the true response of the unknown object. For example, a victim of toxic chemical exposure may not report a symptom because of a delayed onset of that symptom. Unlike the noise model often assumed in the literature, where repeated querying results in independent realizations of the noise, persistent query noise is a more stringent noise model where repeated queries results in the same response.

	q_1	q_2	q_3	Π
prone to error	×	✓	✓	
θ_1	0	0	0	$\frac{1}{4}$
θ_2	1	1	1	$\frac{3}{4}$

(a)

	q_1	q_2	q_3	$\widetilde{\Pi}_1(p = 0.5)$	$\widetilde{\Pi}_2(p = 0.25)$
Θ^1	0	0	0	$\frac{1}{12}$	$\frac{3}{20}$
	1	0	0	0	0
	0	1	0	$\frac{1}{12}$	$\frac{1}{20}$
	0	0	1	$\frac{1}{12}$	$\frac{1}{20}$
Θ^2	1	1	1	$\frac{1}{4}$	$\frac{9}{20}$
	0	1	1	0	0
	1	0	1	$\frac{1}{4}$	$\frac{3}{20}$
	1	1	0	$\frac{1}{4}$	$\frac{3}{20}$

(b)

Figure 3.5: For the toy example shown in (a) consisting of 2 objects and 3 queries with an $\epsilon = 1$, (b) demonstrates the construction of matrix $\widetilde{\mathbf{B}}$. The probability distribution of the objects in $\widetilde{\mathbf{B}}$ are generated using the noise model described in Section 3.6.1, where only queries q_2 and q_3 are assumed to be prone to error.

Before we address this problem, we need to introduce some additional notation. Given an object identification problem $(\mathbf{B}, \Pi)$, let δ denote the minimum Hamming distance between any two rows of the matrix $\mathbf{B}$. Also, we refer to the bit string consisting of observed query responses as an input string. The input string can differ from the true bit string (corresponding to the row vector of the true object in matrix $\mathbf{B}$) due to persistent query noise. However, we further assume that the number of query responses in error cannot exceed $\epsilon := \lfloor \frac{\delta - 1}{2} \rfloor$. Note that in the persistent noise model, this assumption is required for unique identification of the unknown object. Given this noise setting, the goal of object identification under persistent noise is to uniquely identify the unknown object θ using as few queries as possible, where the responses to queries can be in error.

This problem can be posed as a group identification problem as follows: Given an object identification problem $(\mathbf{B}, \Pi)$ with M objects and N queries that is susceptible to ϵ errors, create $(\widetilde{\mathbf{B}}, \widetilde{\Pi})$ with M groups of objects and N queries, where each object group in this new matrix is formed by considering all possible bit strings that differ from the original bit string in at most ϵ positions, i.e., the size of each object group in $\widetilde{\mathbf{B}}$ is $\sum_{e=0}^{\epsilon} \binom{N}{e}$. Figure 3.5(b) demonstrates construction of $\widetilde{\mathbf{B}}$ for the toy example shown in Figure 3.5(a) consisting of 2 objects and 3 queries with an $\epsilon = 1$.

Each bit string in the object set Θ^i of $\widetilde{\mathbf{B}}$ corresponds to one of the possible input strings when the true object is θ_i and at most ϵ errors occur. Also note that, by definition of ϵ, no two bit strings in the matrix $\widetilde{\mathbf{B}}$ can be the same. Thus, the problem of rapidly identifying an unknown object θ from $(\mathbf{B}, \Pi)$ in the presence of at most ϵ persistent errors, reduces to the problem of identifying the group of the unknown object from $(\widetilde{\mathbf{B}}, \widetilde{\Pi})$. The probability distribution $\widetilde{\Pi}$ of the bit strings in $\widetilde{\mathbf{B}}$ depends on the prior Π and the error model. In the following section, we describe one specific error model that arises commonly in applications such as active learning, image processing and computer vision, and demonstrate the computation of $\widetilde{\Pi}$ under that error model.

Given that this problem can be reduced to a group identification problem, the unknown object can be rapidly identified in the presence of persistent query noise using any group identification algorithm including GISA and modified GISA. In addition, the near-optimal property of modified GISA guarantees that the expected number of queries required to identify an unknown object under persistent noise is logarithmically close to that of an optimal algorithm, as stated in the result below.

Corollary III.13. *Let* $(\mathbf{B}, \Pi)$ *denote an object identification problem that is susceptible to* ϵ *persistent errors. Let* $\widehat{K}$ *denote the expected number of queries required to identify an unknown object under persistent noise using modified GISA, and let* K^* *denote the expected number of queries required by an optimal algorithm. Then it holds* *that*

$$\widehat{K} \leq \left(2\ln\left(\frac{1}{\sqrt{3}\widetilde{\pi}_{\min}}\right)+1\right)K^*,$$

where $\widetilde{\pi}_{\min} = \min\{\widetilde{\pi} \in \widetilde{\Pi} : \widetilde{\pi} > 0\}$.

Proof. The result follows from Theorem III.9. □

3.6.1 Constant Noise Rate

We now consider a noise model that has been used in the context of pool-based active learning with a faulty oracle (*Nowak*, 2009; *Hanneke*, 2007), experimental design (*Rényi*, 1961), computer vision, and image processing (*Korostelev and Kim*, 2000), where the responses to some queries are assumed to be randomly flipped.

We will describe a general version of this noise model. Given N queries, consider the case where a fraction ν of them are prone to error. The query response to each of these νN queries can be in error with a probability $0 \leq p \leq 0.5$, where the errors occur independently. Then, the probability of e errors occurring is given by

$$\Pr(e \ \text{ errors}) = \frac{\binom{N\nu}{e}p^e(1-p)^{N\nu-e}}{\sum_{e'=0}^{\epsilon'}\binom{N\nu}{e'}p^{e'}(1-p)^{N\nu-e'}}, \quad 0 \leq e \leq \epsilon'$$

where $\epsilon' := \min(\epsilon, N\nu)$ denotes the maximum number of persistent errors that could occur. Note that this probability model corresponds to a truncated binomial distribution.

Given an object identification problem $(\mathbf{B}, \Pi)$ that is susceptible to ϵ errors, let $\widetilde{\mathbf{B}}$ denote the extended binary matrix constructed as described in Section 3.6. The probability distribution $\widetilde{\Pi}$ of the objects in $\widetilde{\mathbf{B}}$ can be computed as follows. For an object belonging to group i in $\widetilde{\mathbf{B}}$, if its response to a query that is not prone to error differs from the true response of object θ_i in $\mathbf{B}$, then the probability $\widetilde{\pi}$ of that object in $\widetilde{\mathbf{B}}$ is 0. On the other hand, if its response differs in $e \leq \epsilon'$ queries that are prone to error, then its probability is given by

$$\frac{p^e(1-p)^{N\nu-e}}{\sum_{e'=0}^{\epsilon'}\binom{N\nu}{e'}p^{e'}(1-p)^{N\nu-e'}}\ \pi_i.$$

Figure 3.5(b) shows the probability distribution of the objects in $\widetilde{\mathbf{B}}$ using the probability model described above with $p = 0.5$ ($\widetilde{\Pi}_1$) and $p = 0.25$ ($\widetilde{\Pi}_2$) for the toy example shown in Figure 3.5(a) where only queries q_2 and q_3 are prone to error.

However, one possible concern with this approach for object identification under persistent noise could be a memory related issue of explicitly maintaining the matrix $\widetilde{\mathbf{B}}$ due to the combinatorial explosion in its size. Interestingly, for the noise model described here, the relevant quantities for query selection in GBS, GISA and modified GISA (i.e., the reduction factors) can be efficiently computed without explicitly constructing the matrix $\widetilde{\mathbf{B}}$, described in detail in Appendix A.

3.7 Experimental Evaluation

We perform three sets of experiments, demonstrating our algorithms for group identification, object identification using query groups, and object identification with persistent noise. In each case, we compare the performances of the proposed algorithms to standard algorithms such as the splitting algorithm, using synthetic data as well as a real dataset, the WISER database. The WISER database is a toxic chemical database describing the binary relationship between 298 toxic chemicals and 79 acute symptoms. The symptoms are grouped into 10 categories (e.g., neurological, cardio) as determined by NLM, and the chemicals are grouped into 16 categories (e.g., pesticides, corrosive acids) as determined by a toxicologist and a Hazmat expert.

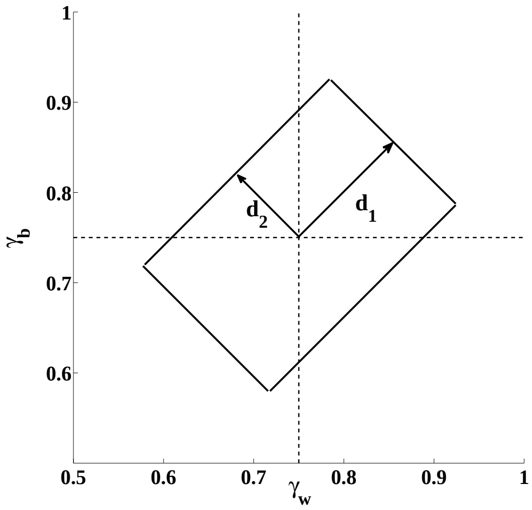


Figure 3.6: Random data model - The query parameters $(\gamma_w(q), \gamma_b(q))$ are restricted to lie in the rectangular space

3.7.1 Group Identification

Here, we consider a group identification problem $(\mathbf{B}, \Pi)$ where the objects are grouped into m groups given by $\mathbf{y} = (y_1, \cdots, y_M)$, $y_i \in \{1, \cdots, m\}$, with the task of identifying the group of an unknown object from the object set Θ through as few queries from Q as possible. First, we consider random datasets generated using a random data model and compare the performances of GBS, GISA and modified GISA for group identification in these random datasets. Then, we compare the performance of these algorithms on the WISER database. In both these experiments, we assume a uniform *a priori* probability distribution on the objects.

3.7.1.1 Random Datasets

We consider random datasets of the same size as the WISER database, with 298 objects and 79 queries where the objects are grouped into 16 classes with the same group sizes as that in the WISER database. We associate each query in a random dataset with two parameters, $\gamma_w \in [0.5, 1]$ which reflects the correlation of the object responses *within* a group, and $\gamma_b \in [0.5, 1]$ which captures the correlation of the

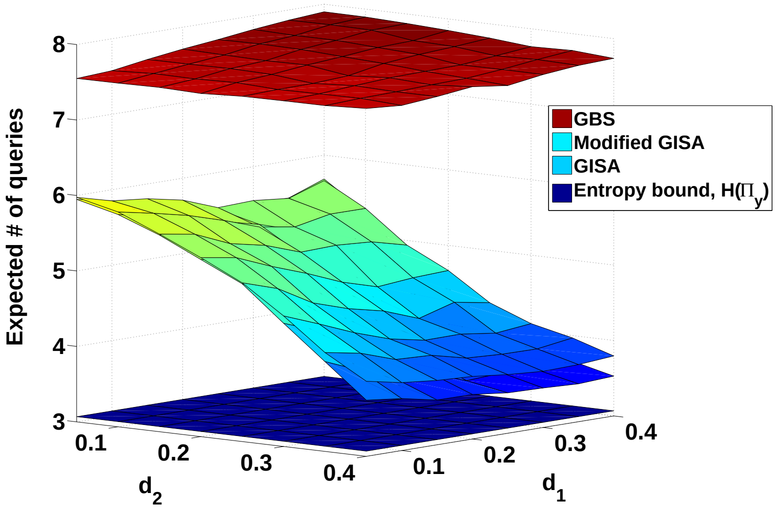


Figure 3.7: Expected number of queries required to identify the group of an object using GBS, GISA and modified GISA on random datasets generated using the proposed random data model. Note that GISA and modified GISA achieve almost similar performance on these datasets, with GISA performing slightly better than modified GISA.

object responses *between* groups. When γ_w is close to 0.5, each object within a group is equally likely to exhibit 0 or 1 as its response to the query, whereas, when γ_w is close to 1, most of the objects within a group are highly likely to exhibit the same response to the query. Similarly, when γ_b is close to 0.5, each group is equally likely to exhibit 0 or 1 as its response to the query, where a group response corresponds to the majority vote of the object responses within a group, while, as γ_b tends to 1, most of the groups are highly likely to exhibit the same response.

Given a (γ_w, γ_b) pair for a query in a random dataset, the object responses for that query are created as follows

1. Generate a Bernoulli random variable, x

2. For each group $i \in \{1, \cdots, m\}$, assign a binary label b_i, where $b_i = x$ with probability γ_b

3. For each object in group i, assign b_i as the object response with probability γ_w

Given the correlation parameters $(\gamma_w(q), \gamma_b(q)) \in [0.5, 1]^2, \forall q \in Q$, a random dataset

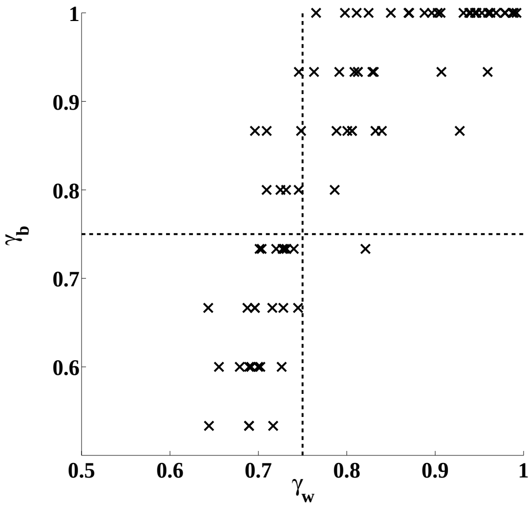


Figure 3.8: Scatter plot of the query parameters in the WISER database

can be created by following the above procedure for each query. Conversely, we describe in Section 3.7.1.2 on how to estimate these parameters for a given dataset.

Figure 3.7 compares the mean $\mathbb{E}[K(T)]$ for GBS, GISA and modified GISA in 100 randomly generated datasets (for each value of d_1 and d_2), where the random datasets are created such that the query parameters are uniformly distributed in the rectangular space governed by d_1, d_2 as shown in Figure 3.6. This demonstrates the improved performance of GISA and modified GISA over GBS in group identification. Especially, note that $\mathbb{E}[K(T)]$ tends close to the entropy bound $H(\Pi_{\mathbf{y}})$ using both GISA and modified GISA as d_2 increases.

This is due to the increment in the number of queries in the fourth quadrant of the parameter space as d_2 increases. Specifically, as the correlation parameters γ_w, γ_b tends to 1 and 0.5 respectively, choosing that query eliminates approximately half the groups with each group being either completely eliminated or completely included, i.e. the group reduction factors tend to 1 for these queries. Such queries are preferable in group identification with both GISA and modified GISA being specifically designed to search for those queries leading to their strikingly improved performance over GBS as d_2 increases.

Algorithm	$\mathbb{E}[K(T)]$
modified GISA	7.291 ± 0.001
GISA	7.792 ± 0.001
GBS	7.948 ± 0.003
Random Search	16.328 ± 0.177

Table 3.1: Expected number of queries required to identify the group of an object in the WISER database

3.7.1.2 WISER Database

Table 3.1 compares the expected number of queries required to identify the group of an unknown object in the WISER database using GISA, modified GISA, GBS and random search, where the group entropy in the WISER database is given by $H(\Pi_{\mathbf{y}}) = 3.068$. The table reports the 95% symmetric confidence intervals based on random trails, where the randomness in GISA, modified GISA and GBS is due to the presence of multiple best splits at each internal node.

However, the improvement of both GISA and modified GISA over GBS on WISER is less than was observed for many of the random datasets discussed above. To understand this, we developed a method to estimate the correlation parameters of the queries for a given dataset $\mathbf{B}$. For each query in the dataset, the correlation parameters can be estimated as follows

1. For every group $i \in \{1, \cdots, m\}$, let b_i denote the group response given by the majority vote of object responses in the group and let $\widehat{\gamma_w^i}$ denote the fraction of objects in the group with similar response as b_i

2. Denote by a binary variable x, the majority vote of the group responses $\mathbf{b} = [b_1, \cdots, b_m]$

3. Then, $\widehat{\gamma_b}$ is given by the fraction of groups with similar response as x, and $\widehat{\gamma_w} = \frac{1}{m} \sum_i \widehat{\gamma_w^i}$

Now, we use the above procedure to estimate the query parameters for all queries in the WISER database, shown in Figure 3.8. Note from this figure that there is just one query in the fourth quadrant of the parameter space and there are no queries with γ_w close to 1 and γ_b close to 0.5. In words, chemicals in the same group tend to behave differently and chemicals in different groups tend to exhibit similar response to the symptoms. This is a manifestation of the non-specificity of the symptoms in the WISER database as reported by *Bhavnani et al.* (2007).

3.7.2 Object Identification under Group Queries

In this section, we consider an object identification problem under group queries $(\mathbf{B}, \Pi)$ where the queries are *a priori* grouped into n groups given by $\mathbf{z} = (z_1, \cdots, z_N)$, $z_i \in \{1, \cdots, n\}$, with the task of identifying an unknown object from the set Θ through as few queries from Q as possible, where the user is presented with a query group at each stage to choose from. Note that this approach is midway between a complete active search strategy and a complete passive search strategy. Hence, we primarily compare the performance of GQSA to a completely active search strategy such as GBS and a completely passive search strategy like random search where the user randomly chooses the queries from the set Q to answer. In addition, we also compare GQSA to other possible heuristics where we choose a query group i that minimizes $\min_{q \in Q^i} p_i(q)\rho_a(q)$ or $\max_{q \in Q^i} p_i(q)\rho_a(q)$ at each internal node 'a'.

First, we compare the performances of these algorithms on random datasets generated using a random data model. Then, we compare them in the WISER database. In both these experiments, we assume uniform *a priori* probability distribution on the objects as well as on queries within a group. The latter probability distribution corresponds to the probability of a user selecting a particular query q from a query group, $p_i(q), \forall i = 1, \cdots, n$.

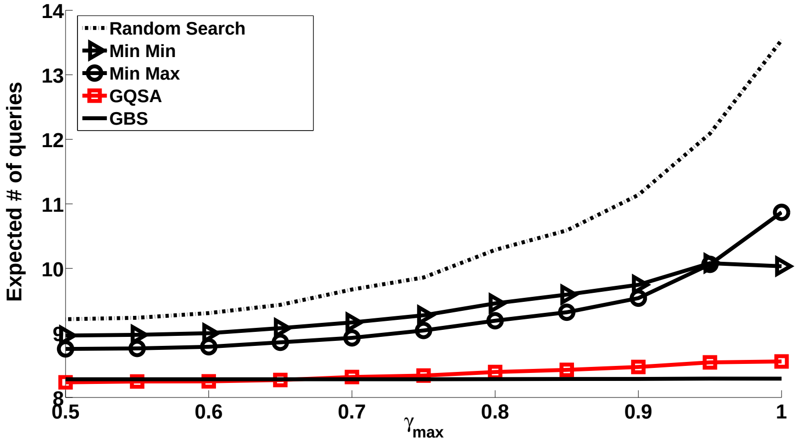


Figure 3.9: Compares the average query complexity of different algorithms for object identification under group queries in synthetic datasets

3.7.2.1 Random Datasets

Here, we consider random datasets of the same size as the WISER database, with 298 objects and 79 queries where the queries are grouped into 10 groups with the same group sizes as that in the WISER database. We associate a random dataset with a parameter $\gamma_{max} \in [0.5, 1]$, where γ_{max} corresponds to the maximum permissible value of γ_b for a query in the random dataset. Given a γ_{max}, a random dataset is created as follows

1. For each query group, generate a $\gamma_b \in [0.5, \gamma_{max}]$

2. For each query in the query group, generate a Bernoulli random variable x and give each object the same query label as x with probability γ_b

Figure 3.9 compares the mean $\mathbb{E}[K(T)]$ for the respective algorithms in 100 randomly generated datasets, for each value of γ_{max}. The min min corresponds to the heuristic where we minimize $\min_{q \in Q^i} p_i(q)\rho_a(q)$ at each internal node and the min max corresponds to the heuristic where we minimize $\max_{q \in Q^i} p_i(q)\rho_a(q)$. Note from the figure that in spite of not being a completely active search strategy, the performance of GQSA is comparable to that of GBS and better than the other algorithms.

Algorithm	$\mathbb{E}[K(T)]$
GBS	8.283 ± 0.000
GQSA	11.360 ± 0.096
$\min_i \min_{q \in Q^i} p_i(q)\rho_a(q)$	13.401 ± 0.116
$\min_i \max_{q \in Q^i} p_i(q)\rho_a(q)$	18.697 ± 0.357
Random Search	20.251 ± 0.318

Table 3.2: Expected number of queries required to identify an object under group queries in the WISER database

3.7.2.2 WISER Database

Table 3.2 compares the expected number of queries required to identify an unknown object under group queries in the WISER database using the respective algorithms, where the entropy of the objects in the WISER database is given by $H(\Pi) = 8.219$. The table reports the 95% symmetric confidence intervals based on random trials, where the randomness in GBS is due to the presence of multiple best splits at each internal node.

Once again, it is not surprising that GBS outperforms GQSA as GBS is fully active, i.e, it always chooses the best split, whereas GQSA does not always pick the best split, since a human is involved. Yet, the performance of GQSA is not much worse than that of GBS. In fact, if we were to fully model the time-delay associated with answering a query, then GQSA might have a smaller "time to identification," because presumably it would take less time to answer the queries on average.

3.7.3 Object Identification under Persistent Noise

In Section 3.6, we showed that identifying an unknown object in the presence of persistent query noise can be reduced to a group identification problem. Hence, any group identification algorithm can be adopted to solve this problem. Here, we compare the performance of GBS, GISA and modified GISA under the noise model described in Section 3.6.1.

Note that this noise model requires the knowledge of the $N\nu$ queries from the set

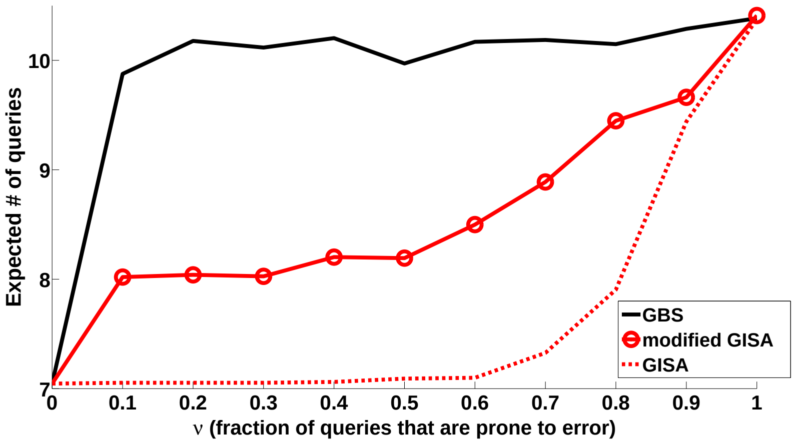
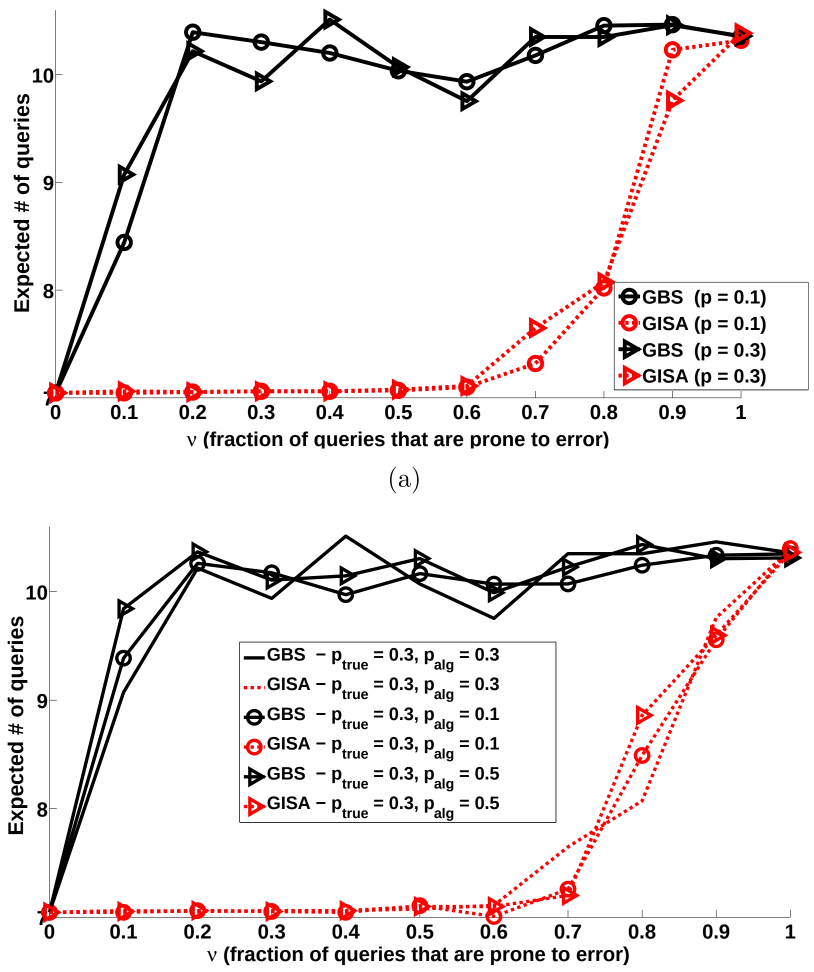


Figure 3.10: Compares the performance of GBS, modified GISA and GISA in identifying the true object in the presence of persistent query noise described in Section 3.6.1 for $p = 0.5$.

Q that are prone to error. We assume this knowledge in all our experiments in this section. Below, we show the procedure adopted to simulate the error model,

1. Select the fraction ν of the N queries that are prone to error
2. Generate $e \in \{0, \cdots, \epsilon'\}$ according to the selected probability model (p value)
3. Choose e queries from the above $N\nu$ set of queries
4. Flip the object responses of these e queries in the true object

We compare the performance of GBS, GISA and modified GISA on a subset of the WISER database consisting of 131 toxic chemicals and 79 symptom queries with $\epsilon = 2$. Figure 3.7.3 shows the expected number of queries required by GBS, GISA and modified GISA to identify the true object in the presence of a maximum of ϵ persistent errors for different values of ν, when the probability of query error p is 0.5. Note that except for the extreme cases where $\nu = 0$ and $\nu = 1$, GISA and modified GISA have great improvement over GBS. When $\nu = 0, 1$, GBS, GISA and modified GISA reduce to the same algorithm. Similar performance has been observed for different values of p as shown in Figure 3.11(a). However, we do not show modified GISA in this figure to avoid cramping.



(a)

(b)

Figure 3.11: (a) Compares the performance of GBS, modified GISA and GISA in identifying the true object in the presence of persistent query noise described in Section 3.6.1 for different values of p (b) Compares the performance of GBS and GISA under persistent noise in the presence of discrepancies between the true value of p, p_{true} and the value used in the algorithm p_{alg}

Also, note that to compute the probability distribution $\widetilde{\Pi}$ of the objects in the extended matrix $\widetilde{\mathbf{B}}$, we require the knowledge of p. Though this probability can be estimated with the help of external knowledge sources beyond the database such as domain experts, user surveys or by analyzing past query logs, the estimated value of p can vary slightly from its true value. Hence, we tested the sensitivity of the three algorithms to error in the value of p and noted that there is not much change in their performance to discrepancies in the value of p as shown in Figure 3.11(b). Once again, we do not show the results of modified GISA to avoid cramping.

CHAPTER IV

Diagnosis under Exponential Query costs

4.1 Introduction

As noted in Chapter III, the splitting algorithm or generalized binary search (GBS) is tailored to minimize the expected number of queries required to identify an unknown object θ, thereby implicitly assuming that the incremental cost for each additional query is a constant. However, in time-critical applications such as the emergency response problem of toxic chemical identification, the cost of additional queries may grow significantly. Moreover, if some chemicals are less prevalent (i.e., have a small prior), GBS may require an unacceptably large number of queries to identify them. This problem is further compounded when the prior probabilities π_i are inaccurately specified.

To address these issues, we consider an objective function where the cost of identifying an object grows exponentially in the number of queries, i.e., the cost of identifying an object using d queries is λ^d for some fixed $\lambda > 1$. Specifically, the expected cost of identifying an unknown object θ using a given tree T is defined to be

$$L_\lambda(\Pi, \mathbf{d}) := \log_\lambda \left(\sum_{i=1}^{M} \pi_i \lambda^{d_i} \right), \tag{4.1}$$

where $\lambda > 1$ and $\mathbf{d} = (d_1, \cdots, d_M)$, d_i denoting the depth of object θ_i in the given tree. In the limiting case where λ tends to 1 and ∞, this cost function reduces to the average depth and the worst case depth of a tree, respectively. That is,

$$L_1(\Pi, \mathbf{d}) = \lim_{\lambda \to 1} L_\lambda(\Pi, \mathbf{d}) = \sum_{i=1}^{M} \pi_i d_i, \text{ and}$$
$$L_\infty(\Pi, \mathbf{d}) = \lim_{\lambda \to \infty} L_\lambda(\Pi, \mathbf{d}) = \max_{i \in \{1, \cdots, M\}} d_i.$$

The above cost function was first proposed by *Campbell* (1965) in the context of source coding for the design of prefix-free codes, where an optimal binary decision tree (i.e., optimal binary prefix-free codes) that minimizes $L_\lambda(\Pi, \mathbf{d})$ can be obtained by a modified version of the Huffman algorithm (*Hu et al.*, 1979; *Parker*, 1980; *Humblet*, 1981; *Schulz*, 2008). This cost function has also been used recently in the design of alphabetic codes (*Baer*, 2006) and random search trees (*Schulz*, 2008), where efficient optimal or greedy algorithms have been presented. However, there does not exist an algorithm to the best of our knowledge that constructs a good suboptimal decision tree for the problem of object/group identification under exponential costs. Moreover, note that as GBS is tailored to minimize $L_1(\Pi, \mathbf{d})$, it may not produce a good suboptimal solution for the exponential cost function with $\lambda > 1$. Hence, we derive extensions of GBS and GGBS specifically customized to minimize $L_\lambda(\Pi, \mathbf{d})$.

Once again, we take a coding-theoretic approach to arrive at these new, greedy algorithms. In particular, we use a result by *Campbell* (1966) which states that the exponential cost of any tree T is bounded below by the α-Rényi entropy, i.e.,

$$L_\lambda(\Pi, \mathbf{d}) \geq H_\alpha(\Pi) := \frac{1}{1-\alpha} \log_2 \left(\sum_{i=1}^{M} \pi_i^\alpha \right), \tag{4.2}$$

where $\alpha = \frac{1}{1+\log_2 \lambda}$. For brevity, we will drop the dependence of the cost function on $\mathbf{d}$ and denote it as $L_\lambda(\Pi)$ in the rest of this chapter. The work in this chapter is based on *Bellala, Bhavnani and Scott* (2010).

4.2 Object Identification under Exponential Costs

We begin with the problem of object identification where the goal is to identify an unknown object $\theta \in \Theta$ in as few queries from Q as possible. We derive an explicit formula for the gap in Campbell's lower bound, and then use this formula to derive a family of greedy algorithms that minimize the exponential cost function $L_\lambda(\Pi)$ for $\lambda > 1$.

As noted earlier, the exponential cost function $L_\lambda(\Pi)$ reduces to the average depth and the worst case depth in the limiting cases where λ tends to one and infinity, respectively. In these limiting cases, the entropy lower bound on the cost function reduces to the Shannon entropy $H(\Pi)$ and $\log_2 M$, respectively.

Given an object identification problem $(\mathbf{B}, \Pi)$, let $\mathcal{T}(\mathbf{B}, \Pi)$ denote the set of decision trees that can uniquely identify all the objects in the set Θ.

Theorem IV.1. *For any $\lambda > 1$ and any $T \in \mathcal{T}(\mathbf{B}, \Pi)$, the exponential cost $L_\lambda(\Pi)$ is given by*

$$\lambda^{L_\lambda(\Pi)} = \lambda^{H_\alpha(\Pi)} + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \Big[(\lambda - 1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \Big] \tag{4.3}$$

where d_a denotes the depth of any internal node 'a' in the tree, Θ_a denotes the set of objects that reach node 'a', $\pi_{\Theta_a} = \sum_{\{i:\theta_i \in \Theta_a\}} \pi_i$, $\alpha = \frac{1}{1+\log_2 \lambda}$ and $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{\{i:\theta_i \in \Theta_a\}} \left(\frac{\pi_i}{\pi_{\Theta_a}} \right)^\alpha \right]^{1/\alpha}$.

Proof. Special case of Theorem IV.4. □

Theorem IV.1 provides an explicit formula for the gap in the Campbell's lower bound, namely, the term in summation over internal nodes $\mathcal{I}$ in (4.3). Using this result, the problem of finding a decision tree with minimum $L_\lambda(\Pi)$ can be formulated as the following optimization problem:

$$\min_{T \in \mathcal{T}(\mathbf{B},\Pi)} \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[(\lambda - 1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right] \tag{4.4}$$

As we show in Section 4.2.1, this optimization problem is a generalized version of an optimization problem that is NP-complete. Hence, we propose a suboptimal approach to solve this optimization problem where we minimize the objective function locally rather than globally. As before, we take a top-down approach and minimize the objective function by minimizing the term $\pi_{\Theta_a}\Big[(\lambda-1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{l(a)})$ $+\frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{r(a)})\Big]$ at each internal node, starting from the root node. Note that the terms that depend on the query chosen at node 'a' are $\pi_{\Theta_{l(a)}}, \pi_{\Theta_{r(a)}}, \mathcal{D}_\alpha(\Theta_{l(a)})$ and $\mathcal{D}_\alpha(\Theta_{r(a)})$. Hence, the objective function to be minimized at each internal node reduces to $C_a := \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{r(a)})$. This algorithm, which we refer to as λ-GBS, can be summarized as shown below.

```
λ-GBS
Initialization : Let the leaf set consist of the root node, $Q_{root} = \emptyset$
while some leaf node 'a' has $|\Theta_a| > 1$ do
    for each query $q \in Q \setminus Q_a$ do
        Find $\Theta_{l(a)}$ and $\Theta_{r(a)}$ produced by making a split with query $q$
        Compute the cost $C_a(q)$ of making a split with query $q$
    end
    Choose a query with the least cost $C_a$ at node 'a'
    Form child nodes $l(a), r(a)$
end
```

In the following two sections, we will show that in the limiting case when λ tends to one, where the average exponential depth reduces to the average linear depth, λ-GBS reduces to GBS, and in the case when λ tends to infinity, λ-GBS reduces to GBS with uniform prior, i.e., $\pi_i = 1/M$, $\forall i$. The latter algorithm is GBS with the true prior distribution Π replaced by a uniform distribution.

4.2.1 Average case scenario

We will use the result in the following corollary to show that in the limiting case where λ tends to 1, λ-GBS reduces to GBS.

Corollary IV.2. *In the limiting case where λ tends to* 1, (4.3) *reduces to*

$$L_1(\Pi) = H(\Pi) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a}[1 - H(\rho_a)] \tag{4.5}$$

where $H(\cdot)$ denotes the Shannon entropy and ρ_a denotes the reduction factor defined in § 3.2.

Proof. The result follows from Theorem IV.1 by taking the limit as λ tends to 1 and applying L'Hôpital's rule on both sides of (4.3). □

Note from the above corollary that in the limiting case where λ tends to 1, the optimization problem in (4.4) reduces to

$$\min_{T \in \mathcal{T}(\mathbf{B},\Pi)} \sum_{a \in \mathcal{I}} \pi_{\Theta_a}[1 - H(\rho_a)],$$

thereby reducing λ-GBS to GBS.

4.2.2 Worst case scenario

We now present the other limiting case of the family of greedy algorithms λ-GBS where $\lambda \to \infty$. As noted in Section 4.2, the exponential cost function $L_\lambda(\Pi)$ reduces to the worst case depth of any leaf node in this case. Note that GBS under a uniform prior (i.e., to choose a query that evenly splits the remaining objects at each internal node) is an intuitive algorithm for minimizing the worst case depth. As we show below, λ-GBS reduces to this algorithm as $\lambda \to \infty$.

We begin by noting that the cost function minimized at each internal node of a tree in λ-GBS is $C_a := \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)})$. Since $\log_\lambda$ is a monotonic function, this is equivalent to minimizing the function $\log_\lambda(C_a)$. It then follows from Corollary IV.3 that in the limiting case where λ tends to infinity, this criterion reduces to minimizing $\max\{|\Theta_{l(a)}|, |\Theta_{r(a)}|\}$. Hence, in this limiting case, λ-GBS reduces to GBS with uniform prior, thereby completely eliminating the dependence of the algorithm on the prior distribution Π. More generally, as λ increases, λ-GBS becomes less sensitive to the prior distribution, and therefore more robust to any misspecification of the prior.

Corollary IV.3. *In the limiting case where $\lambda \to \infty$, the optimization problem*

$$\min \log_\lambda \left(\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right) \to \min\max\{|\Theta_{l(a)}|, |\Theta_{r(a)}|\}$$

Proof. Applying L'Hôpital's rule, we get

$$\lim_{\lambda\to\infty} \log_\lambda \left(\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right) = \max\{\log_2 |\Theta_{l(a)}|, \log_2 |\Theta_{r(a)}|\}$$

Since $\log_2$ is a monotonic increasing function, the optimization problem, $\min\max \{\log_2 |\Theta_{l(a)}|, \log_2 |\Theta_{r(a)}|\}$ is equivalent to the optimization problem, $\min\max\{|\Theta_{l(a)}|, |\Theta_{r(a)}|\}$. □

4.3 Group Identification under Exponential Costs

For the sake of completeness, we will now consider the problem of group identification where the cost of identifying the group of an object grows exponentially in the number of queries. In Section 3.3, we considered a special case of this problem where the cost grows linearly in the number of queries. In this context, we also noted that a greedy decision tree constructed for group identification can have multiple objects ending in the same leaf node and multiple leaves ending in the same group. For a tree with L leaves, we let $\mathcal{L}^i \subset \mathcal{L} = \{1, \cdots, L\}$ denote the set of leaves that terminate in group i. Also, we let $\Theta_a^i \subseteq \Theta_a$ denote the set of objects belonging to group i that reach internal node $a \in \mathcal{I}$ in a tree, where $\Theta^i \subseteq \Theta$ denotes the set of objects belonging to group i at the root node of any tree.

Given a group identification problem $(\mathbf{B}, \Pi, \mathbf{y})$ where $\mathbf{y}$ denotes the group labels, let $\mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$ denote the set of decision trees that can uniquely identify the groups of all objects in the set Θ. For any decision tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, let d_j denote the depth of leaf node $j \in \mathcal{L}$. Let random variable K denote the exponential cost incurred in identifying the group of an unknown object $\theta \in \Theta$. Then, the average exponential cost $L_\lambda(\Pi)$ of identifying the group of the unknown object θ using a given tree is defined as

$$\begin{aligned}
\lambda^{L_\lambda(\Pi)} &= \sum_{i=1}^{m} \Pr(\theta \in \Theta^i)\mathbb{E}[K|\theta \in \Theta^i] \\
&= \sum_{i=1}^{m} \pi_{\Theta^i} \left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} \lambda^{d_j} \right] \\
\Longrightarrow L_\lambda(\Pi) &= \log_\lambda \left(\sum_{i=1}^{m} \pi_{\Theta^i} \left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} \lambda^{d_j} \right] \right)
\end{aligned}$$

In the limiting case where λ tends to one and infinity, the cost function $L_\lambda(\Pi)$ reduces to

$$\begin{aligned}
L_1(\Pi) &:= \lim_{\lambda \to 1} L_\lambda(\Pi) = \sum_{i=1}^{m} \pi_{\Theta^i} \left[\sum_{j \in \mathcal{L}^i} \frac{\pi_{\Theta_j}}{\pi_{\Theta^i}} d_j \right], \\
L_\infty(\Pi) &:= \lim_{\lambda \to \infty} L_\lambda(\Pi) = \max_{j \in \mathcal{L}} \ d_j.
\end{aligned}$$

Theorem IV.4. *For any $\lambda > 1$ and any tree $T \in \mathcal{T}(\mathbf{B}, \Pi, \mathbf{y})$, the exponential cost*

$L_\lambda(\Pi)$ *of identifying the group of an object is given by*

$$\lambda^{L_\lambda(\Pi)} = \lambda^{H_\alpha(\Pi_{\mathbf{y}})} + \sum_{a\in\mathcal{I}} \pi_{\Theta_a}\Big[(\lambda-1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{r(a)})\Big] \tag{4.6}$$

where $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^m})$ *denotes the probability distribution of the object groups induced by the labels* $\mathbf{y}$*, and* $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{i=1}^m \left(\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}\right)^\alpha\right]^{1/\alpha}$ *with* $\alpha = \frac{1}{1+\log_2\lambda}$, $\pi_{\Theta^i} = \sum_{\{k:y_k=i\}} \pi_k$ *and* $\pi_{\Theta_a^i} = \sum_{\{k:\theta_k\in\Theta_a, y_k=i\}} \pi_k$.

Proof. The proof is given in Appendix B. □

Note that the definition of $\mathcal{D}_\alpha(\Theta_a)$ in this theorem is a generalization of that in Theorem IV.1. Also note that Theorem IV.1 is a special case of this theorem where each group is of size 1.

Using the result in the above theorem, the problem of finding a decision tree with minimum cost function $L_\lambda(\Pi)$ can be formulated as the following optimization problem:

$$\min_{T\in\mathcal{T}(\mathbf{B},\Pi,\mathbf{y})} \sum_{a\in\mathcal{I}} \pi_{\Theta_a}\left[(\lambda-1)\lambda^{d_a} - \mathcal{D}_\alpha(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]. \tag{4.7}$$

This optimization problem being the generalized version of the optimization problem in (4.4) is NP-complete. Hence, we propose a suboptimal algorithm to solve this optimization problem where we take a top-down approach and minimize the objective function by minimizing the term $C_a := \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}}\mathcal{D}_\alpha(\Theta_{r(a)})$ at each internal node, starting from the root node. This algorithm, which we refer to as λ-GGBS, is summarized below.

```
$\lambda$-Group Generalized Binary Search ($\lambda$-GGBS)
Initialization : Let the leaf set consist of the root node, $Q_{root} = \emptyset$
while some leaf node 'a' has more than one group of objects do
    for each query $q \in Q \setminus Q_a$ do
        Compute $\{\Theta^i_{l(a)}\}_{i=1}^m$ and $\{\Theta^i_{r(a)}\}_{i=1}^m$ produced by making a split with query $q$
        Compute the cost $C_a(q)$ of making a split with query $q$
    end
    Choose a query with the least cost $C_a$ at node 'a'
    Form child nodes $l(a), r(a)$
end
```

4.3.1 Average case scenario

We now consider the limiting case where λ tends to 1, and show that λ-GGBS reduces to GGBS in this case.

Corollary IV.5. *In the limiting case where λ tends to* 1, (4.6) *reduces to*

$$L_1(\Pi) = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i) \right] \tag{4.8}$$

where $\Pi_{\mathbf{y}} = (\pi_{\Theta^1}, \cdots, \pi_{\Theta^m})$ denotes the probability distribution of the object groups induced by the labels $\mathbf{y}$, ρ_a denotes the reduction factor defined in § 3.2, ρ_a^i denotes the group reduction factor defined in § 3.3 and $H(\cdot)$ denotes the Shannon entropy.

Proof. The result follows by taking the limit as λ tends to 1 and applying L'Hôpital's rule on both sides of (4.6). For more details, refer to Appendix B. □

It follows from the above corollary that in the limiting case where λ tends to 1, the optimization problem in (4.7) reduces to

$$\min_{T \in \mathcal{T}(\mathbf{B},\Pi,\mathbf{y})} \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i) \right],$$

thereby reducing λ-GGBS to GGBS.

4.3.2 Worst case scenario

We now present λ-GGBS in the limiting case where λ tends to infinity. As noted above, the exponential cost function $L_\lambda(\Pi)$ reduces to the worst case depth of any leaf node in this limiting case.

We begin by noting that the cost function minimized at each internal node of a tree in λ-GGBS is $C_a := \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)})$. Since $\log_\lambda$ is a monotonic function, this is equivalent to minimizing the function $\log_\lambda(C_a)$. Then, defining N_a to be the number of groups at any node 'a' in a tree, i.e., $N_a = |\{i \in \{1, \cdots, m\} : \Theta_a^i \neq \emptyset\}|$, it follows from Corollary IV.6 that in the limiting case where $\lambda \to \infty$, the criterion in λ-GGBS reduces to minimizing $\max\{N_{l(a)}, N_{r(a)}\}$ at each internal node in the tree.

Corollary IV.6. *In the limiting case where $\lambda \to \infty$, the optimization problem*

$$\min \log_\lambda \left(\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right) \longrightarrow \min \max\{N_{l(a)}, N_{r(a)}\}$$

where $\mathcal{D}_\alpha(\Theta_a) = \left[\sum_{i=1}^m \left(\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \right)^\alpha \right]^{\frac{1}{\alpha}}$

Proof. Applying L'Hôpital's rule, we get

$$\lim_{\lambda \to \infty} \log_\lambda \left(\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} \mathcal{D}_\alpha(\Theta_{r(a)}) \right) = \max\{\log_2 N_{l(a)}, \log_2 N_{r(a)}\}$$

Since $\log_2$ is a monotonic increasing function, the optimization problem, $\min \max \{\log_2 N_{l(a)}, \log_2 N_{r(a)}\}$ is equivalent to the optimization problem, $\min \max\{N_{l(a)}, N_{r(a)}\}$. □

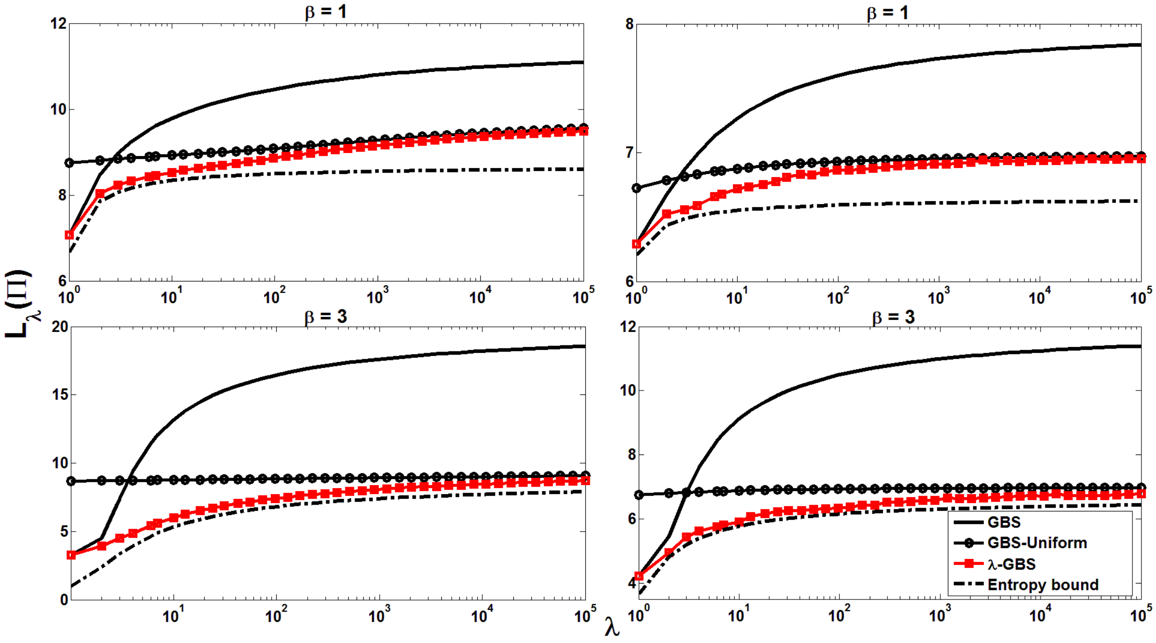


Figure 4.1: Experiments to demonstrate the improved performance of λ-GBS over GBS and GBS with uniform prior. The plots in the first column correspond to the WISER database and those in the second column correspond to synthetic data.

4.4 Experimental Evaluation

We demonstrate the performance of the proposed algorithms through experiments on both synthetic data and the WISER database. In particular, we compare the performance of λ-GBS with that of GBS and GBS under a uniform prior for different values of λ. Figure 4.1 demonstrates the improved performance of λ-GBS over a range of λ values. Each curve in this figure corresponds to the average value (averaged over 100 repetitions) of the cost function $L_\lambda(\Pi)$ as a function of λ.

The plots in the first column correspond to the WISER database. Here, in each repetition, the prior is generated according to Zipf's law, i.e., $(k^{-\beta}/\sum_{i=1}^{M} i^{-\beta})_{k=1}^{M}$, $\beta \geq 0$, after randomly permuting the objects. Note that in the special case, when $\beta = 0$, this reduces to the uniform distribution and as β increases, it tends to a skewed distribution with most of the probability mass concentrated on a single object.

The plots in the second column correspond to synthetic data based on an active learning application. We consider a two-dimensional setting where the classifiers are restricted to be linear classifiers of the form $sign(x_i - c)$, $sign(c - x_i)$, where $i = 1, 2$ and c takes on 25 distinct values. The number of distinct classifiers is therefore 100, and the number of queries is $26^2 = 676$. The goal is to identify the classifier by selecting queries judiciously. Here, the prior is generated such that the classifiers that are close to $x_i = 0$ are more likely than the ones away from the axes, with their relative probability decreasing according to Zipf's law $k^{-\beta}$, $\beta \geq 0$. Hence, the prior is the same in each repetition. However, the randomness in each repetition comes from the greedy algorithms due to the presence of multiple best splits at each internal node. Note that in all the experiments, λ-GBS performs better than GBS and GBS with uniform prior. We also see that λ-GBS converges to GBS as $\lambda \to 1$ and to GBS with uniform prior as $\lambda \to \infty$.

CHAPTER V

Diagnosis under Persistent Query Noise

5.1 Introduction

In this chapter, we re-visit the problem of active diagnosis under persistent query noise. As mentioned in Chapter III, the problem of active diagnosis/active learning in the presence of query noise has been studied in the literature (*Kääriäinen*, 2006; *Nowak*, 2008, 2009), where the noise is assumed to be independent, in that posing the same query twice may yield different responses. This assumption suggests repeated selection of a query as a possible strategy to overcome query noise. The algorithms presented in (*Kääriäinen*, 2006; *Nowak*, 2008, 2009) are based on this principle. However, in certain applications, resampling or repeating a query does not change the query response, thereby confining an active diagnosis algorithm to non-repeatable queries.

For example, in the emergency response problem of toxic chemical identification (*Bhavnani et al.*, 2007), a first responder is faced with the task of rapidly identifying the toxic chemical by posing symptom-based queries to a victim. The responses to these symptom queries are often in error due to reasons such as mis-identification of a symptom by a victim or a delayed onset of a symptom, in which case the victim's response is unlikely to change upon repeated queries. Similarly, in a fault diagnosis problem, the response to alarms/probes could be in error due to faulty alarms, in which case these responses would not change on repeated interrogations.

This more stringent noise model where queries cannot be resampled is referred to as persistent noise (*Rényi*, 1961; *Hanneke*, 2007). It has been studied earlier in the situation where the number of persistent errors is restricted such that unique identification of the unknown object θ is guaranteed (*Bellala et al.*, 2011b; *Golovin et al.*, 2010), as described in more detail in Section 3.6. In particular, the number of query errors is restricted to be less than half of the minimum Hamming distance between any two object bit strings (equivalently, any two row vectors in $\mathbf{B}$). This is often not reasonable as the minimum Hamming distance could be very small, such as in WISER where it is equal to 1.

In this chapter, we consider the problem of active diagnosis under persistent noise with no restriction on the number of persistent errors. We assume the object set Θ and the query set Q are finite, and that only one object from Θ is "present". Unlike the previous two noise models where the unknown object θ can be identified with certainty after sufficiently many queries, in this model it may not be possible to identify θ even after all queries are made.

In this setting, *Rish et al.* (2005) proposed the use of mutual information or the conditional entropy as a criterion for selecting queries, where queries are chosen sequentially to minimize the uncertainty in θ (or maximize information gain) given the observed responses to the past queries. After observing responses to a set of queries, the unknown object is then estimated to be the object with the maximum *a posteriori* probability, θ_{MAP}.

However, there are two limitations with this approach. First, in situations with moderate to high noise, or where the Hamming distance between object bit strings is low, the object with the maximum *a posteriori* probability will be equal to the true object θ with low probability. Even in the case where θ_{MAP} does converge to the true object θ, it may require a large number of queries to be inputted. Second, this algorithm assumes knowledge of the underlying query noise model; in particular, it assumes knowledge of the probability of query errors, which is required to compute the information gain in the query selection stage. However, this information is often not known.

To address these issues, we propose a novel rank-based approach where we output a ranked list of objects rather than θ_{MAP}, where the ranking is based on the posterior probabilities. The rank-based approach is motivated by the fact that in many applications there is a domain expert who makes the final decision on the possible identity of the unknown object θ. Such a ranking can be useful to a domain expert who will use domain expertise and other sources of information to determine θ from the ranked list. Thus, we propose a greedy algorithm to minimize the expected rank of the unknown object θ. Moreover, the proposed greedy algorithm exhibits an interesting property in that it does not require knowledge of the underlying noise distribution, unlike the entropy-based algorithm. Finally, the work in this chapter is based on *Bellala, Bhavnani and Scott* (2011a).

5.2 Data Model

We consider the input to an active diagnosis problem as a bipartite diagnosis graph (BDG) or a binary matrix $\mathbf{B}$ denoting the relation between a set of M different objects Θ and N distinct queries Q.

We associate each object $\theta_i \in \Theta$ with a binary random variable X_i, where $X_i = 1$ when the unknown object $\theta = \theta_i$, and 0 otherwise. Then, $\mathbf{X} = (X_1, \cdots, X_M)$ is a binary random vector denoting the states of all the objects in Θ, where $\mathbf{X} \in \{\mathbb{I}_1, \cdots, \mathbb{I}_M\}$, $\mathbb{I}_i$ being a binary vector whose ith element is 1 and remaining elements are 0.

Similarly, let Z_j be a binary random variable denoting the observed response to query q_j. Then, $\mathbf{Z} = (Z_1, \cdots, Z_N)$ is a binary random vector denoting the observed

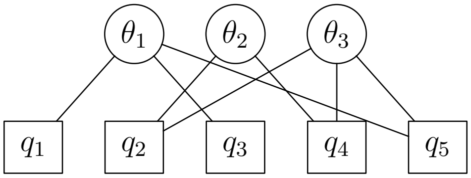


Figure 5.1: A toy bipartite diagnosis graph (BDG) where the circled nodes denote the objects and the square nodes denote queries.

responses to all queries in Q, where $\mathbf{Z} \in \{0,1\}^N$.

In addition, we let $Q_{\mathcal{A}}$ denote the subset of queries indexed by $\mathcal{A} \subseteq \{1, \cdots, N\}$, and $\mathbf{Z}_{\mathcal{A}}$ the random variables associated with those queries, e.g, if $\mathcal{A} = \{1, 4, 7\}$, then $Q_{\mathcal{A}} = \{q_1, q_4, q_7\}$ and $\mathbf{Z}_{\mathcal{A}} = (Z_1, Z_4, Z_7)$. Also, for any query j, let $\mathbf{pa}_j$ denote the objects that are connected to it in the BDG. Then, $\mathbf{X}_{\mathbf{pa}_j}$ denotes the states of all the objects connected to query j, e.g., for query 2 in Figure 5.1, $\mathbf{X}_{\mathbf{pa}_2} = (X_2, X_3)$.

We need to specify the joint distribution of $(\mathbf{X}, \mathbf{Z})$, and more generally $(\mathbf{X}, \mathbf{Z}_{\mathcal{A}})$ for any $\mathcal{A}$, which can be defined in terms of a prior probability distribution on $\mathbf{X}$ and a conditional distribution on $\mathbf{Z}_{\mathcal{A}}$ given $\mathbf{X}$. The prior probability distribution on $\mathbf{X}$ is given by $\Pi = (\pi_1, \cdots, \pi_M)$, where $\pi_i = \Pr(\mathbf{X} = \mathbb{I}_i) = \Pr(X_i = 1) = \Pr(\theta = \theta_i)$, and $X_i = 1 \iff \mathbf{X} = \mathbb{I}_i$. To define the conditional distribution on $\mathbf{Z}_{\mathcal{A}}$ given $\mathbf{X}$, we make the standard assumption that the observed responses to queries are conditionally independent given the states of the objects connected to them, i.e.,

$$\Pr(\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}} | X_i = 1) = \prod_{j \in \mathcal{A}} \Pr(Z_j = z_j | X_i = 1),$$

where once again by $X_i = 1$ we implicitly mean $\mathbf{X} = \mathbb{I}_i$. This assumption holds reasonably well in many practical applications as noise is usually generated independently. For example, in the problem of fault diagnosis, it can be reasonable to assume that all connections and alarms fail independently.

Note that in the ideal case when there is no noise, the observed response Z_j to query j is deterministic given the binary states of the objects in $\mathbf{pa}_j$. Specifically, it is given by the OR operation of the binary variables in $\mathbf{X}_{\mathbf{pa}_j}$, i.e., $Z_j = 1 \iff \exists\ i \in \mathbf{pa}_j$ s.t. $X_i = 1$. More generally, it is a noisy OR operation (*Pearl*, 1988), where the conditional distribution of Z_j given $\mathbf{x}_{\mathbf{pa}_j}$ can be defined using standard noise models such as the Y-model (*Le and Hadjicostis*, 2007) or the QMR-DT model (*Jaakkola and Jordan*, 1999).

We derive our rank-based active diagnosis algorithm under this general probability model, and in Section 5.3.3, we consider special cases of the QMR-DT noise model that arise in several applications. In these special cases, we derive a noise adaptive active diagnosis algorithm that does not depend on the underlying noise parameters. The QMR-DT noise model can be described using two sets of parameters as shown below,

$$\begin{aligned}\Pr(Z_j = 0|X_i = 1) &:= 1 - \rho_{0j}, \text{ if } b_{ij} = 0, \text{and}\\ \Pr(Z_j = 0|X_i = 1) &:= (1 - \rho_{0j})\rho_{ij}, \text{ if } b_{ij} = 1,\end{aligned}$$

where $0 \leq \rho_{ij} \leq 1, \forall i \in \{1, \cdots, M\}, \forall j \in \{1, \cdots, N\}$ and $0 \leq \rho_{0j} \leq 1, \forall j\{1, \cdots, N\}$, are the so-called inhibition and leak probabilities, respectively.

5.3 Active Diagnosis under Persistent Noise

We will now formally state the problem of active diagnosis under persistent noise. As mentioned earlier, unique identification of the unknown object θ (equivalently, the binary vector $\mathbf{X}$) is no longer guaranteed. Hence, the goal of active diagnosis under persistent noise is to maximize some function $f(\mathbf{X}; \mathbf{Z}_{\mathcal{A}})$ which captures the quality of the estimate of $\mathbf{X}$ based on the responses to queries in $\mathcal{A}$, subject to a constraint on the number of queries made, i.e.,

$$\begin{aligned} &\max\nolimits_{\mathcal{A}\subseteq\{1,\cdots,N\}} f(\mathbf{X};\mathbf{Z}_{\mathcal{A}}) \\ &\text{s.t.} \quad |\mathcal{A}| \leq k. \end{aligned} \tag{5.1}$$

Finding an optimal solution to this problem is typically not computationally feasible (*Rish et al.*, 2005). Instead, the queries can be chosen sequentially by greedily maximizing the quality function, i.e., given the observed responses to the past queries, the next best query is chosen to be

$$j^* := \arg\max_{j\notin\mathcal{A}} \ \mathbb{E}_{Z_j}[f(\mathbf{X};\mathbf{Z}_{\mathcal{A}}\cup Z_j) - f(\mathbf{X};\mathbf{Z}_{\mathcal{A}})|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}], \tag{5.2}$$

where $\mathbf{Z}_{\mathcal{A}}\cup Z_j$ denotes the random variables associated with queries in $\mathcal{A}\cup\{j\}$.

5.3.1 Entropy-based Active Query Selection

Mutual information has been traditionally chosen as a function to measure the quality of the estimate of the object states $\mathbf{X}$ based on the responses to queries in $\mathcal{A}$. The expression for the quality function $f(\mathbf{X};\mathbf{Z}_{\mathcal{A}})$ is then given by $f(\mathbf{X};\mathbf{Z}_{\mathcal{A}}) = I(\mathbf{X};\mathbf{Z}_{\mathcal{A}}) := H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Z}_{\mathcal{A}})$. However, the optimization problem in (5.1) with mutual information as the quality function is NP-hard (*Rish et al.*, 2005). Alternatively, the greedy approach can be used to choose queries sequentially where given the observed responses $\mathbf{z}_{\mathcal{A}}$ to previously selected queries in $\mathcal{A}$, the next best query is chosen to be the one that maximizes the expected information gain as shown below,

$$\begin{aligned} j^* &= \arg\max_{j\notin\mathcal{A}} \ \mathbb{E}_{Z_j}[I(\mathbf{X};\mathbf{Z}_{\mathcal{A}}\cup Z_j) - I(\mathbf{X};\mathbf{Z}_{\mathcal{A}})|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}] \\ &= \arg\min_{j\notin\mathcal{A}} \sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}})H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}, z). \end{aligned} \tag{5.3}$$

Note that information gain based greedy query selection reduces to choosing a query that minimizes the expected conditional entropy of the object states $\mathbf{X}$. Hence, we will refer to this approach as entropy-based active query selection in the rest of this thesis.

Given the posterior probabilities, the conditional entropy term in (5.3) can be computed as follows

$$H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}, z) = -\sum_{i=1}^{M} \Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{z}_{\mathcal{A}}, z) \log_2 \Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{z}_{\mathcal{A}}, z).$$

However, the computation of these posterior probabilities requires the knowledge of the complete noise distribution or the parameters in the noise model. Moreover, as we show in Section 5.4, entropy-based active query selection can be sensitive to discrepancies in the knowledge of these parameters.

In the next section, we propose a rank-based greedy algorithm that depends instead on the likelihoods and the prior probability distribution. We then exploit this fact in Section 5.3.3 to develop algorithms that do not require knowledge of the query noise parameters.

5.3.2 Rank-based Active Query Selection

Given the observed responses $\mathbf{z}_{\mathcal{A}}$ to a set of queries $Q_{\mathcal{A}}$, we define the worst case rank of an object θ_i to be

$$\begin{aligned} r_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}}) &= \sum_{k=1}^{M} \mathbf{I}\Big\{\Pr(X_k = 1|\mathbf{z}_{\mathcal{A}}) \geq \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}})\Big\} \\ &= \sum_{k=1}^{M} \mathbf{I}\Big\{\pi_k \Pr(\mathbf{z}_{\mathcal{A}}|X_k = 1) \geq \pi_i \Pr(\mathbf{z}_{\mathcal{A}}|X_i = 1)\Big\}, \end{aligned}$$

where $\mathbf{I}\{E\}$ is an indicator function which takes the value 1 when the event E is true, and 0 otherwise. Note that $r_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}})$ takes a small value when θ_i has a high

	θ_1	θ_2	θ_3	θ_4	θ_5
$\Pr(X_i = 1 \vert \mathbf{z}_{\mathcal{A}})$	0.2	0.2	0.3	0.2	0.1
$R(\theta_i \vert \mathbf{z}_{\mathcal{A}})$	4	4	1	4	5

Figure 5.2: Demonstration of worst case ranking

posterior probability and a large value when the posterior probability is small. In addition, when multiple objects have the same posterior probabilities, each object is assigned the worst case ranking, as shown in Figure 5.2.

Given the ranks of all the objects in Θ, we measure the quality of the obtained ranking (thereby, the quality of diagnosis) as

$$f(\mathbf{X}; \mathbf{Z}_{\mathcal{A}}) = \mathbb{E}_{\theta}[r_{wc}(\theta|\mathbf{z}_{\mathcal{A}})] = \sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) r_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}}), \tag{5.4}$$

which corresponds to the expected worst case rank of the unknown object θ. The goal of active diagnosis is to choose queries such that the expected rank is minimized. Substituting this objective function in (5.2), we get the criterion for choosing the next best query to be

$$\begin{aligned} j^* &= \arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) \mathbb{E}_{\theta}[r_{wc}(\theta|\mathbf{z}_{\mathcal{A}}, z)] \\ &= \arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \sum_{i=1}^{M} \frac{\pi_i \Pr(\mathbf{z}_{\mathcal{A}}, z|X_i = 1)}{\Pr(\mathbf{z}_{\mathcal{A}})} r_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}}, z) \\ &= \arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \sum_{i=1}^{M} \pi_i \Pr(\mathbf{z}_{\mathcal{A}}, z|X_i = 1) r_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}}, z) \end{aligned} \tag{5.5}$$

where (5.5) follows as $\Pr(\mathbf{z}_{\mathcal{A}})$ does not depend on query q_j. In the noise-free case with uniform prior, this greedy strategy reduces to GBS (*Dasgupta*, 2004; *Nowak*, 2008) as shown in Appendix C.

In the noisy case, given the knowledge of the prior distribution Π and the noise parameters such as the leak and the inhibition probabilities in the QMR-DT noise model, the greedy algorithm in (5.5) can be implemented efficiently. However, these noise parameters are often not known, and hence it is desirable for a greedy query selection criterion to be robust to any discrepancies in the knowledge of these parameters. As we show in Section 5.4, entropy-based active query selection can be sensitive to discrepancies in the noise parameters.

In the next two sections, we consider two special cases of the noise model discussed in Section 5.2 that appear in many applications. In these two cases, we present a noise adaptive estimate of the query selection criterion in (5.5) that does not require knowledge of the underlying noise parameters.

5.3.3 Noise Adaptive Active Query Selection

We will now present a noise adaptive estimate of the objective in (5.5) under two special cases of the noise model discussed in Section 5.2 that appears in many applications. Specifically, we take advantage of the fact that the above query selection criterion depends on the noise parameters only through the likelihood function, and provide a good upper bound on the likelihood function that is independent of noise parameters. This enables accurate prediction of the worst case rank of the objects without requiring knowledge of the true noise parameters. Furthermore, we show that in some cases it is possible to estimate the true ranks exactly with limited knowledge on the query noise. The bound on the likelihood function is based on the following lemma.

Lemma V.1. *Let h, k be integers with $0 \leq h \leq k$ and $k \geq 1$. Then, for any $0 < p < 1$,*

$$p^h(1-p)^{k-h} \leq \varepsilon_h^h(1-\varepsilon_h)^{k-h} \tag{5.6}$$

where $\varepsilon_h = \frac{h}{k}$. If it is known that $p \leq p_2 < 1$, then (5.6) *holds with $\varepsilon_h = \min\{p_2, \frac{h}{k}\}$.*

If it is known that $p \geq p_1 > 0$*, then* (5.6) *holds with* $\varepsilon_h = \max\{p_1, \frac{h}{k}\}$*. If it is known that* $0 < p_1 \leq p \leq p_2 < 1$*, then* (5.6) *holds with* $\varepsilon_h = \min\{p_2, \max\{p_1, \frac{h}{k}\}\}$*.*

Proof. Refer to Appendix C. □

5.3.3.1 Constant Noise Level

We begin with a special case of the noise model described in Section 5.2, where the responses to some queries are assumed to be randomly flipped. This noise model has been used in the context of pool-based active learning with a faulty oracle (*Hanneke*, 2007; *Nowak*, 2009), experimental design (*Rényi*, 1961), computer vision, and image processing (*Korostelev and Kim*, 2000).

In this setting,

$$\Pr(Z_j = z_j | X_i = 1) = p^{|b_{ij} - z_j|}(1-p)^{1-|b_{ij}-z_j|}.$$

More generally, the likelihood function can be expressed as shown below,

$$\Pr(\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}} | X_i = 1) = p^{\delta_{i,\mathcal{A}}}(1-p)^{|\mathcal{A}| - \delta_{i,\mathcal{A}}},$$

where $\delta_{i,\mathcal{A}} = \sum_{j \in \mathcal{A}} |b_{ij} - z_j|$, is the local Hamming distance between the true responses of object i to queries in $\mathcal{A}$, and the observed responses $\mathbf{z}_{\mathcal{A}}$. Using the result in Lemma V.1, the above likelihood function can be upper bounded by

$$\overline{\Pr}(\mathbf{z}_{\mathcal{A}} | X_i = 1) := \left(\frac{\delta_{i,\mathcal{A}}}{|\mathcal{A}|}\right)^{\delta_{i,\mathcal{A}}} \left(1 - \frac{\delta_{i,\mathcal{A}}}{|\mathcal{A}|}\right)^{|\mathcal{A}| - \delta_{i,\mathcal{A}}}.$$

Note that the lemma also states that given an upper or lower bound on the noise parameter p, this bound can be further improved.

Finally, let $\overline{r}_{wc}(i|\mathbf{z}_{\mathcal{A}})$ denote the estimated worst case rank of object i based on the upper bound on the likelihood function:

$$\overline{r}_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}}) := \sum_{j=1}^{M} \mathbf{I}\Big\{\pi_j \overline{\Pr}(\mathbf{z}_{\mathcal{A}}|X_j = 1) \geq \pi_i \overline{\Pr}(\mathbf{z}_{\mathcal{A}}|X_i = 1)\Big\}. \tag{5.7}$$

Then, the query selection criterion in (5.5) can be replaced by the following noise-independent criterion

$$\arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \sum_{i=1}^{M} \pi_i \overline{\Pr}(\mathbf{z}_{\mathcal{A}}, z|X_i = 1) \overline{r}_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}} \cup z). \tag{5.8}$$

The result in Proposition V.2 presents conditions under which the true rank can be estimated accurately. It states that, under uniform prior on the objects, it suffices to know whether $p < 0.5$ or $p > 0.5$, for the estimated ranks to be exactly equal to the true ranks.

More generally, for any given prior Π with $\rho := \min_i \pi_i / \max_i \pi_i$, it suffices to know whether $p < \frac{\rho}{1+\rho}$ or $p > \frac{1}{1+\rho}$, for the estimated ranks to be equal to the true ranks. Even in the case where $\frac{\rho}{1+\rho} \leq p < 0.5$ or $0.5 < p \leq \frac{1}{1+\rho}$, we observe through experiments that the estimated ranks are equal to the true ranks for most objects.

Proposition V.2. *Let $\mathbf{z}_{\mathcal{A}}$ be the observed responses to a sequence of queries in $\mathcal{A}$, under some unknown noise parameter p. Let $\rho := \min_i \pi_i / \max_i \pi_i$. Given a $\overline{p} \in (0, \frac{\rho}{1+\rho})$ such that $0 < p \leq \overline{p}$, or a $\underline{p} \in (\frac{1}{1+\rho}, 1)$ such that $1 > p \geq \underline{p}$, the estimated ranks $\overline{r}_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}})$ computed only with the knowledge of $\overline{p}$ or $\underline{p}$ are equal to the true ranks $r_{wc}(\theta_i|\mathbf{z}_{\mathcal{A}})$, $\forall\ 1 \leq i \leq M$.*

Proof. Refer to Appendix C. □

5.3.3.2 Response Dependent Noise

We now consider the noise model where the probability of error depends on the true response to a query. When the true response is 0, the probability of observing a noisy response is given by ν_0, and by ν_1 when the true response is 1, i.e.,

$$\begin{aligned}\Pr(Z_j = 0|X_i = 1) = 1 - \nu_0, \text{ if } b_{ij} = 0,\\ \text{and } \Pr(Z_j = 0|X_i = 1) = \nu_1, \text{ if } b_{ij} = 1.\end{aligned}$$

For example, consider the following special case of the QMR-DT noise model described in Section 5.2 where $\rho_{0j} = \rho_0,\ \forall j$ and $\rho_{kj} = \rho,\ \forall k \neq 0, j$. This case reduces to the above setting with $\nu_0 = \rho_0$ and $\nu_1 = (1 - \rho_0)\rho$, where $0 < \rho_0, \rho < 1$ are the leak and inhibition probabilities, respectively.

For any subset of indices $\mathcal{A} \subseteq \{1, \cdots, N\}$, let $\mathcal{A}_0^i = \{j \in \mathcal{A} : b_{ij} = 0\}$ and $\mathcal{A}_1^i = \{j \in \mathcal{A} : b_{ij} = 1\}$ be partitions of $\mathcal{A}$ for each $i = 1, \cdots, M$ such that the true response b_{ij} of object i to queries in $\mathcal{A}_0^i$ is 0, and that in $\mathcal{A}_1^i$ is 1. Then, the likelihood function is given by

$$\Pr(\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}|X_i = 1) = \nu_0^{\delta_{i,\mathcal{A}_0^i}} (1 - \nu_0)^{|\mathcal{A}_0^i| - \delta_{i,\mathcal{A}_0^i}} \cdot \nu_1^{\delta_{i,\mathcal{A}_1^i}} (1 - \nu_1)^{|\mathcal{A}_1^i| - \delta_{i,\mathcal{A}_1^i}}$$

where $\delta_{i,\mathcal{A}_0^i} = \sum_{j \in \mathcal{A}_0^i} |0 - z_j|$ and $\delta_{i,\mathcal{A}_1^i} = \sum_{j \in \mathcal{A}_1^i} |1 - z_j|$, are the local Hamming distances between the true responses of object i to queries in $\mathcal{A}_0^i$ and $\mathcal{A}_1^i$, and that of their observed responses.

Once again, using Lemma V.1, this likelihood function can be upper bounded by

$$\begin{aligned}\overline{\Pr}(\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}|X_i = 1) \quad = \quad & \left(1 - \frac{\delta_{i,\mathcal{A}_0^i}}{|\mathcal{A}_0^i|}\right)^{|\mathcal{A}_0^i| - \delta_{i,\mathcal{A}_0^i}} \left(\frac{\delta_{i,\mathcal{A}_0^i}}{|\mathcal{A}_0^i|}\right)^{\delta_{i,\mathcal{A}_0^i}} \\ & \times \left(1 - \frac{\delta_{i,\mathcal{A}_1^i}}{|\mathcal{A}_1^i|}\right)^{|\mathcal{A}_1^i| - \delta_{i,\mathcal{A}_1^i}} \left(\frac{\delta_{i,\mathcal{A}_1^i}}{|\mathcal{A}_1^i|}\right)^{\delta_{i,\mathcal{A}_1^i}}.\end{aligned}$$

Hence, the ranks of the objects can be estimated using (5.7) and the rank-based query selection can be performed using (5.8), without requiring any knowledge of the query noise parameters.

Unfortunately, it is not possible to extend the result of Proposition V.2 to this case. Yet, the experimental results in Section 5.4 demonstrate that the noise-independent rank-based algorithm performs comparably to the entropy-based algorithm, which requires knowledge of ν_0 and ν_1.

5.4 Experimental Evaluation

We compare the performance of the proposed rank-based algorithm with entropy-based active query selection, GBS, and random search, on 2 synthetic datasets, 1 semi-synthetic dataset, and 1 real dataset. GBS and random search serve as baselines and are not expected to perform well since GBS doesn't account for noise, and random search just selects queries at random.

The first two datasets are random bipartite networks (*Guillaume and Latapy*, 2004) generated using the standard Erdös-Rényi (ER) random network model and the Preferential Attachment (PA) random network model. The third dataset is a network topology built using the BRITE generator (*Medina et al.*, 2001), which simulates an Internet-like topology at the Autonomous Systems level. To generate a bipartite network of components and probes from the BRITE network, we used the approach described by *Rish et al.* (2005) and *Zheng et al.* (2005). The last dataset is the WISER database, which is a toxic chemical database describing the binary relation between 298 toxic chemicals and 79 acute symptoms (*Szczur and Mashayekhi*, 2005).

We generated a random network for each of the random network models considered, where each network consisted of around 200 objects and 300 queries. We generated a BRITE network consisting of 300 objects (components/computers) and around 350 queries (probes). For the synthetic datasets and WISER, we assumed a constant noise rate, and for the BRITE network, we considered the response dependent noise model described in Section 5.3.3.2. Here, we present the results under uniform prior where $\pi_i = 1/M$. We observed similar performance under non-uniform

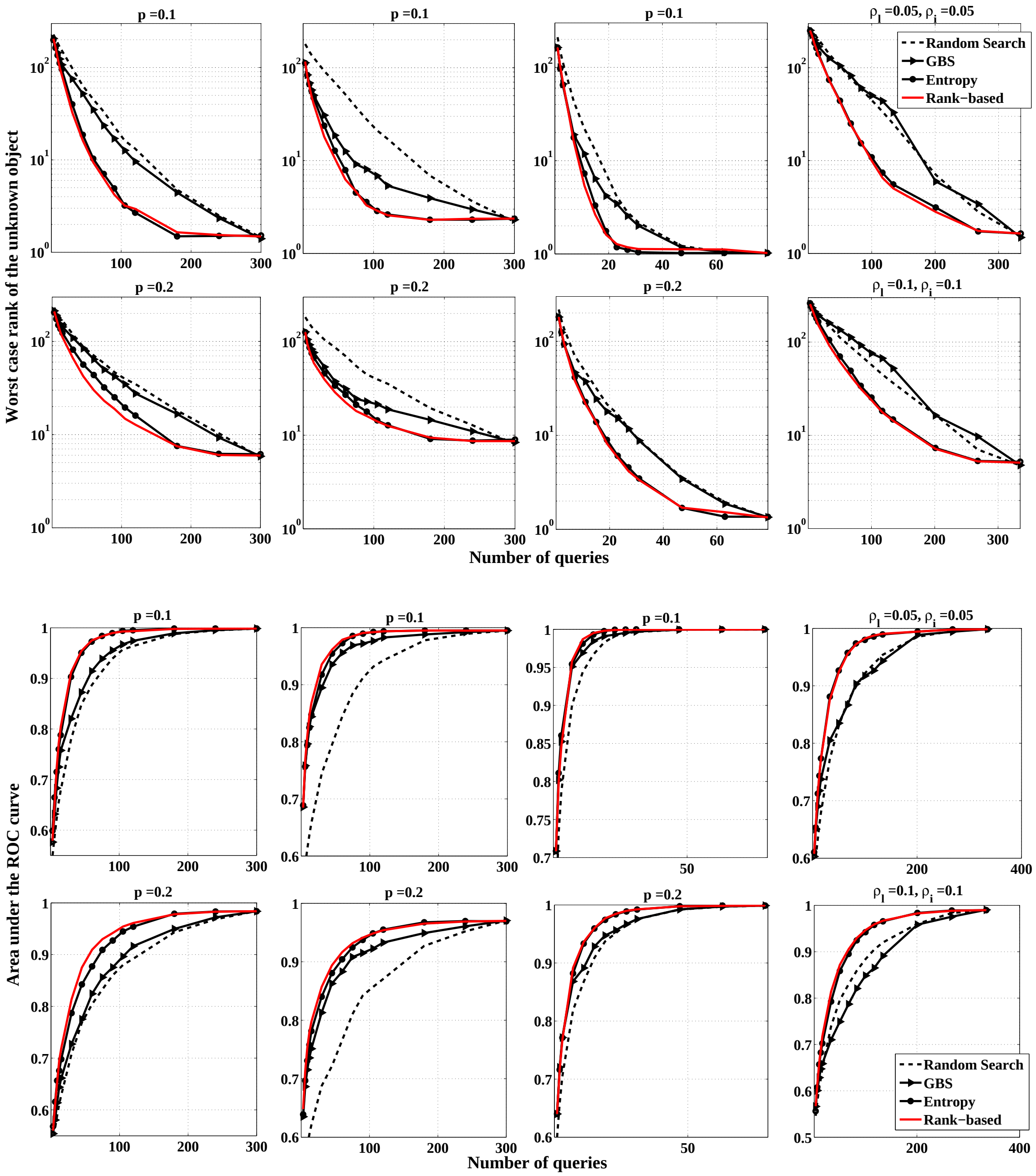


Figure 5.3: The first column corresponds to a dataset generated using the ER model, the second column corresponds to a dataset generated using the PA model, the third column corresponds to the WISER database and the last column corresponds to a BRITE network. In all the experiments, the rank-based algorithm has no knowledge of the noise parameters.

prior as shown in *Bellala et al.* (2011a).

Figure 5.3 shows the worst case rank of the unknown object θ and the area under the ROC curve as a function of the number of queries inputted. The ROC curve is generated as follows: After observing responses to a set of queries, the objects are ranked based on their posterior probabilities where ties involving objects with equal posterior probabilities are broken randomly, instead of a worst case ranking. Given such a ranking of the objects in Θ, the ROC curve can be obtained by varying the threshold t, where the states of the top t objects are declared as 1 and the rest 0 leading to a certain miss rate and false alarm rate.

Each curve in these figures is averaged over 500 random realizations, where each random realization corresponds to a random selection of $\theta \in \Theta$ and random generation of the noisy query responses. The plots in the first column correspond to a dataset generated using the ER model, the second column corresponds to the PA model, the third column corresponds to the WISER database, and the last column to a BRITE network. For the 2 random network models and BRITE, the results were observed to be consistent across different realizations of the underlying bipartite network.

For the ER, PA, and the WISER datasets, we consider two different values for the probability of error, $p = 0.1, 0.2$. The entropy-based query selection is performed assuming the knowledge of p, whereas the rank-based query selection is performed using only the fact that $p < \overline{p} = 0.5$. The BRITE networks are simulated using the QMR-DT noise model, where we considered the inhibition and the leak probabilities to be $(\rho_i, \rho_l) = (0.05, 0.05)$ and $(0.1, 0.1)$. This noise model reduces to that in Section 5.3.3.2 with $\nu_0 = \rho_l$ and $\nu_1 = (1 - \rho_l)\rho_i$. Once again, the entropy-based query selection is performed assuming the knowledge of ν_0 and ν_1, whereas the rank-based query selection is performed using only the fact that $\nu_0, \nu_1 \leq \overline{p} = 0.25$.

Finally, Figure 5.4 demonstrates the sensitivity of entropy-based query selection to mis-specification of the value of noise parameters. For the ER, PA and the WISER datasets, the true noise parameter is $p = 0.25$ while the under-estimated and the over-estimated curves are obtained using $p = 0.15$ and 0.4, respectively. For the

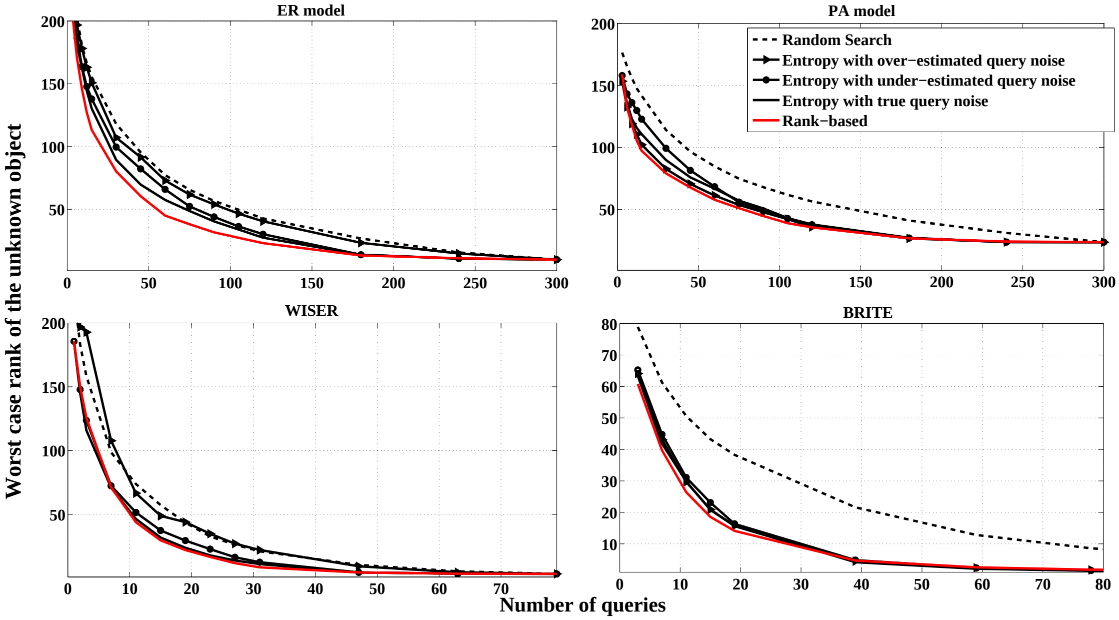


Figure 5.4: Demonstrates the sensitivity of entropy-based query selection to mis specification of the noise parameters

BRITE network, while the true noise parameters are $(0.1, 0.1)$, the other two curves are obtained using $(0.05, 0.05)$ and $(0.15, 0.15)$. Once again, the rank-based algorithm is performed without knowledge of the noise parameters. This demonstrates that the entropy-based query selection can perform poorly when the noise parameters are mis-specified.

In addition, these experiments demonstrate the competitive performance of the proposed rank-based active diagnosis algorithm to entropy-based active query selection, despite not having the knowledge of the underlying noise parameters.

CHAPTER VI

Multiple Fault Diagnosis

6.1 Introduction

In this chapter, we will consider a general version of the diagnosis problem studied in Chapter V, that arises in applications such as medical diagnosis (*Heckerman*, 1990; *Jaakkola and Jordan*, 1999), fault diagnosis in nuclear plants (*Santoso et al.*, 1999), computer networks (*Rish et al.*, 2005; *Zheng et al.*, 2005), and power-delivery systems (*Yongli et al.*, 2006). In these problems, more than one object could be of interest, i.e., more than one object could be in state 1, and the goal is to identify the binary states $\mathbf{X} = (X_1, \cdots, X_M)$ of all the objects based on the (noisy) responses to queries from the set Q.

For example, in the problem of medical diagnosis, the goal is to identify the presence/absence of a set of diseases based on the outcomes of medical tests. Similarly, in a fault diagnosis problem, the goal is to identify the state (faulty/working) of each component based on alarm/probe responses. For simplicity, we will refer to an object with state 1 as a fault in the rest of this chapter.

In recent years, this problem has been formulated as an inference problem on a Bayesian network, with the goal of assigning most likely states to unobserved object nodes based on the outcome of the query nodes. Hence, the goal of active diagnosis is to select queries sequentially so as to maximize the accuracy of diagnosis while minimizing the cost of querying.

In this context, *Zheng et al.* (2005) proposed the use of reduction in conditional entropy (equivalently, mutual information) as a measure to select the most informative subset of queries. They proposed an algorithm that uses the loopy belief propagation (BP) framework to select queries sequentially based on the gain in mutual information, given the observed responses to past queries. This algorithm, which they refer to as BPEA, requires one run of BP for each query selection. Finally, the objects are assigned the most likely states based on the outcome of the selected queries, using a MAP (maximum *a posteriori*) inference algorithm. Refer to Section 6.3.1 for more details.

However, there are two limitations with this approach. First, the MAP estimate may not equal the true state vector $\mathbf{X}$, either due to noise in the observed query responses or due to suboptimal convergence of the MAP inference algorithm. This leads to false alarm and miss rates that may not be tolerable for a given application.

The second issue is that BPEA does not scale to large networks, because the complexity of computing the approximate value of conditional entropy grows exponentially in the maximum degree of the underlying Bayesian network (see Section 6.3.1 for details). As we show in Section 6.4, it becomes intractable even in networks with a few thousand objects. In addition, since this approach relies on belief propagation (BP), it may suffer from the limitations of BP such as slow convergence or oscillation of the algorithm, especially when the prior fault probability is small (*Murphy et al.*, 1999). As we discuss below, the prior fault probability is indeed very low in most real-world diagnosis problems. These factors render BPEA impractical in many large scale, real-world applications.

We address these limitations by proposing an extension of our rank-based approach to the multi-fault scenario, where we output a ranked list of objects based on their posterior probabilities rather than their most likely states. Given such a ranked list, the object states can be estimated by choosing a threshold t, where the top t objects in the ranked list are declared as faults (i.e., state 1) and the remaining as 0. Varying the threshold t leads to a series of estimators with different false alarm and miss rates giving rise to a receiver operating characteristic (ROC) curve. The quality of the obtained ranking is then measured in terms of the area under this ROC curve (AUC). We show how to choose queries greedily such that the AUC, and thus, the quality of diagnosis, is maximized.

The rank-based approach is motivated by the fact that in many applications there is a domain expert who makes the final decision on the objects' states. Such a ranking can be useful to a domain expert who will use domain expertise and other sources of information to choose a threshold t that may lead to a permissible value of false alarm and miss rates for a given application.

To address the second limitation, we circumvent the use of BP in the query selection stage by making the simplifying assumption of a single fault, i.e., the state of only one object can be equal to 1. To be clear, we still intend to apply our algorithm when multiple faults are present; the single fault assumption is used in the design of the algorithm. This assumption is reasonable because the prior fault probability is quite low in many applications. For example, in the problem of fault diagnosis in computer networks, the prior probability of a router failing in any given hour is on the order of 10^{-6} (*Kandula et al.*, 2005). Similarly, in the disease diagnosis problem of QMR-DT, the prior probability of a disease being "present" is typically on the order of 10^{-3} (*Murphy et al.*, 1999).

We show that the AUC criterion can be optimized efficiently under a single-fault assumption. While other criteria such as mutual information can also be optimized efficiently under this assumption, we show that AUC is much more robust to violations of the single fault assumption, which are bound to happen in practice. Finally, we demonstrate through experiments on computer networks that the proposed query

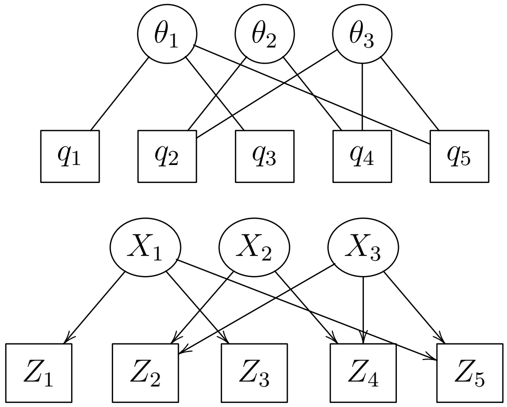


Figure 6.1: (top) A toy bipartite diagnosis graph, (bottom) A Bayesian network corresponding to the given BDG.

selection criterion can achieve performance close to that of BPEA *in a multi-fault setting*, while having a computational complexity that is orders less than that of BPEA (a reduction from exponential to near-quadratic). Thus, it is a fast and a reliable substitute for BPEA in large scale diagnosis problems. Finally, the work in this chapter is based on *Bellala et al.* (2011c).

6.2 Data Model

We will describe a general version of the data model considered in Chapter V. We will consider the input to an active diagnosis problem as a bipartite diagnosis graph denoting the relation between M different objects and N distinct queries, as shown in Figure 6.1.

We denote the state of each object (e.g., presence/absence of a disease) with a binary random variable X_i and the state of each query (i.e., the observed response to a query) by a binary random variable Z_j. Then, $\mathbf{X} = (X_1, \cdots, X_M)$ is a binary random vector denoting the states of all the objects, and $\mathbf{Z} = (Z_1, \cdots, Z_N)$ is a binary random vector denoting the responses to all the queries, where $\mathbf{x} \in \{0,1\}^M$ and $\mathbf{z} \in \{0,1\}^N$ correspond to realizations of $\mathbf{X}$ and $\mathbf{Z}$, respectively.

In addition, for any subset of queries $\mathcal{A} \subseteq \{1, \cdots, N\}$, we denote by $\mathbf{Z}_{\mathcal{A}}$ the random variables associated with those queries, e.g., if $\mathcal{A} = \{1, 4, 7\}$, then $\mathbf{Z}_{\mathcal{A}} = (Z_1, Z_4, Z_7)$. Also, for any query j, let $\mathbf{pa}_j$ denote the objects that are connected to it in the BDG. Then, $\mathbf{X}_{\mathbf{pa}_j}$ denotes the states of all the objects connected to query j, e.g., for query 2 in Figure 6.1, $\mathbf{X}_{\mathbf{pa}_2} = (X_2, X_3)$.

To specify the joint distribution of $(\mathbf{X}, \mathbf{Z}_{\mathcal{A}})$ for any $\mathcal{A}$, we define it in terms of a prior probability distribution on $\mathbf{X}$ and a conditional distribution on $\mathbf{Z}_{\mathcal{A}}$ given $\mathbf{X}$. To define the prior probability distribution on $\mathbf{X}$, we make the standard assumption that the object states are marginally independent, i.e., $\Pr(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{M} \Pr(X_i = x_i)$. Similarly, to define the conditional distribution on $\mathbf{Z}_{\mathcal{A}}$ given $\mathbf{X}$, we make the standard assumption that the observed responses to queries are conditionally independent given the states of the objects connected to them, i.e.,

$$\Pr(\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}} | \mathbf{X} = \mathbf{x}) = \prod_{j \in \mathcal{A}} \Pr(Z_j = z_j | \mathbf{x}_{\mathbf{pa}_j}).$$

These assumptions hold reasonably well in many practical applications. For example, in a fault diagnosis problem, it can be reasonable to assume that the components fail independently and that the alarm responses are conditionally independent given the states of the components they are connected to. These dependencies can be encoded by a Bayesian network as shown in Figure 6.1.

As mentioned in Section 5.2, in the ideal case when there is no noise, the observed response Z_j to query j is deterministic, and is given by the OR operation of the binary variables in $\mathbf{X}_{\mathbf{pa}_j}$, i.e., $Z_j = 1 \iff \exists\, i \in \mathbf{pa}_j$ s.t. $X_i = 1$. More generally, it is a noisy OR operation where the conditional distribution of Z_j given $\mathbf{x}_{\mathbf{pa}_j}$ can be defined using standard noise models such as the Y-model (*Le and Hadjicostis*, 2007) or the QMR-DT model (*Pearl*, 1988).

We derive the AUC based active diagnosis algorithm under this general probability model, and in Section 6.4, we demonstrate the performance of the proposed algorithm in the problem of fault diagnosis in computer networks under the QMR-DT noise model, where

$$\Pr(X_i = x) := (\alpha_i)^x (1 - \alpha_i)^{1-x}, \text{ and}$$
$$\Pr(Z_j = 0 | \mathbf{x}_{\mathbf{pa}_j}) := \rho_{0j} \textstyle\prod_{k \in \mathbf{pa}_j} \rho_{kj}^{x_k}.$$

Here, α_i is the prior fault probability, ρ_{kj} and $(1 - \rho_{0j})$ are the so-called inhibition and leak probabilities, respectively.

6.3 Active Diagnosis under Multiple Faults

As mentioned earlier, the approach in active diagnosis is to maximize some function $f(\mathbf{z}_{\mathcal{A}})$ which denotes the quality of the estimate of $\mathbf{X}$, subject to a constraint on the number of queries made, i.e.,

$$\max_{\mathcal{A} \subseteq \{1, \cdots, N\}} f(\mathbf{z}_{\mathcal{A}})$$
$$\text{s.t. } \ |\mathcal{A}| \leq k.$$

In general, finding an optimal solution to this problem is NP-hard (*Rish et al.*, 2005). Instead, the queries can be chosen sequentially by greedily maximizing the quality function, given the observed responses to the past queries, i.e.,

$$j^* := \arg\max_{j \notin \mathcal{A}} \ \mathbb{E}_{Z_j}[f(\mathbf{z}_{\mathcal{A}} \cup Z_j) - f(\mathbf{z}_{\mathcal{A}}) | \mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}] \tag{6.1}$$

where $\mathbf{z}_{\mathcal{A}} \cup Z_j$ denotes the observed responses to queries in $\mathcal{A} \cup \{j\}$.

6.3.1 Entropy-based Active Query Selection

Zheng et al. (2005) studied the problem of active diagnosis in the multiple fault scenario, where they used mutual information as a function to measure the quality of the estimate of the object states $\mathbf{X}$ based on the responses to queries in $\mathcal{A}$, i.e., $f(\mathbf{X}; \mathbf{Z}_{\mathcal{A}}) = I(\mathbf{X}; \mathbf{Z}_{\mathcal{A}}) := H(\mathbf{X}) - H(\mathbf{X}|\mathbf{Z}_{\mathcal{A}})$. However, the above optimization problem with mutual information as the quality function is NP-hard (*Rish et al.*, 2005). Alternatively, the greedy approach can be used to choose queries sequentially where given the observed responses $\mathbf{z}_{\mathcal{A}}$ to previously selected queries in $\mathcal{A}$, the next best query is chosen to be the one that maximizes the expected information gain as shown below,

$$\begin{aligned} j^* &= \arg\max_{j \notin \mathcal{A}} \ \mathbb{E}_{Z_j}[I(\mathbf{X}; \mathbf{Z}_{\mathcal{A}} \cup Z_j) - I(\mathbf{X}; \mathbf{Z}_{\mathcal{A}})|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}] \\ &= \arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}, z). \end{aligned} \tag{6.2}$$

In the multiple fault scenario, the conditional entropy term is given by

$$H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}, z) = - \sum_{\mathbf{x} \in \{0,1\}^M} \Pr(\mathbf{x}|\mathbf{z}_{\mathcal{A}}, z) \log_2 \Pr(\mathbf{x}|\mathbf{z}_{\mathcal{A}}, z).$$

Note that direct computation of the above expression is intractable. However, *Zheng et al.* (2005) note that under the independence assumptions of Section 6.2, the conditional entropy can be simplified such that the query selection criterion in (6.2) is reduced to

$$\begin{aligned} \arg\min_{j \notin \mathcal{A}} \quad & \Bigg[- \sum_{\mathbf{x}_{\mathbf{pa}_j}, z} \Pr(\mathbf{x}_{\mathbf{pa}_j}, z|\mathbf{z}_{\mathcal{A}}) \log_2 \Pr(Z_j = z|\mathbf{x}_{\mathbf{pa}_j}) \\ & \qquad + \sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) \log_2 \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) + \text{ const} \Bigg]. \end{aligned}$$

In addition, they propose an approximation algorithm that uses the loopy belief propagation (BP) infrastructure to compute the above expression, which they refer to as belief propagation for entropy approximation (BPEA). Interestingly, this algorithm requires only one run of loopy BP for each query selection. After observing responses $\mathbf{z}_{\mathcal{A}}$ to a set of queries in $\mathcal{A}$, the object states are then estimated to be

$$\mathbf{x}^{\text{MAP}} := \underset{\mathbf{x} \in \{0,1\}^M}{\arg\max} \; \Pr(\mathbf{X} = \mathbf{x} | \mathbf{z}_{\mathcal{A}}),$$

where the MAP estimator is obtained using a loopy version of the max-product algorithm. As far as we know, BPEA is the best known solution to the problem of active query selection in the multiple fault scenario.

However, this approach does not scale to large networks as BPEA involves a term whose computation grows exponentially in the number of parents to a query node. If m denotes the maximum number of parents to any query node, i.e., $m := \max_{j \in \{1,\cdots,N\}} |\mathbf{pa}_j|$, then the computational complexity of choosing a query using BPEA is $O(N2^m)$, thus making it intractable in networks where m is greater than 25 or even less, especially when real-time query selection is desired.

Recently, *Cheng et al.* (2010) proposed a speed up to query selection using BPEA by reducing the number of queries to be investigated at each stage. However, the exponential complexity still remains. Alternatively, we propose to assume a single fault in the query selection stage. As mentioned earlier, this assumption is motivated by the fact that in most diagnosis problems, the prior fault probability is very low. However, it is important for the query selection criterion to be robust to violations of the single fault assumption, as multiple faults could be present in practice. As we show in Section 6.3.2.2, entropy-based query selection is not robust to such violations, and can perform poorly when multiple faults are present.

In the next section, we derive a new query selection criterion that sequentially

	X_1	X_2	X_3	X_4	X_5
$\Pr(X_i = 1 \mid \mathbf{z}_{\mathcal{A}})$	0.3	0.15	0.35	0.15	0.05

Figure 6.2: A rank order corresponding to this example is $\mathbf{r} = (3, 1, 2, 4, 5)$.

chooses queries such that the area under the ROC curve of the rank-based output is maximized. We will show that the proposed query selection criterion can be implemented efficiently under a single fault assumption, and in Section 6.4, we will demonstrate how the AUC-based query selection can achieve performance close to that of BPEA, even when multiple faults occur, making it a viable substitute for BPEA in large scale networks.

6.3.2 AUC-based Active Query Selection

AUC has been used earlier as a performance criterion in the classification setting with decision tree classifiers (*Ferri et al.*, 2002; *Cortes and Mohri*, 2003) and boosting (*Long and Servedio*, 2007), in the problem of ranking (*Ataman et al.*, 2006), and in an active learning setting (*Culver et al.*, 2006). In all the earlier settings, the AUC of a classifier is estimated using the training data whose binary labels are known. However, in our setting, neither are the object states (binary labels) known nor does there exist any training data. Hence, we propose a simple estimator for the AUC based on the posterior probabilities of the object states. Specifically, we propose 3 variants of this estimator, and discuss some interesting properties of each of these variants in the two settings.

Given the observed responses $\mathbf{z}_{\mathcal{A}}$ to queries in $\mathcal{A}$, let the objects be ranked based on their posterior fault probabilities, i.e., $\Pr(X_i = 1|\mathbf{z}_{\mathcal{A}})$, where ties involving objects with the same posterior probability are broken randomly. Then, let $\mathbf{r} = (r(1), \cdots, r(M))$ denote the rank order of the objects, where $r(i)$ denotes the index of the ith ranked object. For example, a rank order corresponding to the toy example in Figure 6.2 is $\mathbf{r} = (3, 1, 2, 4, 5)$. Also, note that $\mathbf{r}$ depends on the queries chosen $\mathcal{A}$ and their observed responses $\mathbf{z}_{\mathcal{A}}$, though it is not explicitly shown in our notation.

Given this ranked list of objects, we get a series of estimators $\{\widehat{\mathbf{x}}^t\}_{t=0}^{M}$ for the object state vector $\mathbf{X}$, where $\widehat{\mathbf{x}}^t$ corresponds to the estimator which declares the states of the top t objects in the ranked list as 1 and the remaining as 0. For example, $\widehat{\mathbf{x}}^2 = (1, 0, 1, 0, 0)$ for the toy example shown in Figure 6.2.

These estimators have different false alarm and miss rates. The miss and false alarm rates associated with $\widehat{\mathbf{x}}^t$ are given by

$$\begin{aligned}\mathrm{MR}_t &= \frac{\sum_{\{i:\widehat{x}_i^t=0\}} \mathbf{I}\{X_i = 1\}}{\sum_{i=1}^{M} \mathbf{I}\{X_i = 1\}} = \frac{\sum_{i=t+1}^{M} \mathbf{I}\{X_{r(i)} = 1\}}{\sum_{i=1}^{M} \mathbf{I}\{X_i = 1\}},\\ \mathrm{FAR}_t &= \frac{\sum_{\{i:\widehat{x}_i^t=1\}} \mathbf{I}\{X_i = 0\}}{\sum_{i=1}^{M} \mathbf{I}\{X_i = 0\}} = \frac{\sum_{i=1}^{t} \mathbf{I}\{X_{r(i)} = 0\}}{\sum_{i=1}^{M} \mathbf{I}\{X_i = 0\}},\end{aligned}$$

where $\mathbf{I}\{E\}$ is an indicator function which takes the value 1 when the event E is true, and 0 otherwise.

However, since the true states of the objects are not known, the false alarm and the miss rates need to be estimated. Given the responses $\mathbf{z}_{\mathcal{A}}$ to queries in $\mathcal{A}$, these two error rates can be approximated by using the expected value of the numerator and denominator conditioned on these responses as shown below:

$$\widehat{\mathrm{MR}}_t(\mathbf{z}_{\mathcal{A}}) = \frac{\sum_{i=t+1}^{M} \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})}{\sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}})}, \tag{6.3a}$$

$$\widehat{\mathrm{FAR}}_t(\mathbf{z}_{\mathcal{A}}) = \frac{\sum_{i=1}^{t} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}})}{\sum_{i=1}^{M} \Pr(X_i = 0|\mathbf{z}_{\mathcal{A}})}. \tag{6.3b}$$

Using these estimates, the ROC curve can then be obtained by varying the threshold t from 0 to M leading to different false alarm and miss rates. For example, $\widehat{\mathbf{x}}^0$ which declares the states of all the objects to be equal to 0, has a false alarm rate of 0 and a miss rate of 1. On the other hand, $\widehat{\mathbf{x}}^M$ which declares the states of all objects as 1, has a false alarm rate of 1 with a miss rate of 0. The other estimators have false

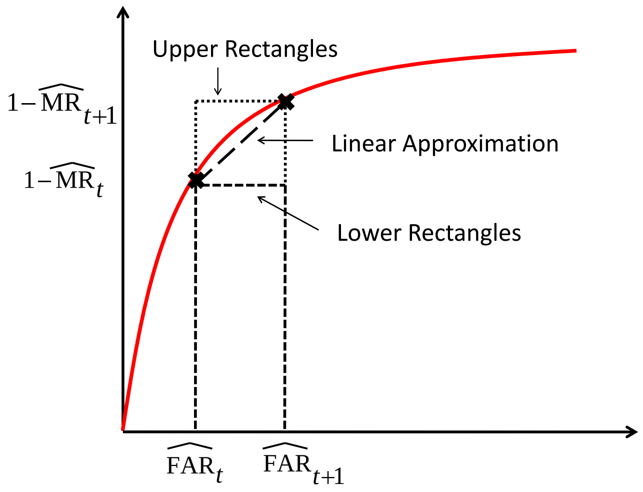

Upper Rectangles
Linear Approximation
Lower Rectangles
$1-\widehat{\mathrm{MR}}_{t+1}$
$1-\widehat{\mathrm{MR}}_{t}$
$\widehat{\mathrm{FAR}}_{t}$
$\widehat{\mathrm{FAR}}_{t+1}$

Figure 6.3: Demonstrates the different approximations for area under the ROC curve

alarm and miss rates that span the space between these two extremes.

Finally, the area under this ROC curve can be estimated using a piecewise approximation with either lower rectangles, upper rectangles or a linear approximation as shown in Figure 6.3. As we discuss later, each of these variants have some interesting properties in different settings, and as we show in Appendix D, the expected worst case rank criterion proposed in Chapter V is a special case of the AUC criterion. The expressions related to each of the three approximations are as given below:

$$\begin{aligned}
\underline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) &= \sum_{t=0}^{M-1}\left(1-\widehat{\mathrm{MR}}_t\right)\left(\widehat{\mathrm{FAR}}_{t+1}-\widehat{\mathrm{FAR}}_t\right)\\
\underline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) &= \sum_{t=0}^{M-1}\left(1-\widehat{\mathrm{MR}}_{t+1}\right)\left(\widehat{\mathrm{FAR}}_{t+1}-\widehat{\mathrm{FAR}}_t\right)\\
\underline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) &= \sum_{t=0}^{M-1}\left(1-\frac{\widehat{\mathrm{MR}}_t+\widehat{\mathrm{MR}}_{t+1}}{2}\right)\left(\widehat{\mathrm{FAR}}_{t+1}-\widehat{\mathrm{FAR}}_t\right),
\end{aligned}$$

where we dropped the dependence of $\widehat{\mathrm{MR}}_t$ and $\widehat{\mathrm{FAR}}_t$ on $\mathbf{z}_{\mathcal{A}}$ to avoid cramping. Further, noting that $\widehat{\text{FAR}}_M = 1$ and $\widehat{\text{FAR}}_0 = 0$, $\underline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}})$ can be re-written as

$$\begin{aligned}\underline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) &= \sum_{t=0}^{M-1}(1-\widehat{\text{MR}}_t)(\widehat{\text{FAR}}_{t+1}-\widehat{\text{FAR}}_t)\\ &= \widehat{\text{FAR}}_M - \widehat{\text{FAR}}_0 - \sum_{t=0}^{M-1}\widehat{\text{MR}}_t(\widehat{\text{FAR}}_{t+1}-\widehat{\text{FAR}}_t)\\ &= 1-\sum_{t=0}^{M-1}\widehat{\text{MR}}_t(\widehat{\text{FAR}}_{t+1}-\widehat{\text{FAR}}_t),\end{aligned}$$

where $\sum_{t=0}^{M-1}\widehat{\text{MR}}_t\ (\widehat{\text{FAR}}_{t+1}-\widehat{\text{FAR}}_t)$ corresponds to an estimate of the area *above* the ROC curve using lower rectangles, which we denote by $\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}})$. Similarly, the estimates of the area *above* the ROC curve using upper rectangles or a linear approximation are given by,

$$\begin{aligned}\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) &= \sum_{t=0}^{M-1}\widehat{\text{MR}}_{t+1}\left(\widehat{\text{FAR}}_{t+1}-\widehat{\text{FAR}}_t\right)\\ \overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) &= \sum_{t=0}^{M-1}\frac{\widehat{\text{MR}}_t+\widehat{\text{MR}}_{t+1}}{2}\left(\widehat{\text{FAR}}_{t+1}-\widehat{\text{FAR}}_t\right).\end{aligned}$$

Substituting the estimates for miss rate and false alarm rate from (6.3a) and (6.3b), the corresponding approximations for the area above the ROC curve are given by

$$\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) = \frac{\sum_{i=1}^{M}\sum_{j=i}^{M}\Pr(X_{r(i)}=0|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)}=1|\mathbf{z}_{\mathcal{A}})}{\sum_{i=1}^{M}\Pr(X_i=1|\mathbf{z}_{\mathcal{A}})\sum_{i=1}^{M}\Pr(X_i=0|\mathbf{z}_{\mathcal{A}})} \tag{6.4a}$$

$$\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) = \frac{\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}\Pr(X_{r(i)}=0|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)}=1|\mathbf{z}_{\mathcal{A}})}{\sum_{i=1}^{M}\Pr(X_i=1|\mathbf{z}_{\mathcal{A}})\sum_{i=1}^{M}\Pr(X_i=0|\mathbf{z}_{\mathcal{A}})} \tag{6.4b}$$

$$\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) = \frac{\sum_{i=1}^{M-1}\sum_{j=i+1}^{M}\Pr(X_{r(i)}=0|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)}=1|\mathbf{z}_{\mathcal{A}})}{\sum_{i=1}^{M}\Pr(X_i=1|\mathbf{z}_{\mathcal{A}})\sum_{i=1}^{M}\Pr(X_i=0|\mathbf{z}_{\mathcal{A}})} + \frac{\sum_{i=1}^{M}\Pr(X_i=0|\mathbf{z}_{\mathcal{A}})\Pr(X_i=1|\mathbf{z}_{\mathcal{A}})}{2\sum_{i=1}^{M}\Pr(X_i=1|\mathbf{z}_{\mathcal{A}})\sum_{i=1}^{M}\Pr(X_i=0|\mathbf{z}_{\mathcal{A}})}. \tag{6.4c}$$

Using the AUC as a quality function, the goal of active diagnosis is to maximize the accuracy of diagnosis given by the estimate of AUC, subject to a constraint on the number of queries made, i.e.,

$$\begin{gathered}\max\nolimits_{\mathcal{A}\subseteq\{1,\cdots,N\}} \underline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}}) \\ \text{s.t.}\quad |\mathcal{A}| \leq k,\end{gathered}$$

where $\underline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$ corresponds to an estimate of the AUC using any of the above approximations. More generically, in the rest of this paper, we will use the terms $\underline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$ and $\overline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$ to denote any of the above approximations for area under the ROC curve and area above the ROC curve, respectively.

Once again the above optimization problem is NP-hard. Hence, we resort to the greedy strategy, where substituting this quality function in (5.2), we get the criterion for greedily choosing a query to be

$$j^* = \arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \Pr(Z_j = z | \mathbf{z}_{\mathcal{A}}) \overline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}} \cup z). \tag{6.5}$$

Note that both the query selection criterion in (6.5) and the different approximations to the quality function $\overline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$ in (6.4) depend only on the posterior probabilities of unobserved nodes given the states of the observed nodes. Since these probabilities can be approximated using loopy belief propagation, the AUC-based active query selection can be performed using loopy BP similar to the entropy-based active query selection in BPEA.

However, our main focus is on active diagnosis for large scale networks where query selection using loopy BP is slow and possibly intractable. In the next section, we show how the proposed AUC-based query selection can be performed efficiently under a single fault assumption. In addition, we also argue that the AUC criterion under a single fault assumption is robust to violations of the assumption leading to a good choice of queries even when multiple faults are present.

6.3.2.1 Active Query Selection under Single Fault Assumption

In order to avoid the use of loopy BP in the query selection stage, we make the simplifying assumption of a single fault. Under this assumption, the object state vector $\mathbf{X}$ is restricted to belong to the set $\{\mathbb{I}_1, \cdots, \mathbb{I}_M\}$ in the query selection stage. This reduction in the state space of the object vector allows for query selection to be performed efficiently without the need for loopy belief propagation.

More specifically, the posterior probabilities required to choose queries sequentially in (6.5) can be computed as follows. Using the conditional independence assumption of Section 6.2, $\Pr(Z = z|\mathbf{z}_\mathcal{A})$ can be computed as

$$\Pr(Z = z|\mathbf{z}_\mathcal{A}) = \sum_{i=1}^{M} \Pr(Z = z|\mathbf{X} = \mathbb{I}_i) \Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{z}_\mathcal{A}),$$

where the posterior probabilities $\Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{z}_\mathcal{A})$ can be updated efficiently in $O(M)$ time as

$$\Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{Z}_\mathcal{A} = \mathbf{z}_\mathcal{A}, Z_j = z) = \frac{\Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{Z}_\mathcal{A} = \mathbf{z}_\mathcal{A}) \Pr(Z_j = z|\mathbf{X} = \mathbb{I}_i)}{\sum_{k=1}^{M} \Pr(\mathbf{X} = \mathbb{I}_k|\mathbf{Z}_\mathcal{A} = \mathbf{z}_\mathcal{A}) \Pr(Z_j = z|\mathbf{X} = \mathbb{I}_k)}.$$

Also, note that under a single fault assumption,

$$\sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_\mathcal{A}) = \sum_{i=1}^{M} \Pr(\mathbf{X} = \mathbb{I}_i|\mathbf{z}_\mathcal{A}) = 1, \quad \text{and} \tag{6.6a}$$

$$\sum_{i=1}^{M} \Pr(X_i = 0|\mathbf{z}_\mathcal{A}) = \sum_{i=1}^{M} 1 - \Pr(X_i = 1|\mathbf{z}_\mathcal{A}) = M - 1. \tag{6.6b}$$

Using these constraints, the estimates for the area above the ROC curve in (6.4) can be equivalently expressed as shown in the following proposition.

Proposition VI.1. *Under the single fault assumption, the estimates for the area above the ROC curve,* $\overline{\mathbf{A}}_{lr}(\mathbf{z}_\mathcal{A})$, $\overline{\mathbf{A}}_{l}(\mathbf{z}_\mathcal{A})$ *and* $\overline{\mathbf{A}}_{ur}(\mathbf{z}_\mathcal{A})$ *in* (6.4) *can be equivalently expressed as*

$$\overline{\mathbf{A}}_{lr}(\mathbf{z}_\mathcal{A}) = \frac{\sum_{i=1}^{M} \left[2i + \Pr(X_{r(i)} = 0|\mathbf{z}_\mathcal{A})\right] \Pr(X_{r(i)} = 1|\mathbf{z}_\mathcal{A})}{2(M-1)} \tag{6.7a}$$

$$\overline{\mathbf{A}}_{l}(\mathbf{z}_\mathcal{A}) = \frac{\sum_{i=1}^{M} [2i] \Pr(X_{r(i)} = 1|\mathbf{z}_\mathcal{A})}{2(M-1)} \tag{6.7b}$$

$$\overline{\mathbf{A}}_{ur}(\mathbf{z}_\mathcal{A}) = \frac{\sum_{i=1}^{M} \left[2i - \Pr(X_{r(i)} = 0|\mathbf{z}_\mathcal{A})\right] \Pr(X_{r(i)} = 1|\mathbf{z}_\mathcal{A})}{2(M-1)} \tag{6.7c}$$

Proof. Refer to Appendix D. □

Note from this result that given a ranked list of the objects along with their posterior probabilities, the complexity of estimating the area above the ROC curve $\overline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$ under a single fault assumption is $O(M)$. Since the posterior probabilities can also be updated efficiently in $O(M)$ time, the complexity of computing $\overline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$ is dominated by the complexity of sorting, which is $O(M \log M)$. Hence, the computational complexity of choosing a query at each stage using the AUC-based criterion under a single fault assumption is $O(NM \log M)$. This lets active query selection be tractable even in large networks.

However, as mentioned earlier, multiple faults could be present in practice, and hence it is important for a query selection criterion under a single fault assumption to be robust to violations of that assumption. In the next section, we will provide an intuitive explanation as to why the proposed AUC criterion makes a robust choice of queries under a single fault assumption, while the entropy-based criterion fails to do so. We will demonstrate the same through extensive experiments in Section 6.4.

Before we proceed to provide this intuitive explanation, we will briefly digress to mention another interesting property exhibited by AUC approximated using lower rectangles or a linear approximation as given by Theorem VI.2 below. In particular, it can be shown that these two AUC estimators are adaptive monotone (*Golovin and Krause*, 2010), i.e., the accuracy of diagnosis given by $\underline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}})$ or $\underline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}})$ is guaranteed to increase by acquiring more query information (equivalently, the area above the ROC curve given by $\overline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}})$ or $\overline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}})$ is guaranteed to decrease by acquiring more query information).

Theorem VI.2. *Under the single fault assumption, the quality function $\underline{\mathbf{A}}(\mathbf{Z}_{\mathcal{A}})$ estimated using either lower rectangles or a linear approximation, is adaptive monotone, i.e., $\forall \mathcal{A}' \subseteq \mathcal{A}$*

$$\underline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}'}) \leq \underline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}}) \quad \textit{and} \quad \underline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}'}) \leq \underline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}})$$

Proof. Refer to Appendix D. □

6.3.2.2 Robustness to violation of single-fault assumption

The following result helps to explain why entropy-based query selection under a single fault assumption performs poorly in a multi-fault setting.

Proposition VI.3. *Under the single fault assumption, along with the conditional independence assumption of Section 6.2, the entropy-based query selection criterion in* (6.2) *reduces to*

$$j^* := \arg\min_{j \notin \mathcal{A}} \sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) H\Big(\Pr(Z_j = 0|X_i = 1)\Big) - H\Big(\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})\Big) \tag{6.8}$$

where $H(p) := -p\log_2 p - (1-p)\log_2(1-p)$ *denotes the binary entropy function.*

Proof. Refer to Appendix D. □

As noted in (6.6a), under a single fault assumption, the posterior fault probabilities are constrained to sum to 1. Hence, objects with high posterior fault probability decrease the posterior fault probabilities of the remaining objects. Given this scenario, note from (6.8) in Proposition VI.3, that both the terms in this query selection criterion are highly dominated by the object(s) with high posterior fault probabilities (even the second term, since $\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}}) = \sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) \Pr(Z_j = 0|X_i = 1)$). Hence, at any given stage, the query chosen according to this criterion is highly biased towards objects that already have a high posterior fault probability. This could lead to a poor choice of queries as the objects with high posterior fault probability need not have their true states as 1, especially in the initial stages.

On the other hand, the AUC-based criterion under single fault assumption chooses queries at each stage by taking into account its effect on all the objects, leading to

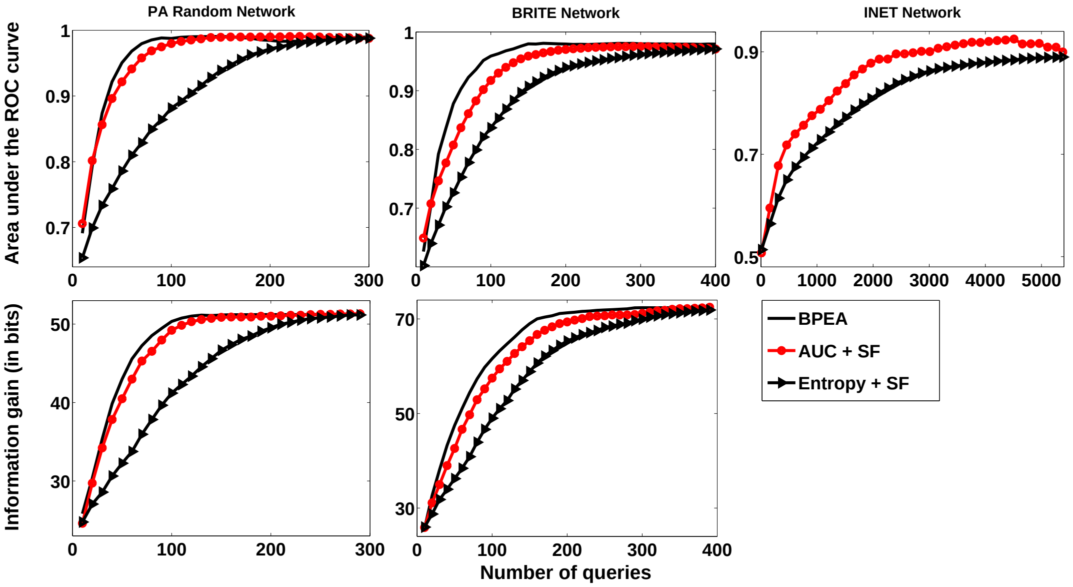


Figure 6.4: Demonstrates the competitive performance of the AUC-based query selection under single fault assumption to that of BPEA, while having a computational complexity that is orders less (near quadratic vs. exponential complexity of BPEA). On INET, we only compare AUC+SF with Entropy+SF as BPEA becomes slow and intractable.

a more balanced and informative choice of queries. This can be observed from the expressions for the estimators for area above the ROC curve in (6.7), where the object with the highest posterior fault probability $X_{r(1)}$ is assigned the least weight, with monotonically increasing weights as the posterior fault probability of the objects decreases. This forces to choose a query that takes in to consideration the effect on all the objects.

Though all three approximations for AUC are robust to violations of the single fault assumption, for reasons similar to the above and explained in detail in Appendix D, AUC approximated using upper rectangles turns out to be a better choice for active diagnosis of multiple faults under a single fault assumption.

6.4 Experimental Evaluation

We compare the performance of the proposed AUC-based active query selection under single fault assumption (AUC+SF) with BPEA and entropy-based active query selection under single fault assumption (Entropy+SF), on 1 synthetic dataset and 2 computer networks. Unlike *Zheng et al.* (2005) and *Cheng et al.* (2010) who only considered networks of size up to 500 components and 580 probes, here we also consider a large scale network.

The first dataset is a random bipartite diagnosis graph (*Guillaume and Latapy*, 2004) generated using the standard Preferential Attachment (PA) random graph model. The second and the third datasets are network topologies built using the BRITE (*Medina et al.*, 2001) and the INET (*Winick and Jamin*, 2002) generators, which simulate an Internet-like topology at the Autonomous Systems level. To generate a BDG of components and probes from these topologies, we used the approach described by *Rish et al.* (2005) and *Zheng et al.* (2005).

For the random graph model considered, we generated a random BDG consisting of 300 objects and 300 queries. We generated a BRITE network consisting of 300 components and around 400 probes, and an INET network consisting of 4000 components and 5380 probes. We consider the QMR-DT noise model described in Section 6.2; parameters are given below. We compare the 3 query selection criteria under 2 performance measures, AUC and Information gain.

Figure 6.4 compares their performance as a function of the number of queries inputted. Information gain is computed using BPEA. To compute the area under the ROC curve, we rank the objects based on their posterior fault probabilities that are computed using a single-fault assumption. Alternatively, note that these posterior probabilities could also be computed using BP for the PA and BRITE networks (BP is slow and intractable on the INET). For performance of the three query selection criteria under AUC computed with BP based rankings, refer *Bellala et al.* (2011c).

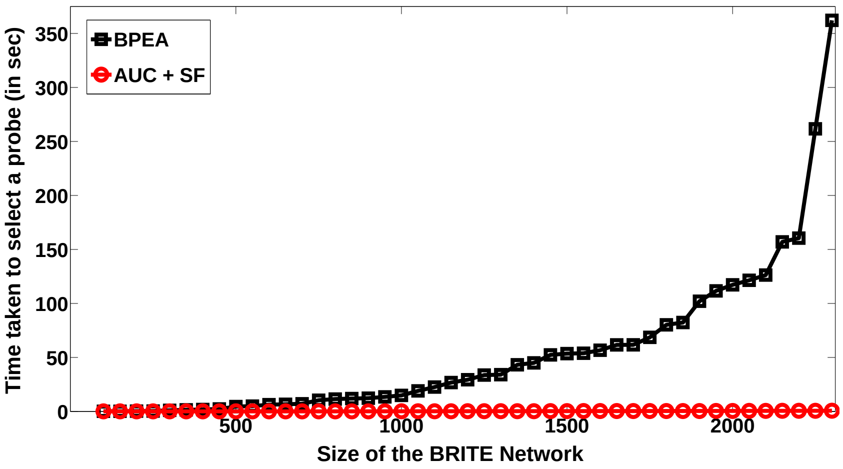


Figure 6.5: Comparison of time complexity of selecting a query using BPEA and AUC+SF.

Using this alternate approach does not change our conclusions.

We used the inference engines in the libDAI (*Mooij*, 2010) package for implementing BPEA and BP. However, BPEA (and BP) became slow and intractable on the INET, with BP often not converging and resulting in oscillations. Hence, on this network, we only compare the performance of AUC+SF and Entropy+SF based on the AUC criterion which is computed based on rankings obtained from posterior probabilities under a single-fault assumption.

The results in this figure correspond to a prior fault probability value of 0.03, with the leak and inhibition probabilities at 0.05[1]. Each curve in this figure is averaged over 200 random realizations, where each random realization corresponds to a random state of $\mathbf{X}$ and random generation of the noisy query responses. For the PA and BRITE models, the results were observed to be consistent across different realizations of the underlying bipartite network. For INET, we considered only one network with 25 probe stations.

Note from this figure that AUC+SF invariably performs better than Entropy+SF, and comparable to BPEA. We observed similar comparable performance of AUC+SF

[1] Refer *Bellala et al.* (2011c) for results on other values of prior, leak and inhibition probabilities

to that of BPEA, for different values of leak and inhibition probabilities, and other low values of prior fault probabilities (*Bellala et al.*, 2011c). In addition, note from Figure 6.5 that the time complexity of selecting a query grows exponentially for BPEA, whereas for AUC+SF, it grows near quadratically ($O(NM \log M)$) with the time taken to select a probe being less than 2 seconds even in networks with 2000 components.

These experiments demonstrate the competitive performance of AUC-based active query selection under single fault assumption to that of BPEA, besides having a computational complexity that is orders less than that of BPEA, demonstrating its potential as a fast and a reliable substitute for BPEA under low prior, in large scale diagnosis problems.

CHAPTER VII

Concluding Remarks

7.1 Summary of Contributions

In this thesis, we developed algorithms that broaden existing methods for object identification to incorporate factors that are specific to a given task and environment. These algorithms are greedy algorithms derived in a common, principled framework. Specifically, the proposed algorithms can be broadly classified into the following two frameworks.

Coding-Theoretic Framework: In Chapters III and IV, we considered extensions of the standard object identification problem to the group-based and the exponential cost settings. To address these problems, we show that a standard algorithm for object identification, known as the splitting algorithm or generalized binary search (GBS), can be viewed as a generalization of Shannon-Fano coding. We then use this interpretation to extend GBS to the group-based and the exponential cost settings. In particular, we prove the exact formulas for the cost function in each case that close the previously known lower bounds related to Shannon and Rényi entropies. These exact formulas are then optimized in a greedy, top-down manner to construct a decision tree. We demonstrate the improved performance of the proposed algorithms over GBS through simulations on a real world toxic chemical database known as WISER. We also develop a logarithmic approximation bound for group identification using the notion of adaptive submodularity.

Decision-Theoretic Framework: In Chapters V and VI, we study the problem of active diagnosis under persistent query noise in two different settings - single fault and multiple faults. In this context, we note that traditional approaches such as entropy-based active query selection have several drawbacks. Specifically, in the multiple fault scenario, an entropy-based active query selection algorithm such as BPEA relies on loopy belief propagation making it slow and intractable. Thus, we propose to make the simplifying assumption of a single fault in the query selection stage. Under this assumption, several query selection criterion can be implemented efficiently. However, we note that entropy-based active query selection under a single fault assumption performs poorly in a multiple fault setting. Hence, we propose a new query selection criterion, where the queries are selected sequentially such that the area under the ROC curve (AUC) of a rank-based output is maximized. We demonstrate the competitive performance of the proposed algorithm to BPEA in the context of fault diagnosis in computer networks. The competitive performance of the proposed algorithm, while having a computational complexity that is orders less than that of BPEA (near quadratic vs. the exponential complexity of BPEA), makes it a fast and a reliable substitute for BPEA in large scale diagnosis problems. Furthermore, we show that the proposed rank-based algorithm has another interesting feature in the single fault scenario, in that it does not require knowledge of the underlying query noise distribution. On the other hand, entropy-based active query selection requires knowledge of these noise parameters, and can be sensitive to mis-specification of these values.

7.2 Future Directions

While this work is a step towards making active diagnosis algorithms better suited for real-world diagnosis tasks, there are still several interesting issues that deserve to be examined in the future.

In the context of group identification, we showed that the query selection criterion in the proposed Group-GBS algorithm can be slightly modified such that it is adaptive submodular and strong adaptive monotone, thereby guaranteeing near-optimality. It would be interesting to see if similar modifications are possible with λ-GBS.

In the context of object identification under persistent query noise, we presented two algorithms - one that is near-optimal but restricts the number of persistent errors, and the other that neither restricts the number of persistent errors nor requires knowledge of the underlying noise distribution, but with no performance guarantees. Ideally, one would prefer to have an algorithm that exhibits both these properties, near-optimal and noise independent. This is still an open problem which deserves to be examined in the future.

In the context of active diagnosis under multiple objects, we made a significant progress in terms of the time-complexity while making little compromise on the performance. However, since the proposed approach is based on a single fault assumption in the query selection stage, it is only effective for prior fault probability values up to 0.1 (i.e., 10% of the components are faulty). As stated in Chapter VI, this was acceptable in applications such as disease diagnosis and fault diagnosis where the prior fault probability is very low. However, it is still an open problem to find a good, tractable solution in applications where the prior fault probability could be high.

In addition, it would be interesting to study the robustness of the proposed AUC-based algorithm when the faults are not independent as assumed in this thesis, but are correlated. Such as scenario can arise in applications such as fault diagnosis in power-delivery systems where the state of a component could effect that of the others. In this context, it would also be interesting to study the robustness of other variants of AUC, such as partial AUC. Moreover, *Gupta* (2001) showed that there exists a relation between the AUC function and weighted variants of Information gain. In future, these variants of Information gain should also be studied for their robustness properties.

Finally, in this thesis, we restricted our attention to applications where the responses to queries do not change over time. However, in applications such as disease diagnosis, symptoms might evolve over time, where a disease can be characterized by the sequence with which these symptoms emerge. In such applications, a “Plan ahead” sampling that incorporates any time information can be more effective.

APPENDICES

APPENDIX A

Appendix for Group Diagnosis

Proof of Theorem III.7

Let T_a denote a subtree from any node 'a' in the tree T and let $\mathcal{L}_a$ denote the set of leaf nodes in this subtree. Then, let μ_a denote the expected depth of the leaf nodes in this subtree, given by

$$\mu_a \quad = \quad \sum_{j \in \mathcal{L}_a} \frac{\pi_{\Theta_j}}{\pi_{\Theta_a}} d_j^a$$

where d_j^a corresponds to the depth of leaf node j in the subtree T_a, and let H_a denote the entropy of the probability distribution of the classes at the root node of the subtree T_a, i.e.

$$H_a = -\sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}$$

Now, we show using induction that for any subtree T_a in the tree T, the following relation holds

$$\pi_{\Theta_a}\mu_a - \pi_{\Theta_a}H_a = \sum_{s\in\mathcal{I}_a}\pi_{\Theta_s}\Big[1 - H(\rho_s) + \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i)\Big] - \sum_{s\in\mathcal{L}_a}\pi_{\Theta_s}I(\Theta_s)$$

where $\mathcal{I}_a, \mathcal{L}_a$ denotes the set of internal nodes and the set of leaf nodes in the subtree T_a, respectively.

The relation holds trivially for any subtree rooted at a leaf node of the tree T with both the left hand side and the right hand side of the expression equal to $-\pi_{\Theta_a}I(\Theta_a)$ (Note from (3.6) that $I(\Theta_a) = H_a$). Now, assume the above relation holds for the subtrees rooted at the left and right child nodes of node 'a'. Then, using Lemma A.1 we have

$$\begin{aligned}
\pi_{\Theta_a}[\mu_a - H_a] &= \pi_{\Theta_{l(a)}}[\mu_{l(a)} - H_{l(a)}] + \pi_{\Theta_{r(a)}}[\mu_{r(a)} - H_{r(a)}] \\
&\quad + \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}H(\rho_a^i)\right] \\
&= \sum_{s\in\mathcal{I}_{l(a)}}\pi_{\Theta_s}\left[1 - H(\rho_s) + \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i)\right] \\
&\quad + \sum_{s\in\mathcal{I}_{r(a)}}\pi_{\Theta_s}\left[1 - H(\rho_s) + \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i)\right] \\
&\quad + \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}H(\rho_a^i)\right] \\
&\quad - \sum_{s\in\mathcal{L}_{l(a)}}\pi_{\Theta_s}I(\Theta_s) - \sum_{s\in\mathcal{L}_{r(a)}}\pi_{\Theta_s}I(\Theta_s) \\
&= \sum_{s\in\mathcal{I}_a}\pi_{\Theta_s}\left[1 - H(\rho_s) + \sum_{i=1}^{m}\frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}}H(\rho_s^i)\right] - \sum_{s\in\mathcal{L}_a}\pi_{\Theta_s}I(\Theta_s)
\end{aligned}$$

thereby completing the induction. Finally, the result follows by applying the relation to the tree T whose probability mass at the root node, $\pi_{\Theta_a} = 1$.

Lemma A.1.

$$\begin{aligned}\pi_{\Theta_a}[\mu_a - H_a] &= \pi_{\Theta_{l(a)}}[\mu_{l(a)} - H_{l(a)}] + \pi_{\Theta_{r(a)}}[\mu_{r(a)} - H_{r(a)}] \\ &\quad + \pi_{\Theta_a}\left[1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)\right]\end{aligned}$$

Proof. We first note that $\pi_{\Theta_a}\mu_a$ for a subtree T_a can be decomposed as

$$\begin{aligned}\pi_{\Theta_a}\mu_a &= \sum_{j\in\mathcal{L}_a} \pi_{\Theta_j} d_j^a \\ &= \sum_{j\in\mathcal{L}_{l(a)}} \pi_{\Theta_j} d_j^a + \sum_{j\in\mathcal{L}_{r(a)}} \pi_{\Theta_j} d_j^a \\ &= \sum_{j\in\mathcal{L}_{l(a)}} \pi_{\Theta_j}(d_j^a - 1) + \sum_{j\in\mathcal{L}_{r(a)}} \pi_{\Theta_j}(d_j^a - 1) + \sum_{j\in\mathcal{L}_a} \pi_{\Theta_j} \\ &= \pi_{\Theta_{l(a)}}\mu_{l(a)} + \pi_{\Theta_{r(a)}}\mu_{r(a)} + \pi_{\Theta_a} \qquad \text{(A.1)}\end{aligned}$$

Similarly, $\pi_{\Theta_a} H_a$ can be decomposed as

$$\begin{aligned}
\pi_{\Theta_a} H_a &= \sum_{i=1}^{m} \pi_{\Theta_a^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_a^i}} \\
&= \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_a^i}} + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_a^i}} \\
&= \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_{l(a)}^i}} + \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_{l(a)}^i}}{\pi_{\Theta_a^i}} \\
&\quad + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_{r(a)}^i}} + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_{r(a)}^i}}{\pi_{\Theta_a^i}} \\
&\quad + \sum_{i=1}^{m} \pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{l(a)}}} + \sum_{i=1}^{m} \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{r(a)}}} \\
&= \pi_{\Theta_{l(a)}} H_{l(a)} + \pi_{\Theta_{r(a)}} H_{r(a)} \\
&\quad - \sum_{i=1}^{m} \left[\pi_{\Theta_{l(a)}^i} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_{l(a)}^i}} + \pi_{\Theta_{r(a)}^i} \log \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_{r(a)}^i}} \right] \\
&\quad + \left[\pi_{\Theta_{l(a)}} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{l(a)}}} + \pi_{\Theta_{r(a)}} \log \frac{\pi_{\Theta_a}}{\pi_{\Theta_{r(a)}}} \right] \\
&= \pi_{\Theta_{l(a)}} H_{l(a)} + \pi_{\Theta_{r(a)}} H_{r(a)} - \sum_{i=1}^{m} \pi_{\Theta_a^i} H(\rho_a^i) + \pi_{\Theta_a} H(\rho_a) \qquad \text{(A.2)}
\end{aligned}$$

The result follows from (A.1) and (A.2) above. □

Proof of Theorem III.8

From relation (A.2) in Lemma A.1, we have

$$H_a - \left[\frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} H_{l(a)} + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} H_{r(a)} \right] = - \left[-H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i) \right]$$

Thus, maximizing the impurity based objective function with entropy function as the impurity function is equivalent to minimizing the cost function $\Delta_a := 1 - H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i)$

Proof of Theorem III.9

Before we prove the result in Theorem III.9, we need to introduce some additional notation and review some definitions from *Golovin and Krause* (2010). Let $f : 2^Q \times \Theta \to \mathbb{R}_{\geq 0}$ be a utility/reward function that depends on the queries chosen and the unknown object $\theta \in \Theta$. For any $\mathcal{A} \subseteq \{1, \cdots, N\}$, let $Q_{\mathcal{A}}$ denote the subset of queries indexed by $\mathcal{A}$, and let $\mathbf{Z}_{\mathcal{A}}$ be a binary random vector denoting the responses to queries in $Q_{\mathcal{A}}$. In addition, given a tree T, let $Q(T, \theta_i)$ denote the queries made along the path from the root node to the leaf node terminating in object θ_i. Then, for any $S > 0$ that denotes the minimum desired reward, an optimal tree T^* is defined to be

$$T^* \in \arg\min_{T} \mathbb{E}[K(T)] \text{ such that } f(Q(T, \theta), \theta) \geq S, \ \forall \theta \in \Theta.$$

Finding an optimal tree T^* is NP-complete and hence we need to resort to greedy approaches.

Definition A.2. ***(Conditional Expected Marginal Gain)*** Given the observed responses $\mathbf{z}_{\mathcal{A}}$ to queries in $Q_{\mathcal{A}}$, the *conditional expected marginal gain* of choosing a new query $q \notin Q_{\mathcal{A}}$ is given by

$$\Delta(q|\mathbf{z}_{\mathcal{A}}) := \mathbb{E}_{\theta}[f(Q_{\mathcal{A}} \cup \{q\}, \theta) - f(Q_{\mathcal{A}}, \theta)|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}], \tag{A.3}$$

where the expectation is taken with respect to Π.

A greedy algorithm to solve the above optimization problem is to construct a decision tree in a top-down manner, where at each internal node, a query that maximizes $\Delta(q|\mathbf{z}_{\mathcal{A}})$, i.e. $\arg\max_{q \notin Q_{\mathcal{A}}} \Delta(q|\mathbf{z}_{\mathcal{A}})$ is chosen, where $Q_{\mathcal{A}}$ denotes the queries leading to that node with $\mathbf{z}_{\mathcal{A}}$ being the responses.

Definition A.3. ***(Strong Adaptive Monotonicity)*** A function $f : 2^Q \times \Theta \to \mathbb{R}_{\geq 0}$ is *strongly adaptive monotone* with respect to Π if, informally “selecting more queries never hurts" with respect to the expected reward. Formally, for all $Q_{\mathcal{A}} \subseteq Q$, all $q \notin Q_{\mathcal{A}}$ and all $z \in \{0, 1\}$ such that $\Pr(Z = z|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}) > 0$, we require

$$\mathbb{E}_{\theta}[f(Q_{\mathcal{A}}, \theta)|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}] \leq \mathbb{E}_{\theta}[f(Q_{\mathcal{A}} \cup \{q\}, \theta)|\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}}, Z = z]. \tag{A.4}$$

Definition A.4. ***(Adaptive Submodular)*** A function $f : 2^Q \times \Theta \rightarrow \mathbb{R}_{\geq 0}$ is adaptive submodular with respect to distribution Π if the conditional expected marginal gain of any fixed query does not increase as more queries are selected and their responses are observed. Formally, f is adaptive submodular w.r.t. Π if for all $Q_{\mathcal{A}}$ and $Q_{\mathcal{B}}$ such that $Q_{\mathcal{A}} \subseteq Q_{\mathcal{B}} \subseteq Q$ and for all $q \notin Q_{\mathcal{B}}$, we have

$$\Delta(q|\mathbf{z}_{\mathcal{B}}) \leq \Delta(q|\mathbf{z}_{\mathcal{A}}). \tag{A.5}$$

Theorem A.5. (*Golovin and Krause*, 2010) *Suppose $f : 2^Q \times \Theta \rightarrow \mathbb{R}_{\geq 0}$ is adaptive submodular and strongly adaptive monotone with respect to Π and there exists an S such that $f(Q, \theta) = S$ for all $\theta \in \Theta$. Let η be any value such that $f(Q_{\mathcal{A}}, \theta) > S - \eta$ implies $f(Q_{\mathcal{A}}, \theta) = S$ for all $Q_{\mathcal{A}} \subseteq Q$ and all θ. Let T^* be an optimal tree with the least expected depth and let $\widehat{T}$ be a suboptimal tree constructed using the greedy algorithm, then*

$$\mathbb{E}[K(\widehat{T})] \leq \mathbb{E}[K(T^*)] \left(\ln \left(\frac{S}{\eta} \right) + 1 \right) \tag{A.6}$$

Proof of Theorem III.9

Let the utility function f be defined as $f(Q_{\mathcal{A}}, \theta_i) := 1 - \pi_a^2 + \left(\pi_a^{k_i}\right)^2$, where π_a is the probability mass of the objects remaining after observing responses to queries in $Q_{\mathcal{A}}$ with θ_i as the unknown object, and k_i denoting the group to which θ_i belongs. As shown in Lemma A.6 below, substituting this utility function in (A.3), we get the conditional expected marginal gain to be $3\pi_{l(a)}\pi_{r(a)} - \sum_{i=1}^{m} 3\frac{\pi_a^i}{\pi_a}\pi_{l(a)}^i\pi_{r(a)}^i$, which is the greedy criterion for choosing queries at each internal node.

Now, note that $f(Q, \theta) = 1$, $\forall \theta \in \Theta$. Also, for any $Q_{\mathcal{A}} \subseteq Q$, if $f(Q_{\mathcal{A}}, \theta_i) > 1 - 3\pi_{\min}^2$, it implies $f(Q_{\mathcal{A}}, \theta_i) = 1$, hence $\eta = 3\pi_{\min}^2$. In addition, it follows from Lemma A.6 and Lemma A.7 below that the utility function f defined above is adaptive submodular and strongly adaptive monotone. Hence, the result follows from Theorem A.5.

Lemma A.6. *The utility function f defined above is adaptive submodular.*

Proof. Consider two subsets of Q such that $Q_{\mathcal{A}} \subseteq Q_{\mathcal{B}}$. Let $\mathbf{z}_{\mathcal{A}}, \mathbf{z}_{\mathcal{B}}$ denote the responses to the queries in $Q_{\mathcal{A}}$ and $Q_{\mathcal{B}}$, respectively. Then, we need to show that for any $q \notin Q_{\mathcal{B}}$, $\Delta(q|\mathbf{z}_{\mathcal{A}}) \geq \Delta(q|\mathbf{z}_{\mathcal{B}})$.

Let $\Theta_a \subseteq \Theta$ denote the set of objects whose responses to queries in $Q_{\mathcal{A}}$ are same as those in $\mathbf{z}_{\mathcal{A}}$. Then substituting $f(Q_{\mathcal{A}}, \theta) = 1 - \pi_a^2 + (\pi_a^i)^2$ in (A.3), we get

$$\begin{aligned}
\Delta(q|\mathbf{z}_{\mathcal{A}}) &= \sum_{i=1}^{m} \frac{\pi_{l(a)}^i}{\pi_a} \left[\pi_a^2 - \pi_{l(a)}^2 - (\pi_a^i)^2 + (\pi_{l(a)}^i)^2\right] \\
&\quad + \sum_{i=1}^{m} \frac{\pi_{r(a)}^i}{\pi_a} \left[\pi_a^2 - \pi_{r(a)}^2 - (\pi_a^i)^2 + (\pi_{r(a)}^i)^2\right] \\
&= \frac{\pi_{l(a)}}{\pi_a} \pi_{r(a)} (\pi_a + \pi_{l(a)}) - \sum_{i=1}^{m} \frac{\pi_{l(a)}^i}{\pi_a} \pi_{r(a)}^i (\pi_a^i + \pi_{l(a)}^i) \\
&\quad + \frac{\pi_{r(a)}}{\pi_a} \pi_{l(a)} (\pi_a + \pi_{r(a)}) - \sum_{i=1}^{m} \frac{\pi_{r(a)}^i}{\pi_a} \pi_{l(a)}^i (\pi_a^i + \pi_{r(a)}^i) \\
&= 3\pi_{l(a)}\pi_{r(a)} - \sum_{i=1}^{m} 3 \frac{\pi_a^i}{\pi_a} \pi_{l(a)}^i \pi_{r(a)}^i.
\end{aligned}$$

Similarly, let $\Theta_b \subseteq \Theta$ denote the set of objects whose responses to queries in $Q_{\mathcal{B}}$ are equal to those in $\mathbf{z}_{\mathcal{B}}$. Then, substituting $f(Q_{\mathcal{B}}, \theta) = 1 - \pi_b^2 + (\pi_b^i)^2$ in (A.3), we get $\Delta(q|\mathbf{z}_{\mathcal{B}}) = 3\pi_{l(b)}\pi_{r(b)} - \sum_{i=1}^{m} 3\frac{\pi_b^i}{\pi_b} \pi_{l(b)}^i \pi_{r(b)}^i$.

To prove f is adaptive submodular, we need to show that

$$\pi_{l(a)}\pi_{r(a)} - \sum_{i=1}^{m} \frac{\pi_a^i}{\pi_a} \pi_{l(a)}^i \pi_{r(a)}^i \geq \pi_{l(b)}\pi_{r(b)} - \sum_{i=1}^{m} \frac{\pi_b^i}{\pi_b} \pi_{l(b)}^i \pi_{r(b)}^i,$$

$$\implies \pi_a \pi_b \pi_{l(a)}\pi_{r(a)} - \sum_{i=1}^{m} \pi_a^i \pi_b \pi_{l(a)}^i \pi_{r(a)}^i \geq \pi_a \pi_b \pi_{l(b)}\pi_{r(b)} - \sum_{i=1}^{m} \pi_b^i \pi_a \pi_{l(b)}^i \pi_{r(b)}^i$$

Note that since $Q_{\mathcal{A}} \subseteq Q_{\mathcal{B}}$, $\Theta_b \subseteq \Theta_a$ and hence $\pi_b \leq \pi_a$, $\pi_b^i \leq \pi_a^i$, $\forall i \in \{1, \cdots, m\}$. For any query $q \notin Q_{\mathcal{B}}$, let $\Theta_{l(a)}$ and $\Theta_{r(a)}$ correspond to the objects in Θ_a that respond 0 and 1 to query q respectively. Similarly, let $\Theta_{l(b)}$ and $\Theta_{r(b)}$ correspond to the objects in Θ_b that respond 0 and 1 to query q respectively. Then, $\pi_{l(b)} \leq \pi_{l(a)}$, $\pi_{l(b)}^i \leq \pi_{l(a)}^i$, $\forall i$, and $\pi_{r(b)} \leq \pi_{r(a)}$, $\pi^i_{r(b)} \leq \pi^i_{r(a)}$, $\forall i$. Hence

$$\begin{align}
&\pi_a \pi_b \pi_{l(a)} \pi_{r(a)} - \sum_{i=1}^{m} \pi^i_a \pi_b \pi^i_{l(a)} \pi^i_{r(a)} \notag \\
&\quad = \pi_a \pi_b \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(a)} \pi^j_{r(a)} - \sum_{i=1}^{m} \pi^i_a \pi_b \pi^i_{l(a)} \pi^i_{r(a)} \notag \\
&\quad = \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} (\pi_a - \pi^i_a) \pi_b + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(a)} \pi^j_{r(a)} \tag{A.7a} \\
&\quad \geq \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} (\pi_a - \pi^i_a) \pi_b + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(b)} \pi^j_{r(b)} \tag{A.7b} \\
&\quad = \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} (\pi_a - \pi^i_a)(\pi_b - \pi^i_b) + \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} (\pi_a - \pi^i_a) \pi^i_b \notag \\
&\qquad\quad + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(b)} \pi^j_{r(b)} \tag{A.7c} \\
&\quad \geq \sum_{i=1}^{m} \pi^i_{l(b)} \pi^i_{r(b)} (\pi_a - \pi^i_a)(\pi_b - \pi^i_b) + \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} (\pi_a - \pi^i_a) \pi^i_b \notag \\
&\qquad\quad + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(b)} \pi^j_{r(b)} \tag{A.7d} \\
&\quad \geq \sum_{i=1}^{m} \pi^i_{l(b)} \pi^i_{r(b)} (\pi_a - \pi^i_a)(\pi_b - \pi^i_b) + \sum_{i=1}^{m} \pi^i_{l(b)} \pi^i_{r(b)} (\pi_b - \pi^i_b) \pi^i_a \notag \\
&\qquad\quad + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(b)} \pi^j_{r(b)} \tag{A.7e} \\
&\quad = \sum_{i=1}^{m} \pi^i_{l(b)} \pi^i_{r(b)} \pi_a (\pi_b - \pi^i_b) + \pi_a \pi_b \sum_{i \neq j} \pi^i_{l(b)} \pi^j_{r(b)} \notag \\
&\quad = \pi_a \pi_b \pi_{l(b)} \pi_{r(b)} - \sum_{i=1}^{m} \pi_a \pi^i_b \pi^i_{l(b)} \pi^i_{r(b)} \notag
\end{align}$$

where (A.7e) follows from (A.7d) since

$$\begin{aligned}
&\sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{r(a)} (\pi_a - \pi^i_a) \pi^i_b \\
&\quad = \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{l(b)} \pi^i_{r(a)} (\pi_a - \pi^i_a) + \sum_{i=1}^{m} \pi^i_{r(a)} \pi^i_{r(b)} \pi^i_{l(a)} (\pi_a - \pi^i_a) \\
&\quad \geq \sum_{i=1}^{m} \pi^i_{l(a)} \pi^i_{l(b)} \pi^i_{r(b)} (\pi_b - \pi^i_b) + \sum_{i=1}^{m} \pi^i_{r(a)} \pi^i_{r(b)} \pi^i_{l(b)} (\pi_b - \pi^i_b) \\
&\quad = \sum_{i=1}^{m} \pi^i_{l(b)} \pi^i_{r(b)} (\pi_b - \pi^i_b) \pi^i_a,
\end{aligned}$$

thus proving that f is adaptive submodular. □

Lemma A.7. *The utility function f as defined above is strongly adaptive monotone.*

Proof. Consider any subset of queries $Q_{\mathcal{A}} \subseteq Q$, and let $\mathbf{z}_{\mathcal{A}}$ denote the responses to these queries. Let Θ_a denote the set of objects whose responses to queries in $Q_{\mathcal{A}}$ are equal to those of $\mathbf{z}_{\mathcal{A}}$. For any query $q \notin Q_{\mathcal{A}}$, let $\Theta_{l(a)}$ and $\Theta_{r(a)}$ correspond to the objects in Θ_a that respond 0 and 1 to query q respectively.

For strong adaptive monotonicity, we need to show that

$$\begin{aligned}
1 - \pi_a^2 + \sum_{i=1}^{m} \frac{\left(\pi^i_a\right)^3}{\pi_a} &\leq 1 - \pi^2_{l(a)} + \sum_{i=1}^{m} \frac{\left(\pi^i_{l(a)}\right)^3}{\pi_{l(a)}}, \quad \text{if } \pi_{l(a)} > 0 \\
\text{and } 1 - \pi_a^2 + \sum_{i=1}^{m} \frac{\left(\pi^i_a\right)^3}{\pi_a} &\leq 1 - \pi^2_{r(a)} + \sum_{i=1}^{m} \frac{\left(\pi^i_{r(a)}\right)^3}{\pi_{r(a)}}, \quad \text{if } \pi_{r(a)} > 0.
\end{aligned}$$

We will show the first inequality, and the second inequality can be shown in a similar manner. Given $\pi_{l(a)} > 0$, we need to show that

$$\pi_a^3 \pi_{l(a)} - \pi^3_{l(a)} \pi_a \geq \sum_{i=1}^{m} \left(\pi^i_a\right)^3 \pi_{l(a)} - \left(\pi^i_{l(a)}\right)^3 \pi_a.$$

Note that

$$
\begin{aligned}
&\pi_a^3\pi_{l(a)} - \pi_{l(a)}^3\pi_a \\
&= \left(\pi_{l(a)} + \pi_{r(a)}\right)^3 \pi_{l(a)} - \pi_{l(a)}^3 \left(\pi_{l(a)} + \pi_{r(a)}\right) \\
&= \pi_{r(a)}^3\pi_{l(a)} + 3\pi_{l(a)}^2\pi_{r(a)}^2 + 2\pi_{l(a)}^3\pi_{r(a)} && \text{(A.8a)} \\
&\geq \sum_{i=1}^{m} \left[\left(\pi_{r(a)}^i\right)^3 \pi_{l(a)} + 3\pi_{l(a)}\pi_{l(a)}^i \left(\pi_{r(a)}^i\right)^2\right] + 2\pi_{l(a)}^3\pi_{r(a)} && \text{(A.8b)} \\
&= \sum_{i=1}^{m} \left[\left(\pi_{r(a)}^i\right)^3 \pi_{l(a)} + 3\pi_{l(a)}\pi_{l(a)}^i \left(\pi_{r(a)}^i\right)^2 - \pi_{r(a)} \left(\pi_{l(a)}^i\right)^3\right] \\
&\qquad + 2\pi_{l(a)}^3\pi_{r(a)} + \sum_{i=1}^{m} \pi_{r(a)} \left(\pi_{l(a)}^i\right)^3 && \text{(A.8c)} \\
&\geq \sum_{i=1}^{m} \left[\left(\pi_{r(a)}^i\right)^3 \pi_{l(a)} + 3\pi_{l(a)}\pi_{l(a)}^i \left(\pi_{r(a)}^i\right)^2 - \left(\pi_{l(a)}^i\right)^3 \pi_{r(a)} + 3\left(\pi_{l(a)}^i\right)^2 \pi_{r(a)}^i\pi_{l(a)}\right] \\
& && \text{(A.8d)} \\
&= \sum_{i=1}^{m} \left\{\pi_{l(a)} \left[\left(\pi_{l(a)}^i\right)^3 + 3\left(\pi_{l(a)}^i\right)^2 \pi_{r(a)}^i + 3\pi_{l(a)}^i \left(\pi_{r(a)}^i\right)^2 + \left(\pi_{r(a)}^i\right)^3\right]\right. \\
&\qquad \left. - \left(\pi_{l(a)}^i\right)^3 \pi_{l(a)} - \left(\pi_{l(a)}^i\right)^3 \pi_{r(a)}\right\} && \text{(A.8e)} \\
&= \sum_{i=1}^{m} \left(\pi_a^i\right)^3 \pi_{l(a)} - \left(\pi_{l(a)}^i\right)^3 \pi_a
\end{aligned}
$$

where (A.8b) follows from (A.8a) as $\pi_{r(a)}^3\pi_{l(a)}$ and $3\pi_{l(a)}\pi_{l(a)}\pi_{r(a)}^2$ has more non-negative terms than $\sum_{i=1}^{m} \left(\pi_{r(a)}^i\right)^3 \pi_{l(a)}$, $\sum_{i=1}^{m} 3\pi_{l(a)}\pi_{l(a)}^i \left(\pi_{r(a)}^i\right)^2$, respectively. Also (A.8d) follows from (A.8c) since

$$\begin{aligned}
&\pi_{r(a)}\left[2\pi_{l(a)}^3 + \sum_{i=1}^{m}\left(\pi_{l(a)}^i\right)^3\right] \\
&\quad = \pi_{r(a)}\left[\sum_{i=1}^{m} 3\left(\pi_{l(a)}^i\right)^3 + 6\sum_{i\neq j}\left(\pi_{l(a)}^i\right)^2 \pi_{l(a)}^j + 6\sum_{i\neq j\neq k}\pi_{l(a)}^i\pi_{l(a)}^j\pi_{l(a)}^k\right] \\
&\quad = \left(\sum_{h=1}^{m}\pi_{r(a)}^h\right)\left[\sum_{i=1}^{m} 3\left(\pi_{l(a)}^i\right)^3 + 6\sum_{i\neq j}\left(\pi_{l(a)}^i\right)^2 \pi_{l(a)}^j + 6\sum_{i\neq j\neq k}\pi_{l(a)}^i\pi_{l(a)}^j\pi_{l(a)}^k\right] \\
&\quad \geq 3\sum_{i=1}^{m}\left(\pi_{l(a)}^i\right)^3 \pi_{r(a)}^i + 3\sum_{i\neq j}\left(\pi_{l(a)}^i\right)^2 \pi_{r(a)}^i\pi_{l(a)}^j \\
&\quad = 3\pi_{l(a)}\sum_{i=1}^{m}\left(\pi_{l(a)}^i\right)^2 \pi_{r(a)}^i,
\end{aligned}$$

thus proving that f is strongly adaptive monotone. □

Proof of Theorem III.12

Let T_a denote a subtree from any node 'a' in the tree T and let $\mathcal{L}_a$ denote the set of leaf nodes in this subtree. Then, let μ_a denote the expected number of queries required to identify the group of an object terminating in a leaf node of this subtree, given by

$$\mu_a \quad = \quad \sum_{j\in\mathcal{L}_a}\frac{\pi_{\Theta_j}}{\pi_{\Theta_a}}\tilde{p}_j^a d_j^a$$

where $d_j^a, \tilde{p}_j^a$ denotes the depth of leaf node j in the subtree T_a and the probability of reaching that leaf node given $\theta \in \Theta_j$, respectively, and let H_a denote the entropy of the probability distribution of the object groups at the root node of this subtree, i.e.

$$H_a = -\sum_{i=1}^{m}\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}\log\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}$$

Now, we show using induction that for any subtree T_a in the tree T, the following relation holds

$$\pi_{\Theta_a}\mu_a - \pi_{\Theta_a}H_a = \sum_{s\in\mathcal{I}_a} \tilde{p}^a_s \pi_{\Theta_s}\left\{1 - \sum_{q\in Q^{z_s}} p_{z_s}(q)\left[H(\rho_s(q)) - \sum_{i=1}^{m}\frac{\pi_{\Theta^i_s}}{\pi_{\Theta_s}}H(\rho^i_s(q))\right]\right\}$$

where $\mathcal{I}_a$ denotes the set of internal nodes in the subtree T_a.

The relation holds trivially for any subtree rooted at a leaf node of the tree T with both the left hand side and the right hand side of the expression being equal to 0. Now, assume the above relation holds for all subtrees rooted at the child nodes of node 'a'. Note that node 'a' has a set of left and right child nodes, each set corresponding to one query from the query group selected at that node. Then, using the decomposition in Lemma A.1 on each query from this query group, we have

$$\begin{aligned}
1\cdot\pi_{\Theta_a}[\mu_a - H_a] &= \sum_{q\in Q^{z_a}} p_{z_a}(q)\pi_{\Theta_a}[\mu_a - H_a] \\
&= \sum_{q\in Q^{z_a}} p_{z_a}(q)\Big\{\pi_{\Theta_{l^q(a)}}[\mu_{l^q(a)} - H_{l^q(a)}] + \pi_{\Theta_{r^q(a)}}[\mu_{r^q(a)} - H_{r^q(a)}] \\
&\quad + \pi_{\Theta_a}\left[1 - H(\rho_a(q)) - \sum_{i=1}^{m}\frac{\pi_{\Theta^i_a}}{\pi_{\Theta_a}}H(\rho^i_a(q))\right]\Bigg\} \\
&= \sum_{q\in Q^{z_a}} p_{z_a}(q)\Big\{\pi_{\Theta_{l^q(a)}}[\mu_{l^q(a)} - H_{l^q(a)}] + \pi_{\Theta_{r^q(a)}}[\mu_{r^q(a)} - H_{r^q(a)}]\Big\} \\
&\quad + \pi_{\Theta_a}\left\{1 - \sum_{q\in Q^{z_a}} p_{z_a}(q)\left[H(\rho_a(q)) - \sum_{i=1}^{m}\frac{\pi_{\Theta^i_a}}{\pi_{\Theta_a}}H(\rho^i_a(q))\right]\right\}
\end{aligned}$$

where $l^q(a), r^q(a)$ correspond to the left and right child of node 'a' when query q is chosen from the query group and $\mu_{l^q(a)}, \pi_{\Theta_{l^q(a)}}, H_{l^q(a)}$ correspond to the expected depth of a leaf node in the subtree $T_{l^q(a)}$, probability mass of the objects at the root node of this subtree, and the entropy of the probability distribution of the objects at the root node of this subtree respectively. Now, using the induction hypothesis, we get

$$\begin{aligned}
&\pi_{\Theta_a}\mu_a - \pi_{\Theta_a} H_a \\
&\quad = \sum_{q\in Q^{z_a}} p_{z_a}(q) \left\{ \sum_{s\in \mathcal{I}_{l^q(a)}} \tilde{p}_s^{l^q(a)} \pi_{\Theta_s} \left[1 - \sum_{q\in Q^{z_s}} p_{z_s}(q) \left(H(\rho_s(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i(q))\right)\right]\right\} \\
&\qquad + \sum_{q\in Q^{z_a}} p_{z_a}(q) \left\{ \sum_{s\in \mathcal{I}_{r^q(a)}} \tilde{p}_s^{r^q(a)} \pi_{\Theta_s} \left[1 - \sum_{q\in Q^{z_s}} p_{z_s}(q) \left(H(\rho_s(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i(q))\right)\right]\right\} \\
&\qquad + \pi_{\Theta_a} \left\{1 - \sum_{q\in Q^{z_a}} p_{z_a}(q) \left[H(\rho_a(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i(q))\right]\right\} \\
&\quad = \sum_{s\in\mathcal{I}_a} \tilde{p}_s^a \pi_{\Theta_s} \left\{1 - \sum_{q\in Q^{z_s}} p_{z_s}(q) \left[H(\rho_s(q)) - \sum_{i=1}^{m} \frac{\pi_{\Theta_s^i}}{\pi_{\Theta_s}} H(\rho_s^i(q))\right]\right\}
\end{aligned}$$

thereby completing the induction. Finally, the result follows by applying the relation to the subtree rooted at the root node of T, whose probability mass $\pi_{\Theta_a} = 1$.

Miscellanies

Reduction factor calculation in the persistent noise model

At any internal node $a \in \mathcal{I}$ in a tree, let δ_i^a denote the Hamming distance between the query responses up to this internal node (Q_a) and the true responses of object θ_i to those queries. Also, let n_a denote the number of queries from the set of $N\nu$ queries (that were prone to error) in the set $Q \backslash Q_a$ and for a query $q \in Q \backslash Q_a$, denote by $b_i(q)$ the binary response of object θ_i to that query. Denote by the set $I^a = \{i : \delta_i^a \leq \epsilon'\}$, the object groups with non-zero number of objects at this internal node. All the formulas below come from routine calculations based on probability model 2.

For a query $q \in Q \setminus Q_a$, that is not prone to error, the reduction factor and the group reduction factors generated by choosing that query at node 'a' are as follows. The group reduction factor of any group $i \in I^a$ is equal to 1 and the reduction factor is given by

$$\begin{aligned}
\rho_a &= \frac{\max\{A,B\}}{\sum\limits_{i\in I_0^a\bigcap I_1^a}\pi_i\left[\sum\limits_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}\right]},\\
A &= \sum_{i\in I_0^a}\pi_i\left[\sum_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}\right],\\
B &= \sum_{i\in I_1^a}\pi_i\left[\sum_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}\right],
\end{aligned}$$

where $I_0^a = \{i\in I^a : b_i(q)=0\}$, $I_1^a = \{i\in I^a : b_i(q)=1\}$ and $\tau_i^a = \min(n_a, \epsilon' - \delta_i^a)$.

In addition, for a query $q\in Q\setminus Q_a$ that is prone to error, denote by $\delta_i^{l(a)}, \delta_i^{r(a)}$ the Hamming distance between the user responses to queries up to the left and right child node of node 'a' with query q chosen at node 'a', and the true responses of object θ_i to those queries. In particular, $\delta_i^{l(a)} = \delta_i^a + |b_i(q)-0|$ and $\delta_i^{r(a)} = \delta_i^a + |b_i(q)-1|$. Then, the reduction factor and the group reduction factors generated by choosing this query at node 'a' are as follows. The group reduction factor of a group $i\in I^a$ whose $\delta_i^a = \epsilon'$ is equal to 1 and that of a group whose $\delta_i^a < \epsilon'$ is given by

$$\begin{aligned}
\rho_a^i &= \frac{\max\{A,B\}}{\sum\limits_{e=0}^{\tau_i^a}\binom{n_a}{e}p^{e+\delta_i^a}(1-p)^{N\nu-e-\delta_i^a}},\\
A &= \sum_{e=0}^{\tau_i^{l(a)}}\binom{n_a-1}{e}p^{e+\delta_i^{l(a)}}(1-p)^{N\nu-e-\delta_i^{l(a)}},\\
B &= \sum_{e=0}^{\tau_i^{r(a)}}\binom{n_a-1}{e}p^{e+\delta_i^{r(a)}}(1-p)^{N\nu-e-\delta_i^{r(a)}},
\end{aligned}$$

where $\tau_i^{l(a)} = \min(n_a-1, \epsilon'-\delta_i^{l(a)})$ and $\tau_i^{r(a)} = \min(n_a-1, \epsilon'-\delta_i^{r(a)})$, and the reduction factor is given by

$$\begin{aligned}
\rho_a &= \frac{\max\{A,B\}}{\sum_{i\in I^a} \pi_i \left[\sum_{e=0}^{\tau_i^a} \binom{n_a}{e} p^{e+\delta_i^{r(a)}} (1-p)^{N\nu-e-\delta_i^{r(a)}} \right]}, \\
A &= \sum_{i\in I^{l(a)}} \pi_i \left[\sum_{e=0}^{\tau_i^{l(a)}} \binom{n_a-1}{e} p^{e+\delta_i^{l(a)}} (1-p)^{N\nu-e-\delta_i^{l(a)}} \right], \\
B &= \sum_{i\in I^{r(a)}} \pi_i \left[\sum_{e=0}^{\tau_i^{r(a)}} \binom{n_a-1}{e} p^{e+\delta_i^{r(a)}} (1-p)^{N\nu-e-\delta_i^{r(a)}} \right],
\end{aligned}$$

APPENDIX B

Appendix for Diagnosis under Exponential Query costs

Proof of Theorem IV.4

Define two new functions $\widetilde{L}_\lambda$ and $\widetilde{H}_\alpha$ as

$$\begin{aligned}\widetilde{L}_\lambda &:= \frac{1}{\lambda - 1}\left[\sum_{j\in\mathcal{L}} \pi_{\Theta_j}\lambda^{d_j} - 1\right] = \sum_{j\in\mathcal{L}} \pi_{\Theta_j}\left[\sum_{k=0}^{d_j-1}\lambda^k\right]\\ \widetilde{H}_\alpha &:= 1 - \frac{1}{\left(\sum_{i=1}^m \pi_{\Theta^i}^\alpha\right)^{\frac{1}{\alpha}}}\end{aligned}$$

Noting that the cost function $L_\lambda(\Pi)$ can be written as,

$$L_\lambda(\Pi) = \log_\lambda\left(\sum_{j\in\mathcal{L}} \pi_{\Theta_j}\lambda^{d_j}\right),$$

the new function $\widetilde{L}_\lambda$ can be related to the cost function $L_\lambda(\Pi)$ as

$$\lambda^{L_\lambda(\Pi)} = (\lambda - 1)\widetilde{L}_\lambda + 1 \tag{B.1}$$

Similarly, $\widetilde{H}_\alpha$ is related to the α-Rényi entropy $H_\alpha(\Pi_{\mathbf{y}})$ as

$$H_\alpha(\Pi_{\mathbf{y}}) = \frac{1}{1-\alpha}\log_2 \sum_{i=1}^{m} \pi_{\Theta^i}^{\alpha} = \frac{1}{\alpha \log_2 \lambda}\log_2 \sum_{i=1}^{m} \pi_{\Theta^i}^{\alpha} = \log_\lambda \left(\sum_{i=1}^{m} \pi_{\Theta^i}^{\alpha}\right)^{\frac{1}{\alpha}} \tag{B.2a}$$

$$\implies \lambda^{H_\alpha(\Pi_{\mathbf{y}})} = \left(\sum_{i=1}^{m} \pi_{\Theta^i}^{\alpha}\right)^{\frac{1}{\alpha}} = \left(\sum_{i=1}^{m} \pi_{\Theta^i}^{\alpha}\right)^{\frac{1}{\alpha}} \widetilde{H}_\alpha + 1 \tag{B.2b}$$

where we use the definition of α, i.e., $\alpha = \frac{1}{1+\log_2 \lambda}$ in (B.2a).

Now, we note from Lemma B.1 that $\widetilde{L}_\lambda$ can be decomposed as

$$\widetilde{L}_\lambda = \sum_{a \in \mathcal{I}} \lambda^{d_a} \pi_{\Theta_a}$$

$$\implies \lambda^{L_\lambda(\Pi)} = 1 + \sum_{a \in \mathcal{I}} (\lambda - 1)\lambda^{d_a} \pi_{\Theta_a} \tag{B.3}$$

where d_a denotes the depth of internal node 'a' in the tree T. Similarly, note from Lemma B.2 that $\widetilde{H}_\alpha$ can be decomposed as

$$\widetilde{H}_\alpha = \frac{1}{\left(\sum_{i=1}^{m} \pi_{\Theta^i}^{\alpha}\right)^{\frac{1}{\alpha}}} \sum_{a \in \mathcal{I}} \left[\pi_{\Theta_a}\mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]$$

$$\implies \lambda^{H_\alpha(\Pi_{\mathbf{y}})} = 1 + \sum_{a \in \mathcal{I}} \left[\pi_{\Theta_a}\mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_\alpha(\Theta_{r(a)})\right]. \tag{B.4}$$

Finally, the result follows from (B.3) and (B.4) above.

Lemma B.1. *The function $\widetilde{L}_\lambda$ can be decomposed over the internal nodes in a tree T, as*

$$\widetilde{L}_\lambda = \sum_{a \in \mathcal{I}} \lambda^{d_a} \pi_{\Theta_a}$$

where d_a denotes the depth of internal node $a \in \mathcal{I}$ and π_{Θ_a} is the probability mass of the objects at that node.

Proof. Let T_a denote a subtree from any internal node 'a' in the tree T and let $\mathcal{I}_a, \mathcal{L}_a$

denote the set of internal nodes and leaf nodes in the subtree T_a, respectively. Then, define $\widetilde{L}_\lambda^a$ in the subtree T_a to be

$$\widetilde{L}_\lambda^a = \textstyle\sum_{j \in \mathcal{L}_a} \frac{\pi_{\Theta_j}}{\pi_{\Theta_a}} \left[\sum_{k=0}^{d_j^a - 1} \lambda^k\right]$$

where d_j^a denotes the depth of leaf node $j \in \mathcal{L}_a$ in the subtree T_a.

Now, we show using induction that for any subtree T_a in the tree T, the following relation holds

$$\pi_{\Theta_a} \widetilde{L}_\lambda^a = \sum_{s \in \mathcal{I}_a} \lambda^{d_s^a} \pi_{\Theta_s} \tag{B.5}$$

where d_s^a denotes the depth of internal node $s \in \mathcal{I}_a$ in the subtree T_a.

The relation holds trivially for any subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose both child nodes terminate as leaf nodes, with both the left hand side and the right hand side of the expression equal to π_{Θ_a}. Now, consider a subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose left child (or right child) alone terminates as a leaf node. Assume that the above relation holds true for the subtree rooted at the right child of node 'a'. Then,

$$\begin{aligned}
\pi_{\Theta_a}\widetilde{L}^a_\lambda &= \sum_{j\in\mathcal{L}_a}\pi_{\Theta_j}\left[\sum_{k=0}^{d_j^a-1}\lambda^k\right] \\
&= \sum_{\{j\in\mathcal{L}_a:d_j^a=1\}}\pi_{\Theta_j} + \sum_{\{j\in\mathcal{L}_a:d_j^a>1\}}\pi_{\Theta_j}\left[\sum_{k=0}^{d_j^a-1}\lambda^k\right] \\
&= \pi_{\Theta_{l(a)}} + \sum_{\{j\in\mathcal{L}_a:d_j^a>1\}}\pi_{\Theta_j}\left[1+\lambda\sum_{k=0}^{d_j^a-2}\lambda^k\right] \\
&= \pi_{\Theta_a} + \lambda\sum_{j\in\mathcal{L}_{r(a)}}\pi_{\Theta_j}\left[\sum_{k=0}^{d_j^{r(a)}-1}\lambda^k\right] \\
&= \pi_{\Theta_a} + \lambda\sum_{s\in\mathcal{I}_{r(a)}}\lambda^{d_s^{r(a)}}\pi_{\Theta_s}
\end{aligned}$$

where the last step follows from the induction hypothesis. Finally, consider a subtree T_a rooted at an internal node $a\in\mathcal{I}$ whose neither child node terminates as a leaf node. Assume that the relation in (B.5) holds true for the subtrees rooted at its left and right child nodes. Then,

$$\begin{aligned}
\pi_{\Theta_a}\widetilde{L}^a_\lambda &= \sum_{j\in\mathcal{L}_a}\pi_{\Theta_j}\left[\sum_{k=0}^{d_j^a-1}\lambda^k\right] \\
&= \sum_{j\in\mathcal{L}_{l(a)}}\pi_{\Theta_j}\left[1+\lambda\sum_{k=0}^{d_j^a-2}\lambda^k\right] + \sum_{j\in\mathcal{L}_{r(a)}}\pi_{\Theta_j}\left[1+\lambda\sum_{k=0}^{d_j^a-2}\lambda^k\right] \\
&= \pi_{\Theta_a} + \lambda\sum_{j\in\mathcal{L}_{l(a)}}\pi_{\Theta_j}\left[\sum_{k=0}^{d_j^{l(a)}-1}\lambda^k\right] + \lambda\sum_{j\in\mathcal{L}_{r(a)}}\pi_{\Theta_j}\left[\sum_{k=0}^{d_j^{r(a)}-1}\lambda^k\right] \\
&= \pi_{\Theta_a} + \lambda\left[\sum_{s\in\mathcal{I}_{l(a)}}\lambda^{d_s^{l(a)}}\pi_{\Theta_s} + \sum_{s\in\mathcal{I}_{r(a)}}\lambda^{d_s^{r(a)}}\pi_{\Theta_s}\right] = \sum_{s\in\mathcal{I}_a}\lambda^{d_s^a}\pi_{\Theta_s}
\end{aligned}$$

thereby completing the induction. Finally, the result follows by applying the relation in (B.5) to the tree T whose probability mass at the root node, $\pi_{\Theta_a}=1$. □

Lemma B.2. *The function $\widetilde{H}_\alpha$ can be decomposed over the internal nodes in a tree T, as*

$$\widetilde{H}_\alpha = \frac{1}{\left(\sum_{i=1}^m \pi_{\Theta^i}^\alpha\right)^{\frac{1}{\alpha}}} \sum_{a \in \mathcal{I}} \left[\pi_{\Theta_a} \mathcal{D}_\alpha(\Theta_a) - \pi_{\Theta_{l(a)}} \mathcal{D}_\alpha(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}} \mathcal{D}_\alpha(\Theta_{r(a)})\right]$$

where $\mathcal{D}_\alpha(\Theta_a) := \left[\sum_{i=1}^m \left(\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}}\right)^\alpha\right]^{\frac{1}{\alpha}}$ and π_{Θ_a} denotes the probability mass of the objects at any internal node $a \in \mathcal{I}$.

Proof. Let T_a denote a subtree from any internal node 'a' in the tree T and let $\mathcal{I}_a$ denote the set of internal nodes in the subtree T_a. Then, define $\widetilde{H}_\alpha^a$ in a subtree T_a to be

$$\widetilde{H}_\alpha^a = 1 - \frac{\pi_{\Theta_a}}{\left[\sum_{i=1}^m \pi_{\Theta_a^i}^\alpha\right]^{\frac{1}{\alpha}}}$$

Now, we show using induction that for any subtree T_a in the tree T, the following relation holds

$$\left[\sum_{i=1}^m \pi_{\Theta_a^i}^\alpha\right]^{\frac{1}{\alpha}} \widetilde{H}_\alpha^a = \sum_{s \in \mathcal{I}_a} \left[\pi_{\Theta_s} \mathcal{D}_\alpha(\Theta_s) - \pi_{\Theta_{l(s)}} \mathcal{D}_\alpha(\Theta_{l(s)}) - \pi_{\Theta_{r(s)}} \mathcal{D}_\alpha(\Theta_{r(s)})\right] \qquad \text{(B.6)}$$

Note that the relation holds trivially for any subtree T_a rooted at an internal node $a \in \mathcal{I}$ whose both child nodes terminate as leaf nodes. Now, consider a subtree T_a rooted at any other internal node $a \in \mathcal{I}$. Assume the above relation holds true for the subtrees rooted at its left and right child nodes. Then,

$$\begin{aligned}
\left[\sum_{i=1}^{m}\pi_{\Theta_a^i}^{\alpha}\right]^{\frac{1}{\alpha}}\widetilde{H}_{\alpha}^{a} &= \left[\sum_{i=1}^{m}\pi_{\Theta_a^i}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_a} = \left[\sum_{i=1}^{m}\pi_{\Theta_a^i}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_{l(a)}} - \pi_{\Theta_{r(a)}} \\
&= \left[\sum_{i=1}^{m}\pi_{\Theta_a^i}^{\alpha}\right]^{\frac{1}{\alpha}} - \left[\sum_{i=1}^{m}\pi_{\Theta_{l(a)}^i}^{\alpha}\right]^{\frac{1}{\alpha}} - \left[\sum_{i=1}^{m}\pi_{\Theta_{r(a)}^i}^{\alpha}\right]^{\frac{1}{\alpha}} \\
&\quad + \left(\left[\sum_{i=1}^{m}\pi_{\Theta_{l(a)}^i}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_{l(a)}}\right) + \left(\left[\sum_{i=1}^{m}\pi_{\Theta_{r(a)}^i}^{\alpha}\right]^{\frac{1}{\alpha}} - \pi_{\Theta_{r(a)}}\right) \\
&= \left[\pi_{\Theta_a}\mathcal{D}_{\alpha}(\Theta_a) - \pi_{\Theta_{l(a)}}\mathcal{D}_{\alpha}(\Theta_{l(a)}) - \pi_{\Theta_{r(a)}}\mathcal{D}_{\alpha}(\Theta_{r(a)})\right] \\
&\quad + \left[\sum_{i=1}^{m}\pi_{\Theta_{l(a)}^i}^{\alpha}\right]^{\frac{1}{\alpha}}\widetilde{H}_{\alpha}^{l(a)} + \left[\sum_{i=1}^{m}\pi_{\Theta_{r(a)}^i}^{\alpha}\right]^{\frac{1}{\alpha}}\widetilde{H}_{\alpha}^{r(a)} \\
&= \sum_{s\in\mathcal{I}_a}\left[\pi_{\Theta_s}\mathcal{D}_{\alpha}(\Theta_s) - \pi_{\Theta_{l(s)}}\mathcal{D}_{\alpha}(\Theta_{l(s)}) - \pi_{\Theta_{r(s)}}\mathcal{D}_{\alpha}(\Theta_{r(s)})\right]
\end{aligned}$$

where the last step follows from the induction hypothesis. Finally, the result follows by applying the relation in (B.6) to the tree T. □

Proof of Corollary IV.5

The result in Corollary IV.5 is a special case of that in Theorem IV.4 when $\lambda \to 1$. It follows by taking the logarithm to the base λ on both sides of equation

$$\lambda^{L_\lambda(\Pi)} = \lambda^{H_\alpha(\Pi_{\mathbf{y}})} + \sum_{a\in\mathcal{I}}\pi_{\Theta_a}\left[(\lambda-1)\lambda^{d_a} - \mathcal{D}_{\alpha}(\Theta_a) + \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}}\mathcal{D}_{\alpha}(\Theta_{l(a)}) + \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}}\mathcal{D}_{\alpha}(\Theta_{r(a)})\right],$$

and then finding the limit as $\lambda \to 1$.

Using L'Hôpital's rule, the left hand side (LHS) of the equation reduces to

$$\lim_{\lambda\to1}\log_\lambda(\text{LHS}) = \lim_{\lambda\to1}L_\lambda(\Pi) = \sum_{j\in\mathcal{L}}\pi_{\Theta_j}d_j,$$

where $L_\lambda(\Pi) = \log_\lambda\left(\sum_{j\in\mathcal{L}}\pi_{\Theta_j}\lambda^{d_j}\right)$. Similarly, the right hand side (RHS) of the equation reduces to

$$\lim_{\lambda \to 1} \log_\lambda(\text{RHS}) = H(\Pi_{\mathbf{y}}) + \sum_{a \in \mathcal{I}} \pi_{\Theta_a} \left[1 - \left(H(\Theta_a) - \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} H(\Theta_{l(a)}) - \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} H(\Theta_{r(a)}) \right) \right],$$

where $H(\Theta_a) = -\sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \log_2 \left(\frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} \right)$.

Finally, the result follows by noting that

$$H(\Theta_a) - \frac{\pi_{\Theta_{l(a)}}}{\pi_{\Theta_a}} H(\Theta_{l(a)}) - \frac{\pi_{\Theta_{r(a)}}}{\pi_{\Theta_a}} H(\Theta_{r(a)}) = H(\rho_a) + \sum_{i=1}^{m} \frac{\pi_{\Theta_a^i}}{\pi_{\Theta_a}} H(\rho_a^i),$$

as shown in Theorem III.8.

APPENDIX C

Appendix for Diagnosis under Persistent Query Noise

Proof of Lemma V.1

For any given k and h, let $g(p) := \log[p^h(1-p)^{k-h}]$. It can be easily verified that $g'(p) = 0$ when $p = \frac{h}{k}$ and $g''(p)|_{p=\frac{h}{k}} < 0$ which implies that $g(p) \leq g(\frac{h}{k})$, $\forall\ p$, from which the inequality in (5.6) follows.

In addition, when $p \leq p_2$, we need to show that the bound can be improved to

$$p^h(1-p)^{k-h} \leq \begin{cases} p_2^h(1-p_2)^{k-h} & \text{if } p_2 \leq \frac{h}{k}, \\ \left(\frac{h}{k}\right)^h \left(1-\frac{h}{k}\right)^{k-h} & \text{if } p_2 > \frac{h}{k}. \end{cases}$$

Note that the second part of this result, where $p_2 > h/k$ follows from the above result. Hence, it remains to show that $\forall\ p_2 \leq \frac{h}{k}$, $p^h(1-p)^{k-h} \leq p_2^h(1-p_2)^{k-h}$, which is equivalent to showing that $\forall h \geq kp_2$, $g(p_2) - g(p) \geq 0$.

$$\begin{aligned} g(p_2) - g(p) &= h \log \frac{p_2(1-p)}{p(1-p_2)} + k \log \frac{1-p_2}{1-p} \\ &\geq kp_2 \log \frac{p_2(1-p)}{p(1-p_2)} + k \log \frac{1-p_2}{1-p} \\ &= k \left[p_2 \log \frac{p_2}{p} + (1-p_2) \log \frac{1-p_2}{1-p} \right] \geq 0 \end{aligned}$$

where the first inequality follows from $h \geq kp_2$ (the first log is ≥ 0 since $p \leq p_2$) and the last inequality follows from the non-negativity of Kullback-Leibler divergence. The other two cases can be proved in a similar manner.

Proof of Proposition V.2

Let $|\mathcal{A}| = k$. Consider the case where $\exists\ \overline{p} \in (0, \rho/(1+\rho))$ such that $0 < p \leq \overline{p}$ (The other case where $\exists\ \underline{p} \in (1/(1+\rho), 1)$ such that $1 > p \geq \underline{p}$ can be proved in a similar manner). Note from the definitions of $r_{wc}(\theta|\mathbf{z}_\mathcal{A})$ and $\overline{r}_{wc}(\theta|\mathbf{z}_\mathcal{A})$ that the result follows by showing the following relational equivalence between the true probabilities and the estimated probabilities: $\forall i, j$

$$\pi_i \Pr(\mathbf{z}_\mathcal{A}|X_i = 1) \geq \pi_j \Pr(\mathbf{z}_\mathcal{A}|X_j = 1) \Longleftrightarrow \pi_i \overline{\Pr}(\mathbf{z}_\mathcal{A}|X_i = 1) \geq \pi_j \overline{\Pr}(\mathbf{z}_\mathcal{A}|X_j = 1), \tag{C.1}$$

where the true likelihood and the estimated likelihood of any object θ_i are given by $\Pr(\mathbf{z}_\mathcal{A}|X_i = 1) = p^{h_i}(1-p)^{k-h_i}$ and $\overline{\Pr}(\mathbf{z}_\mathcal{A}|X_i = 1) = \varepsilon_i^{h_i}(1-\varepsilon_i)^{k-h_i}$, $h_i = \delta_{i,\mathcal{A}}$ and $\varepsilon_i := \min\{h_i/k, \overline{p}\}$.

The above equivalence follows trivially for any pair of objects θ_i, θ_j whose $h_i = h_j$. To show that the equivalence holds even when $h_i \neq h_j$, we will show that, for any two objects θ_i, θ_j with priors π_i, π_j,

$$\pi_i \Pr(\mathbf{z}_{\mathcal{A}}|X_i = 1) > \pi_j \Pr(\mathbf{z}_{\mathcal{A}}|X_j = 1) \ \& \ (h_i \neq h_j) \iff h_j > h_i \tag{C.2a}$$

$$\text{and } \pi_i \overline{\Pr}(\mathbf{z}_{\mathcal{A}}|X_i = 1) > \pi_j \overline{\Pr}(\mathbf{z}_{\mathcal{A}}|X_j = 1) \ \& \ (h_i \neq h_j) \iff h_j > h_i. \tag{C.2b}$$

We will first prove (C.2a), followed by (C.2b). Note that $h_j > h_i$ is equivalent to $h_j \geq h_i + 1$. Using the fact that $p < \frac{\rho}{1+\rho}$ and that for any i, j, $\frac{\pi_j}{\pi_i} \leq \frac{\max_k \pi_k}{\min_k \pi_k} = \frac{1}{\rho}$, we can show the converse of (C.2a) as follows. If $h_j - h_i \geq 1$, then

$$\begin{aligned}
&(h_j - h_i) \log \frac{1-p}{p} \geq \log \frac{1-p}{p} > \log \frac{1}{\rho} \geq \log \frac{\pi_j}{\pi_i} \\
\implies \ &\log \pi_i + h_i \log \frac{p}{1-p} > \log \pi_j + h_j \log \frac{p}{1-p} \\
\implies \ &\log \pi_i p^{h_i} (1-p)^{k-h_i} > \log \pi_j p^{h_j} (1-p)^{k-h_j}.
\end{aligned}$$

To prove the forward direction, we need to show that

$$h_j \leq h_i \implies (h_i = h_j) \text{ or } \pi_i \Pr(\mathbf{z}_{\mathcal{A}}|X_i = 1) \leq \pi_j \Pr(\mathbf{z}_{\mathcal{A}}|X_j = 1).$$

If $h_j < h_i$, then $\pi_i \Pr(\mathbf{z}_{\mathcal{A}}|X_i = 1) < \pi_j \Pr(\mathbf{z}_{\mathcal{A}}|X_j = 1)$ using the converse result with dummy variables i and j interchanged, thereby proving (C.2a). Similarly, to prove the converse of (C.2b), we need to show that $h_j > h_i$ leads to $\pi_i \overline{\Pr}(\mathbf{z}_{\mathcal{A}}|X_i = 1) > \pi_j \overline{\Pr}(\mathbf{z}_{\mathcal{A}}|X_j = 1)$, for which we need to consider three different cases.

Case 1 : Let $h_j > h_i \geq k\overline{p} \implies \varepsilon_i = \varepsilon_j = \overline{p}$. Then,

$$\begin{aligned}
&(h_j - h_i) \log \frac{1-\overline{p}}{\overline{p}} \geq \log \frac{1-\overline{p}}{\overline{p}} > \log \frac{1}{\rho} \geq \log \frac{\pi_j}{\pi_i} \\
\implies \ &\log \pi_i + h_i \log \frac{\overline{p}}{1-\overline{p}} > \log \pi_j + h_j \log \frac{\overline{p}}{1-\overline{p}} \\
\implies \ &\log \pi_i \overline{p}^{h_i} (1-\overline{p})^{k-h_i} > \log \pi_j \overline{p}^{h_j} (1-\overline{p})^{k-h_j} \\
\implies &\log \pi_i \varepsilon_i^{h_i} \left(1-\varepsilon_i\right)^{k-h_i} > \log \pi_j \varepsilon_j^{h_j} \left(1-\varepsilon_j\right)^{k-h_j}.
\end{aligned}$$

Case 2 : Let $h_j \geq k\overline{p} > h_i \implies \varepsilon_i = h_i/k$ and $\varepsilon_j = \overline{p}$. Then, following along the same lines as above, we have

$$\begin{aligned}
&\log \pi_i \overline{p}^{h_i}(1-\overline{p})^{k-h_i} > \log \pi_j \overline{p}^{h_j}(1-\overline{p})^{k-h_j} \\
\Longrightarrow &\log \pi_i \left(\frac{h_i}{k}\right)^{h_i} \left(1 - \frac{h_i}{k}\right)^{k-h_i} > \log \pi_j \overline{p}^{h_j}(1-\overline{p})^{k-h_j} \\
\Longrightarrow &\log \pi_i \varepsilon_i^{h_i} \left(1-\varepsilon_i\right)^{k-h_i} > \log \pi_j \varepsilon_j^{h_j} \left(1-\varepsilon_j\right)^{k-h_j}
\end{aligned}$$

where the second statement follows from (5.6) in Lemma V.1.

Case 3 : Let $k\overline{p} > h_j > h_i$, which implies $\varepsilon_i = h_i/k$ and $\varepsilon_j = h_j/k$. Defining $g_1(h) = \log[(h/k)^h(1-h/k)^{k-h}]$ and $g_2(h) = \log \overline{p}^h(1-\overline{p})^{k-h}$, we have,

$$\frac{dg_1}{dh} = \log \frac{h/k}{1-\frac{h}{k}} < \frac{dg_2}{dh} = \log \frac{\overline{p}}{1-\overline{p}} < 0,$$

when $h < k\overline{p}$. This implies that $g_1(h)$ has a larger slope than $g_2(h)$ when $h \in [0, k\overline{p})$, and hence

$$\begin{aligned}
&\log\left(\varepsilon_i\right)^{h_i}\left(1-\varepsilon_i\right)^{k-h_i} - \log\left(\varepsilon_j\right)^{h_j}\left(1-\varepsilon_j\right)^{k-h_j} \\
&> \log \overline{p}^{h_i}(1-\overline{p})^{k-h_i} - \log \overline{p}^{h_j}(1-\overline{p})^{k-h_j} \\
&= (h_j - h_i) \log \frac{1-\overline{p}}{\overline{p}} > \log \frac{\pi_j}{\pi_i} \\
&\Longrightarrow \log \pi_i \varepsilon_i^{h_i}\left(1-\varepsilon_i\right)^{k-h_i} > \log \pi_j \varepsilon_j^{h_j}\left(1-\varepsilon_j\right)^{k-h_j},
\end{aligned}$$

thus proving the converse of (C.2b). The forward direction can be proved using the converse result in the same way as it is done for (C.2a).

Miscellanies

GBS as a special case

In the noise-free case with uniform prior on the objects (i.e., $\pi_i = 1/M$, $\forall i$), the rank-based greedy strategy in (5.5) reduces to GBS (*Dasgupta*, 2004; *Nowak*, 2008). This can be shown by noting that in the noise-free case, the likelihood values are binary with $\Pr(\mathbf{z}_{\mathcal{A}}|X_i = 1) = 1$ for all those objects whose true responses to queries in $\mathcal{A}$ are equal to the observed responses $\mathbf{z}_{\mathcal{A}}$, and 0 otherwise.

Given the responses $\mathbf{z}_{\mathcal{A}}$ to queries in $\mathcal{A}$, let $M(\mathbf{z}_{\mathcal{A}})$ be defined as follows,

$$M(\mathbf{z}_{\mathcal{A}}) = \sum_{i=1}^{M} \mathbf{I}\{\Pr(\mathbf{z}_{\mathcal{A}}|X_i = 1) = 1\}.$$

Then, the worst case rank of all those objects with a likelihood value equal to 1 is given by $M(\mathbf{z}_{\mathcal{A}})$, and that of the remaining objects is given by M.

Under a uniform prior, the greedy query selection criterion in (5.5) then reduces to

$$\begin{aligned} j^* &= \arg\min_{j \notin \mathcal{A}} \frac{1}{M} \sum_{z=0,1} \sum_{i=1}^{M(\mathbf{z}_{\mathcal{A}} \cup z)} M(\mathbf{z}_{\mathcal{A}} \cup z) \\ &= \arg\min_{j \notin \mathcal{A}} \frac{1}{M} \Big[M^2(\mathbf{z}_{\mathcal{A}} \cup 0) + M^2(\mathbf{z}_{\mathcal{A}} \cup 1) \Big], \end{aligned}$$

where $M(\mathbf{z}_{\mathcal{A}} \cup 0) + M(\mathbf{z}_{\mathcal{A}} \cup 1) = M(\mathbf{z}_{\mathcal{A}})$, and $\mathbf{z}_{\mathcal{A}} \cup z$ corresponds to the observed responses to queries in $\mathcal{A} \cup j$. The solution to this constrained optimization problem is to choose a query that most evenly divides the $M(\mathbf{z}_{\mathcal{A}})$ objects, which is the standard splitting algorithm or GBS.

Details of networks generated for experiments

We will now briefly describe how the bipartite networks used in the experiments in Chapters V and VI were generated.

- *Random Networks*: The Erdös-Rényi random networks were generated using an edge density value (p) between 0.02 and 0.2, where p corresponds to the probability that a particular object and query are connected. The Preferential Attachment random network model consists of two parameters, α and ν, where α corresponds to the probability with which an edge is generated uniformly at random, and ν corresponds to the maximum edge degree of the objects in the bipartite diagnosis graph. For more details, refer to *Guillaume and Latapy* (2004). In the networks we generated, we used α values in the range of $[0.1, 0.3]$ and ν was chosen to be less than 10% of the maximum possible edge degree.

- *Computer Networks*: The computer networks used in this paper were generated in a two-stage process consisting of (1) network topology generation and (2) probe set selection. In the first stage, network topologies were created using the BRITE (*Medina et al.*, 2001) and the INET 3.0 (*Winick and Jamin*, 2002) generators, which simulate an internet like topology at the Autonomous Systems (AS) level. More specifically, the BRITE networks were generated using the AS Waxman model under default parameters, where the plane dimensions were scaled based on the number of components. The INET network was also generated using an AS model with default parameters.

  Given this network topology, a random set of K network components were chosen to be designated as probe stations. Probes were then generated by computing the shortest path from each probe station to every component. This set is then decreased in size using a greedy process known as Subtractive search (*Brodie et al.*, 2001), where the probes were selected passively such that the resulting probe set guarantees single fault diagnosis. Once this set has been created, additional probes were added greedily to allow for multiple fault diagnosis. In the INET network we generated, Subtractive search was slow, and hence the probes were selected based on greedy covering.

APPENDIX D

Appendix for Multi Fault Diagnosis

Proof of Proposition VI.1

We will show that the estimates for the area above the ROC curve, $\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}})$, $\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}})$ and $\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}})$ in (6.4) can be equivalently expressed as

$$\begin{aligned}
\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) &= \frac{1}{2} + \frac{\mathbf{U}(\mathbf{z}_{\mathcal{A}}) + \mathbf{V}(\mathbf{z}_{\mathcal{A}})}{2\mathbf{W}(\mathbf{z}_{\mathcal{A}})} \\
\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) &= \frac{1}{2} + \frac{\mathbf{U}(\mathbf{z}_{\mathcal{A}})}{2\mathbf{W}(\mathbf{z}_{\mathcal{A}})} \\
\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) &= \frac{1}{2} + \frac{\mathbf{U}(\mathbf{z}_{\mathcal{A}}) - \mathbf{V}(\mathbf{z}_{\mathcal{A}})}{2\mathbf{W}(\mathbf{z}_{\mathcal{A}})}
\end{aligned}$$

where

$$\mathbf{U}(\mathbf{z}_{\mathcal{A}}) = \sum_{i=1}^{M}(2i - M - 1)\,\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \tag{D.1a}$$

$$\mathbf{V}(\mathbf{z}_{\mathcal{A}}) = \sum_{i=1}^{M}\Pr(X_i = 1|\mathbf{z}_{\mathcal{A}})\,\Pr(X_i = 0|\mathbf{z}_{\mathcal{A}}) \tag{D.1b}$$

$$\mathbf{W}(\mathbf{z}_{\mathcal{A}}) = \sum_{i=1}^{M}\Pr(X_i = 1|\mathbf{z}_{\mathcal{A}})\sum_{i=1}^{M}\Pr(X_i = 0|\mathbf{z}_{\mathcal{A}}).$$

The result in Proposition VI.1 will then follow by observing that under a single fault assumption, $\mathbf{W}(\mathbf{z}_{\mathcal{A}}) = M - 1$.

To prove the above equivalences, we will first show this result for $\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}})$, and the other two results follow by observing that

$$\begin{aligned}
\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) &= \overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) + \frac{\mathbf{V}(\mathbf{z}_{\mathcal{A}})}{\mathbf{W}(\mathbf{z}_{\mathcal{A}})} \\
\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) &= \overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) + \frac{\mathbf{V}(\mathbf{z}_{\mathcal{A}})}{2\mathbf{W}(\mathbf{z}_{\mathcal{A}})}.
\end{aligned}$$

We will now show the equivalence result for $\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}})$. Let $\mathbf{N}(\mathbf{z}_{\mathcal{A}}) := \sum_{i=1}^{M-1} \sum_{j=i+1}^{M} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r(j)} = 1|\mathbf{z}_{\mathcal{A}})$ denote its numerator. Then, the result follows by observing that

$$\begin{aligned}
&\sum_{i=1}^{M} \Pr(X_i = 0|\mathbf{z}_{\mathcal{A}}) \sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) \\
&\quad = \sum_{i=1}^{M} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) \sum_{i=1}^{M} \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\quad = \mathbf{N}(\mathbf{z}_{\mathcal{A}}) + \sum_{i=1}^{M} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) \sum_{j=1}^{i} \Pr(X_{r(j)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\quad = \mathbf{N}(\mathbf{z}_{\mathcal{A}}) + \sum_{i=1}^{M} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \sum_{i=2}^{M} \sum_{j=1}^{i-1} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r(j)} = 1|\mathbf{z}_{\mathcal{A}}), \qquad \text{(D.2)}
\end{aligned}$$

where the last term in the above expression can be expressed in terms of $\mathbf{N}(\mathbf{z}_{\mathcal{A}})$ using the relation $\Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) = 1 - \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})$,

$$\begin{aligned}
&\sum_{i=2}^{M}\sum_{j=1}^{i-1}\Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\quad= \sum_{i=2}^{M}\sum_{j=1}^{i-1}\Bigg[1 - \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) - \Pr(X_{r(j)} = 0|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)} = 0|\mathbf{z}_{\mathcal{A}})\Bigg] \\
&\quad= \sum_{i=2}^{M}\sum_{j=1}^{i-1}\Bigg[- \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) + \Pr(X_{r(j)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)} = 0|\mathbf{z}_{\mathcal{A}})\Bigg] \\
&\quad= \sum_{i=2}^{M} -(i-1)\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) + \sum_{i=1}^{M-1}(M-i)\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \sum_{i=2}^{M}\sum_{j=1}^{i-1}\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(j)} = 0|\mathbf{z}_{\mathcal{A}}) \\
&\quad= \sum_{i=1}^{M}(M-2i+1)\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \sum_{j=1}^{M-1}\sum_{i=j+1}^{M}\Pr(X_{r(j)} = 0|\mathbf{z}_{\mathcal{A}})\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\quad= \sum_{i=1}^{M}(M-2i+1)\Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) + \mathbf{N}(\mathbf{z}_{\mathcal{A}}).
\end{aligned}$$

Finally, substituting the above relation in (D.2), we get

$$\begin{aligned}
&\sum_{i=1}^{M} \Pr(X_i = 0|\mathbf{z}_{\mathcal{A}}) \sum_{i=1}^{M} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad = 2\mathbf{N}(\mathbf{z}_{\mathcal{A}}) + \sum_{i=1}^{M} \Pr(X_{r(i)} = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \sum_{i=1}^{M} (M - 2i + 1) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad = 2\mathbf{N}(\mathbf{z}_{\mathcal{A}}) + \sum_{i=1}^{M} \Pr(X_i = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) \\
&\qquad\quad + \sum_{i=1}^{M} (M - 2i + 1) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})
\end{aligned}$$

from which, the result follows.

Proof of Theorem VI.2

Since $\underline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}}) = 1 - \overline{\mathbf{A}}(\mathbf{z}_{\mathcal{A}})$, the result in Theorem VI.2 follows by showing that $\forall \mathcal{A}' \subseteq \mathcal{A}$

$$\overline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}}) \leq \overline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}'}) \text{ and } \overline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}}) \leq \overline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}'})$$

Let $\mathbf{z}_{\mathcal{A}}$ denote the responses to queries in the set $\mathcal{A}$. To prove adaptive monotonicity for $\overline{\mathbf{A}}_{lr}(\mathbf{Z}_{\mathcal{A}})$, it suffices to show that for any query $j \notin \mathcal{A}$, $\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) - \mathbb{E}_{Z_j}[\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}} \cup Z_j)] \geq 0$ *Golovin and Krause* (2010). Similarly, for $\overline{\mathbf{A}}_{l}(\mathbf{Z}_{\mathcal{A}})$, we need to show that $\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) - \mathbb{E}_{Z_j}[\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}} \cup Z_j)] \geq 0$.

Under single fault assumption, we have

$$\begin{aligned}
\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) &= \frac{1}{2} + \frac{\mathbf{U}(\mathbf{z}_{\mathcal{A}}) + \mathbf{V}(\mathbf{z}_{\mathcal{A}})}{2(M-1)}, \text{ and} \\
\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) &= \frac{1}{2} + \frac{\mathbf{U}(\mathbf{z}_{\mathcal{A}})}{2(M-1)},
\end{aligned}$$

where $\mathbf{U}(\mathbf{z}_{\mathcal{A}})$ and $\mathbf{V}(\mathbf{z}_{\mathcal{A}})$ are as defined in (D.1a) and (D.1b), respectively. Hence, the adaptive monotonicity of $\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}})$ and $\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}})$ follows by showing that $\forall j \notin \mathcal{A}$

$$\begin{aligned}
&\mathbf{U}(\mathbf{z}_{\mathcal{A}}) - \mathbb{E}_{Z_j}[\mathbf{U}(\mathbf{z}_{\mathcal{A}} \cup Z_j)] \geq 0, \text{ and} \\
&\mathbf{V}(\mathbf{z}_{\mathcal{A}}) - \mathbb{E}_{Z_j}[\mathbf{V}(\mathbf{z}_{\mathcal{A}} \cup Z_j)] \geq 0,
\end{aligned}$$

which follow from Lemma D.1 and D.2, below.

Lemma D.1. *Let $\mathbf{z}_{\mathcal{A}}$ denote the observed responses to queries in the set $\mathcal{A}$. Then, for any query $j \notin \mathcal{A}$,*

$$\mathbf{U}(\mathbf{z}_{\mathcal{A}}) - \mathbb{E}_{Z_j}[\mathbf{U}(\mathbf{z}_{\mathcal{A}} \cup Z_j)] \geq 0$$

Proof. Under single fault assumption, $\mathbf{U}(\mathbf{z}_{\mathcal{A}}) = -(M+1) + \sum_{i=1}^{M} 2i \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})$. Hence, the result follows by showing that $\forall\ j \notin \mathcal{A}$,

$$\begin{aligned}
\sum_{i=1}^{M} i \Bigg\{ \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) - \Bigg[&\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r_0(i)} = 1|\mathbf{z}_{\mathcal{A}}, 0) \\
&+ \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r_1(i)} = 1|\mathbf{z}_{\mathcal{A}}, 1) \Bigg] \Bigg\} \geq 0. \qquad \text{(D.3)}
\end{aligned}$$

As mentioned earlier, the rank order depends on the queries chosen $\mathcal{A}$ and their observed responses $\mathbf{z}_{\mathcal{A}}$. Hence, to differentiate the rank orders in the above expression, we use $r(i)$ to denote the rank order of the objects based on the observed responses $\mathbf{z}_{\mathcal{A}}$, and $r_0(i)$, $r_1(i)$ to denote the rank order of the objects based on the observed responses $\mathbf{z}_{\mathcal{A}} \cup 0$ and $\mathbf{z}_{\mathcal{A}} \cup 1$ to queries in $\mathcal{A} \cup \{j\}$.

Note that (D.3) is equivalent to showing

$$\sum_{i=1}^{M}(M-i+1)\Bigg\{\Bigg[\Pr(Z_j=0|\mathbf{z}_{\mathcal{A}})\Pr(X_{r_0(i)}=1|\mathbf{z}_{\mathcal{A}},0)$$
$$+\Pr(Z_j=1|\mathbf{z}_{\mathcal{A}})\Pr(X_{r_1(i)}=1|\mathbf{z}_{\mathcal{A}},1)\Bigg]-\Pr(X_{r(i)}=1|\mathbf{z}_{\mathcal{A}})\Bigg\}\geq 0. \quad \text{(D.4)}$$

Let $\mathbf{f_t}(\mathbf{r},\mathbf{z}_{\mathcal{A}}):=\sum_{i=1}^{t}\Pr(X_{r(i)}=1|\mathbf{z}_{\mathcal{A}})$, i.e., the probability mass of the top t objects in the ranked list given by $\mathbf{r}$. Then,

$$\sum_{i=1}^{M}(M-i+1)\Pr(X_{r(i)}|\mathbf{z}_{\mathcal{A}})=\sum_{t=1}^{M}\mathbf{f_t}(\mathbf{r},\mathbf{z}_{\mathcal{A}}),$$

and hence (D.4) is equivalent to showing

$$\sum_{t=1}^{M}\Bigg[\Pr(Z_j=0|\mathbf{z}_{\mathcal{A}})\mathbf{f_t}(\mathbf{r}_0,\mathbf{z}_{\mathcal{A}}\cup 0)+\Pr(Z_j=1|\mathbf{z}_{\mathcal{A}})\mathbf{f_t}(\mathbf{r}_1,\mathbf{z}_{\mathcal{A}}\cup 1)\Bigg]-\mathbf{f_t}(\mathbf{r},\mathbf{z}_{\mathcal{A}})\geq 0.$$

Now, note that

$$\mathbf{f_t}(\mathbf{r}_0,\mathbf{z}_{\mathcal{A}}\cup 0)\geq\mathbf{f_t}(\mathbf{r},\mathbf{z}_{\mathcal{A}}\cup 0)=\sum_{i=1}^{t}\Pr(X_{r(i)}=1|\mathbf{z}_{\mathcal{A}},0).$$

Since the rank order $\mathbf{r}_0$ corresponds to the decreasing order of the posterior probabilities in $\{\Pr(X_i=1|\mathbf{z}_{\mathcal{A}},0)\}_{i=1}^{M}$, the probability mass of the top t objects in this ranked list is greater than any other t objects. Similarly, $\mathbf{f_t}(\mathbf{r}_1, \mathbf{z}_{\mathcal{A}} \cup 1) \geq \mathbf{f_t}(\mathbf{r}, \mathbf{z}_{\mathcal{A}} \cup 1)$. Hence,

$$\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})\mathbf{f_t}(\mathbf{r}_0, \mathbf{z}_{\mathcal{A}} \cup 0) + \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}})\mathbf{f_t}(\mathbf{r}_1, \mathbf{z}_{\mathcal{A}} \cup 1) \tag{D.5a}$$

$$\geq \Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})\mathbf{f_t}(\mathbf{r}, \mathbf{z}_{\mathcal{A}} \cup 0) + \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}})\mathbf{f_t}(\mathbf{r}, \mathbf{z}_{\mathcal{A}} \cup 1) \tag{D.5b}$$

$$= \sum_{i=1}^{t} \Bigg[\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}, 0) + \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}}) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}, 1) \Bigg] \tag{D.5c}$$

$$= \sum_{i=1}^{t} \Bigg[\Pr(Z_j = 0|X_{r(i)} = 1) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) + \Pr(Z_j = 1|X_{r(i)} = 1) \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \Bigg] \tag{D.5d}$$

$$= \sum_{i=1}^{t} \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) = \mathbf{f_t}(\mathbf{r}, \mathbf{z}_{\mathcal{A}}). \tag{D.5e}$$

Thus proving the inequality.

Note that in the above equation, (D.5d) follows from (D.5c) by observing that under a single fault assumption, $X_i = 1 \iff \mathbf{X} = \mathbf{I}_i$, and hence, using the conditional independence assumption of Section 2, the posterior probability can be expressed as

$$\begin{aligned} \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}, z) &= \Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}, z) \\ &= \frac{\Pr(\mathbf{X} = \mathbf{I}_i) \Pr(\mathbf{z}_{\mathcal{A}}|\mathbf{X} = \mathbf{I}_i) \Pr(Z_j = z|\mathbf{X} = \mathbf{I}_i)}{\Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) \Pr(\mathbf{Z}_{\mathcal{A}} = \mathbf{z}_{\mathcal{A}})} \\ &= \frac{\Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}) \Pr(Z_j = z|\mathbf{X} = \mathbf{I}_i)}{\Pr(Z_j = z|\mathbf{z}_{\mathcal{A}})} \\ &= \frac{\Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) \Pr(Z_j = z|X_i = 1)}{\Pr(Z_j = z|\mathbf{z}_{\mathcal{A}})}. \end{aligned} \tag{D.6}$$

□

Lemma D.2. *Let* $\mathbf{z}_{\mathcal{A}}$ *denote the observed responses to queries in the set* $\mathcal{A}$*. Then,* *for any query $j \notin \mathcal{A}$,*

$$\mathbf{V}(\mathbf{z}_{\mathcal{A}}) - \mathbb{E}_{Z_j}[\mathbf{V}(\mathbf{z}_{\mathcal{A}} \cup Z_j)] \geq 0$$

Proof. Note that under single fault assumption, $\mathbf{V}(\mathbf{z}_{\mathcal{A}}) = 1 - \sum_{i=1}^{M} \Pr^2(X_i = 1|\mathbf{z}_{\mathcal{A}})$. Hence, we need to show that $\forall\ j \notin \mathcal{A}$,

$$\sum_{i=1}^{M} \Bigg\{ \Bigg[\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}}) \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}}, 0) + \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}}) \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}}, 1) \Bigg] \\ - \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}}) \Bigg\} \geq 0. \tag{D.7}$$

Substituting the expression for posterior probability from (D.6) in the LHS of (D.7), we get

$$\begin{aligned}
&\sum_{i=1}^{M} \left\{ \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}}) \left[\frac{\Pr^2(Z_j = 0|X_i = 1)}{\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})} + \frac{\Pr^2(Z_j = 1|X_i = 1)}{\Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}})} - 1 \right] \right\} \\
&= \sum_{i=1}^{M} \left\{ \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}}) \left[\frac{\Big(1 - \Pr(Z_j = 1|X_i = 1)\Big)^2}{\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})} + \frac{\Pr^2(Z_j = 1|X_i = 1)}{\Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}})} - 1 \right] \right\}, \\
&= \sum_{i=1}^{M} \left\{ \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}}) \left[\frac{\Bigg(\Pr(Z_j = 1|X_i = 1) - \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}}) \Bigg)^2}{\Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}}) \Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})} \right] \right\} \\
&\geq 0
\end{aligned}$$

where the last equality follows by using the relation $\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}}) = 1 - \Pr(Z_j = 1|\mathbf{z}_{\mathcal{A}})$, and completing the square. □

Proof of Proposition VI.3

The entropy-based query selection criterion is given by

$$j^* = \arg\min_{j \notin \mathcal{A}} \sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}, z). \tag{D.8}$$

Since, under a single fault assumption, $X_i = 1 \iff \mathbf{X} = \mathbf{I}_i$, we need to show that the above query selection criterion reduces to

$$j^* := \arg\min_{j \notin \mathcal{A}} \sum_{i=1}^{M} \Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}) H\Big(\Pr(Z_j = 0|\mathbf{X} = \mathbf{I}_i)\Big) - H\Big(\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})\Big).$$

We show this by first noting that under a single fault assumption, the conditional entropy reduces to

$$H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}, z) = -\sum_{i=1}^{M} \Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}, z) \log \Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}, z).$$

In addition, as noted in (D.6), under the conditional independence assumption of Section 2, the posterior probability can be expressed as

$$\Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}, z) = \frac{\Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}) \Pr(Z_j = z|\mathbf{X} = \mathbf{I}_i)}{\Pr(Z_j = z|\mathbf{z}_{\mathcal{A}})}. \tag{D.9}$$

Substituting the above expression in (D.8), we get

$$\begin{aligned} &\sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) H(\mathbf{X}|\mathbf{Z}_{\mathcal{A}}, z) \\ &\qquad = -\sum_{z=0,1} \sum_{i=1}^{M} \Bigg[\Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) \Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}, z) \\ &\qquad\qquad\qquad \log \frac{\Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}) \Pr(Z_j = z|\mathbf{X} = \mathbf{I}_i)}{\Pr(Z_j = z|\mathbf{z}_{\mathcal{A}})} \Bigg]. \end{aligned} \tag{D.10}$$

This expression can be broken down into 3 different terms. The first term is given by

$$
\begin{aligned}
-\sum_{z=0,1}\sum_{i=1}^{M}&\Bigg[\Pr(Z_j=z|\mathbf{z}_{\mathcal{A}})\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}},z)\;\log\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}})\Bigg]\\
&=-\sum_{i=1}^{M}\Bigg[\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}})\log\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}})\;\sum_{z=0,1}\Pr(Z_j=z|\mathbf{X}=\mathbf{I}_i)\Bigg]\\
&=H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}),
\end{aligned}
$$

where the second equality follows from (D.9) and the last equality follows since $\sum_z \Pr(Z_j=z|\mathbf{X}=\mathbf{I}_i)=1$.

The second term is given by

$$
\begin{aligned}
-\sum_{z=0,1}\sum_{i=1}^{M}&\Bigg[\Pr(Z_j=z|\mathbf{z}_{\mathcal{A}})\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}},z)\;\log\frac{1}{\Pr(Z_j=z|\mathbf{z}_{\mathcal{A}})}\Bigg]\\
&=-\sum_{z=0,1}\Bigg[\Pr(Z_j=z|\mathbf{z}_{\mathcal{A}})\log\frac{1}{\Pr(Z_j=z|\mathbf{z}_{\mathcal{A}})}\;\sum_{i=1}^{M}\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}},z)\Bigg]\\
&=-H\Big(\Pr(Z_j=0|\mathbf{z}_{\mathcal{A}})\Big),
\end{aligned}
$$

where the last equality follows since $\sum_{i=1}^{M}\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}},z)=1$.

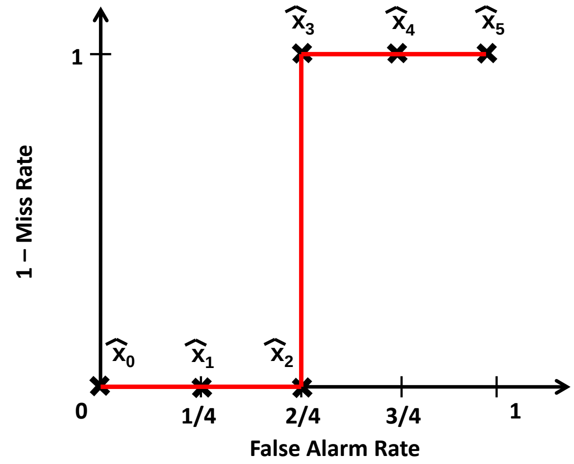
The last term is given by

$$
\begin{aligned}
-\sum_{z=0,1}\sum_{i=1}^{M}&\Bigg[\Pr(Z_j=z|\mathbf{z}_{\mathcal{A}})\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}},z)\;\log\Pr(Z_j=z|\mathbf{X}=\mathbf{I}_i)\Bigg]\\
&=-\sum_{i=1}^{M}\Bigg[\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}})\Bigg(\sum_{z=0,1}\Pr(Z_j=z|\mathbf{X}=\mathbf{I}_i)\;\log\Pr(Z_j=z|\mathbf{X}=\mathbf{I}_i)\Bigg)\Bigg]\\
&=\sum_{i=1}^{M}\Pr(\mathbf{X}=\mathbf{I}_i|\mathbf{z}_{\mathcal{A}})H\Big(\Pr(Z_j=0|\mathbf{X}=\mathbf{I}_i)\Big).
\end{aligned}
$$

	X_1	X_2	X_3	X_4	X_5
true state x_i	0	0	1	0	0
$\Pr(X_i = 1\vert\mathbf{z}_{\mathcal{A}})$	0.3	0.35	0.2	0.05	0.1

Ranked list	X_2	X_1	X_3	X_5	X_4

(a)



(b)

Figure D.1: Demonstrates the ROC curve corresponding to a ranked list of objects, when there is only one object in state 1

Substituting these 3 terms back into (D.10), we get

$$\sum_{z=0,1} \Pr(Z_j = z|\mathbf{z}_{\mathcal{A}}) H(\mathbf{X}|\mathbf{Z}_{\mathcal{A}}, z) = H(\mathbf{X}|\mathbf{z}_{\mathcal{A}}) - H\Big(\Pr(Z_j = 0|\mathbf{z}_{\mathcal{A}})\Big) + \sum_{i=1}^{M} \Pr(\mathbf{X} = \mathbf{I}_i|\mathbf{z}_{\mathcal{A}}) H\Big(\Pr(Z_j = 0|\mathbf{X} = \mathbf{I}_i)\Big),$$

and the result follows since $H(\mathbf{X}|\mathbf{z}_{\mathcal{A}})$ does not depend on the query j.

Miscellanies

Expected rank criterion as a special case

We will now show that the rank-based active query selection criterion proposed in Chapter V is a special case of the AUC-based criterion proposed in Chapter VI.

We begin by noting that in the special case when there is only one fault, the ROC curve corresponding to the rank-based estimators reduces to a step function. In particular, note that the miss rate of an estimator can only take two values in this case, either 0 or 1, as there is only one object whose true state is equal to 1. Hence, the ROC curve corresponding to a ranked list of objects is a step function, where the step corresponds to the location of the faulty object (object with state 1) in the ranked list, as demonstrated by the toy example in Figure D.1. Thus, in this scenario, maximizing the area under the ROC curve (or, minimizing the area above the ROC curve) corresponding to a ranked list of objects is equivalent to minimizing the rank of the faulty object.

In fact, note from (6.7b) that in a single fault scenario, the estimate of the area above the ROC curve using a linear approximation corresponds to the expected rank of the faulty object in the ranked list. Hence, as we show below, the expected worst case rank criterion proposed in Chapter V is an upper bound on the AUC criterion.

$$\begin{aligned}\overline{\mathbf{A}}_l(\mathbf{z}_{\mathcal{A}}) &= \frac{1}{M-1}\sum_{i=1}^{M} i \cdot \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) \\ &= \frac{1}{M-1}\sum_{i=1}^{M} r(i) \cdot \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}) \\ &\leq \frac{1}{M-1}\sum_{i=1}^{M} r_{wc}(i|\mathbf{z}_{\mathcal{A}}) \cdot \Pr(X_i = 1|\mathbf{z}_{\mathcal{A}}).\end{aligned}$$

Choice of upper rectangles

As mentioned in the paper, query selection based on AUC approximated using the upper rectangles performs better than the other two. We will now provide an intuitive explanation for this phenomenon.

Using the result in Proposition VI.1, and noting that $\Pr(X_i = 0|\mathbf{z}_{\mathcal{A}}) = 1 - \Pr(X_i =$

$1|\mathbf{z}_{\mathcal{A}})$, we can re-write the expressions for the area above the ROC curve in (6.7) as

$$\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}}) = \frac{\sum\limits_{i=1}^{M} 2i \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) - \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}})}{2(M-1)} + c_l,$$

$$\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}}) = \frac{\sum\limits_{i=1}^{M} 2i \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}})}{2(M-1)} + c_m,$$

$$\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}}) = \frac{\sum\limits_{i=1}^{M} 2i \Pr(X_{r(i)} = 1|\mathbf{z}_{\mathcal{A}}) + \overset{2}{\Pr}(X_i = 1|\mathbf{z}_{\mathcal{A}})}{2(M-1)} + c_u,$$

where c_l, c_m and c_u are constants that do not contribute to query selection.

Now note that all three approximations have the same first term, which corresponds to the expected rank of the faults in the ranked list. However, they differ with respect to the second term, which makes the crucial difference in terms of the query selected. More specifically, given two or more queries with the the same expected rank value (i.e., same value for the first term), query selected using $\overline{\mathbf{A}}_{ur}(\mathbf{z}_{\mathcal{A}})$ chooses the one that most evenly distributes the posterior probability mass of 1 among all the objects, while query selected using $\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}})$ chooses the one that assigns most of the probability mass to one object, and the query selected using $\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}})$ just picks one at random. Hence, the queries selected using $\overline{\mathbf{A}}_{lr}(\mathbf{z}_{\mathcal{A}})$ and $\overline{\mathbf{A}}_{l}(\mathbf{z}_{\mathcal{A}})$ are more prone to increasing the posterior fault probability of one (or few) object(s), thereby creating a bias towards those objects in the queries selected there after. However, this is overcome by the AUC-based query selection criterion approximated using the upper rectangles.

BIBLIOGRAPHY

BIBLIOGRAPHY

Adler, M., and B. Heeringa (2008), Approximating Optimal binary decision trees, *Proceedings of the 11th International Workshop on Approximation, Randomization and Combinatorial Optimization*, pp. 1–9.

Angluin, D. (2004), Queries revisited, *Theoretical Computer Science*, *313*, 175–194.

Arikan, E. (1996), An inequality on guessing and its application to sequential decoding, *IEEE Transactions on Information Theory*, *IT-42*, 99–105.

Arikan, E., and N. Merhav (1998), Guessing subject to distortion, *IEEE Transactions on Information theory*, *IT-44*, 1041–1056.

Ataman, K., W. N. Street, and Y. Zhang (2006), Learning to rank by maximizing AUC with linear programming, *Proc. IJCNN*, pp. 123–129.

Baer, M. B. (2006), Rényi to Rényi - Source coding under seige, *Proceedings of IEEE International Symposium on Information Theory*, pp. 1258–1262.

Balcan, M. F., A. Beygelzimer, and J. Langford (2006), Agnostic active learning, *Proceedings of the 23rd International Conference on Machine Learning.*

Bellala, G., S. K. Bhavnani, and C. Scott (2010), Extensions of generalized binary search to group identification and exponential costs, *Advances in Neural Information Processing Systems 23 (NIPS)*, pp. 154–162.

Bellala, G., S. K. Bhavnani, and C. Scott (2011a), Active diagnosis under persistent noise with unknown noise distribution: A rank-based approach, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS).*

Bellala, G., S. K. Bhavnani, and C. Scott (2011b), Group-based active query selection for rapid diagnosis in time-critical situations, *to appear in IEEE Transactions on Information Theory.*

Bellala, G., J. Stanley, C. Scott, and S. K. Bhavnani (2011c), Active diagnosis via AUC maximization: An efficient approach for multiple fault identification in large scale, noisy networks, *Proceedings of the 27th International Conference on Uncertainty in Artificial Intelligence (UAI).*

Bhavnani, S. K., A. Abraham, C. Demeniuk, M. Gebrekristos, A. Gong, S. Nainwal, G. Vallabha, and R. Richardson (2007), Network analysis of toxic chemicals and symptoms: Implications for designing first-responder systems, *Proceedings of American Medical Informatics Association.*

Blum, A., J. Jackson, T. Sandholm, and M. Zinkevich (2004), Preference elicitation and query learning, *Journal of Machine Learning Research*, *5*, 649–667.

Brodie, M., I. Rish, and S. Ma (2001), Optimizing probe selection for fault localization, *Distributed Systems Operation and Management.*

Campbell, L. L. (1965), A coding problem and Rényi's entropy, *Information and Control*, *8*(4), 423–429.

Campbell, L. L. (1966), Definition of entropy by means of a coding problem, *Z.Wahrscheinlichkeitstheorie und verwandte Gebiete*, *6*, 113–118.

Chakaravarthy, V. T., V. Pandit, S. Roy, P. Awasthi, and M. Mohania (2007), Decision Trees for Entity Identification: Approximation algorithms and hardness results, *Proceedings of the ACM SIGMOD Symposium on Principles of Database Systems.*

Chakaravarthy, V. T., V. Pandit, S. Roy, and Y. Sabharwal (2009), Approximating Decision trees with multiway branches, *ICALP*, pp. 210–221.

Chen, L., and P. Pu (2004), Survey of preference elicitation methods, *Tech. Rep. IC/200467*, Swiss Federal Institute of Technology in Lausanne(EPFL).

Cheng, L., X. Qui, L. Meng, Y. Qiao, and R. Boutaba (2010), Efficient active probing for fault diagnosis in large scale and noisy networks, *IEEE INFOCOM.*

Cicalese, F., T. Jacobs, E. Laber, and M. Molinaro (2010), On greedy algorithms for decision trees, *In Proceedings of ISAAC.*

Cortes, C., and M. Mohri (2003), AUC optimization vs. error rate minimization, *Advances in Neural Information Processing Systems (NIPS) 15.*

Cover, T. M., and J. A. Thomas (1991), *Elements of Information Theory*, John Wiley.

Culver, M., K. Deng, and S. Scott (2006), Active learning to maximize area under the ROC curve, *Proceedings of the 6th International Conference on Data Mining.*

Dasgupta, S. (2004), Analysis of a greedy active learning strategy, *Advances in Neural Information Processing Systems.*

Dasgupta, S. (2006), Coarse sample complexity bounds for active learning, *Advances in Neural Information Processing Systems.*

Du, D.-Z., and F. K. Hwang (2000), *Combinatorial Group Testing and its Applications*, World Scientific.

Fano, R. M. (1961), *Transmission of Information*, MIT Press.

Ferri, C., P. Flach, and J. Hernández-Orallo (2002), Learning decision trees using the area under the ROC curve, *In Proceedings of International Conference on Machine Learning.*

Garey, M. (1970), Optimal binary decision trees for diagnostic identification problems, Ph.D. thesis, University of Wisconsin, Madison.

Garey, M. (1972), Optimal binary identification procedures, *SIAM Journal on Applied Mathematics*, *23(2)*, 173–186.

Geman, D., and B. Jedynak (1996), An active testing model for tracking roads in satellite images, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(1), 1–14.

Goldman, S. A., M. J. Kearns, and R. E. Schapire (1990), Exact identification of circuits using fixed points of amplification functions, *Proceedings of the Thirty-First Annual Symposium on Foundations of Computer Science.*

Golovin, D., and A. Krause (2010), Adaptive Submodularity: A new approach to active learning and stochastic optimization, *In Proceedings of International Conference on Learning Theory (COLT).*

Golovin, D., D. Ray, and A. Krause (2010), Near-optimal Bayesian active learning with noisy observations, *In Advances in Neural Information Processing Systems (NIPS) 23.*

Guillaume, J., and M. Latapy (2004), *Bipartite Graphs as Models of Complex Networks*, 127-139 pp., Springer.

Gupta, A., R. Krishnaswamy, V. Nagarajan, and R. Ravi (2010), Approximation algorithms for optimal decision trees and adaptive TSP problems, *In Proceedings of ICALP, LNCS.*

Gupta, R. (2001), Quantization strategies for low-power communications, Ph.D. thesis, University of Michigan, Ann Arbor.

Hanawal, M. K., and R. Sundaresan (2008), Guessing revisited: A large deviations approach, *Tech. Rep. TR-PME-2008-08*, DRDO-IISc Program in Mathematical Engineering.

Hanneke, S. (2007), Teaching dimension and the complexity of active learning, *Proceedings of the 20th Conference on Learning Theory.*

Heckerman, D. (1990), A tractable inference algorithm for diagnosing multiple diseases, *In Proceedings of International Conference on Uncertainty in Artificial Intelligence (UAI)*, *5*, 163 – 171.

Hu, T. C., D. J. Kleitman, and J. T. Tamaki (1979), Binary trees optimal under various criteria, *SIAM Journal on Applied Mathematics*, *37*(2), 246–256.

Huffman, D. A. (1952), A method for the construction of minimum-redundancy codes, *Proceedings of the Institute of Radio Engineers.*

Humblet, P. A. (1981), Generalization of Huffman coding to minimize the probability of buffer overflow, *IEEE Transactions on Information Theory*, *IT-27*(2), 230–232.

Hyafil, L., and R. Rivest (1976), Constructing optimal binary decision trees is NP-complete, *Information Processing Letters*, *5(1)*, 15–17.

Jaakkola, T. S., and M. I. Jordan (1999), Variational methods and the QMR-DT databases, *Journal of Artificial Intelligence Research*, *10*, 291–322.

Jackson, J., E. Shamir, and C. Shwartzman (1997), Learning with queries corrupted by classification noise, *Proceedings of the Fifth Israel Symposium on the Theory of Computing Systems*, pp. 45–53.

Johnson, R., J. Huber, and L. Bacon (2003), Adaptive Choice based Conjoint Analysis, *Sawtooth Software Conference Proceedings.*

Johnson, R. M. (1987), Adaptive Conjoint Analysis, *Sawtooth Software Conference Proceedings*, pp. 253–265.

Kääriäinen, M. (2006), Active learning in the non-realizable case, *Algorithmic Learning Theory*, pp. 63–77.

Kandula, S., D. Katabi, and J. P. Vasseur (2005), Shrink: A tool for failure diagnosis in IP networks, *Proc. ACM SIGCOMM MineNet Workshop.*

Kearns, M., and Y. Mansour (1995), On the boosting ability of top-down decision tree learning algorithms, *Proceedings of the Twenty-Eight Annual ACM Symposium on the Theory of Computing.*

Kleindorfer, P. R., J. C. Belke, M. R. Elliott, K. Lee, R. A. Lowe, and H. I. Feldman (2003), Accident epidemiology and the u.s. chemical industry: accident history and worst-case data from rmp*info, *Risk Analysis*, *23*(5), 865–881.

Korostelev, A. P., and J. C. Kim (2000), Rates of convergence of the sup-norm risk in image models under sequential designs, *Statistics and Probability letters*, *46*, 391–399.

Kosaraju, S. R., T. M. Przytycka, and R. S. Borgstrom (1999), On an optimal split tree problem, *Proceedings of 6th International Workshop on Algorithms and Data Structures, WADS*, pp. 11–14.

Le, T., and C. N. Hadjicostis (2007), Max-product algorithms for the generalized multiple-fault diagnosis problem, *IEEE Trans. on Sys., Man and Cybern.*, *37*(6).

Long, P. M., and R. A. Servedio (2007), Boosting the Area under the ROC curve, *Advances in Neural Information Processing Systems (NIPS) 19.*

Loveland, D. W. (1985), Performance bounds for binary testing with arbitrary weights, *Acta Informatica.*

Massey, J. L. (1994), Guessing and entropy, *IEEE International Symposium on Information Theory*, p. 204.

Medina, A., A. Lakhina, I. Matta, and J. Byers (2001), BRITE: An Approach to Universal Topology Generation, *Proc. MASCOT*.

Merhav, N., and E. Arikan (1999), The shannon cipher system with a guessing wiretrapper, *IEEE Transactions on Information Theory*, *45*, 1860–1866.

Mooij, J. M. (2010), libDAI: A free and open source C++ library for discrete approximate inference in graphical models, *Journal of Machine Learning Research.*

Murphy, K. P., Y. Weiss, and M. Jordan (1999), Loopy belief propagation for approximate inference: An empirical study, *In Proceedings of International Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 467–475.

Nowak, R. (2008), Generalized binary search, *Proceedings of the 46th Allerton Conference on Communications, Control and Computing*, pp. 568–574.

Nowak, R. (2009), Noisy generalized binary search, *Advances in Neural Information Processing Systems (NIPS) 21.*

Parker, D. S. (1980), Conditions for the optimality of the huffman algorithm, *SIAM Journal on Computing*, *9*(3), 470–489.

Pearl, J. (1988), *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann, San Mateo, CA.

Pelc, A. (2002), Searching games with errors – fifty years of coping with liars, *Theoretical Computer Science*, *270*, 71–109.

Quinlan, J. R. (1993), *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers.

Rényi, A. (1961), On a problem of information theory, *MTA Mat. Kut. Int. Kozl.*, *6B*, 505 – 516.

Rish, I., M. Brodie, S. Ma, N. Odintsova, A. Beygelzimer, G. Grabarnik, and K. Hernandez (2005), Adaptive diagnosis in distributed systems, *IEEE Trans. on Neural Networks*, *16*(5), 1088 – 1109.

Roy, S., H. Wang, G. Das, U. Nambiar, and M. Mohania (2008), Minimum-effort driven dynamic faceted search in structured databases, *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pp. 13–22.

Santoso, N. I., C. Darken, G. Povh, and J. Erdmann (1999), Nuclear plant fault diagnosis using probabilistic reasoning, *Proc. IEEE/PES*, *2*, 714–719.

Schulz, F. (2008), Trees with exponentially growing costs, *Information and Computation*, *206*.

Shannon, C. E. (1948), A mathematical theory of communication, *Bell Systems Technical Journal*, *27*, 379 – 423.

Sundaresan, R. (2007), Guessing under source uncertainty, *IEEE Transactions on Information Theory*, *53*(1), 269–287.

Swain, M. J., and M. A. Stricker (1993), Promising directions in active vision, *International Journal of Computer Vision*, *11*(2), 109–126.

Szczur, M., and B. Mashayekhi (2005), WISER wireless information system for emergency responders, *Proceedings of American Medical Informatics Association Annual Symposium.*

Takimoto, E., and A. Maruoka (2003), Top-down decision tree learning as information based boosting, *Theoretical Computer Science.*

Winick, J., and S. Jamin (2002), INET-3.0: Internet topology generator, *Tech. Rep. CSE-TR-456-02*, University of Michigan.

Yongli, Z., H. Limin, and L. Jinling (2006), Bayesian networks based approach for power systems fault diagnosis, *IEEE Trans. on P. Del.*, *21*(2), 634–639.

Zheng, A. X., I. Rish, and A. Beygelzimer (2005), Efficient test selection in active diagnosis via entropy approximation, *In Proceedings of International Conference on Uncertainty in Artificial Intelligence (UAI).*