University of California
Los Angeles

Sensitivity Analysis of
Probabilistic Graphical Models

A dissertation submitted in partial satisfaction
of the requirements for the degree
Doctor of Philosophy in Computer Science

by

Hei Chan

2005

The dissertation of Hei Chan is approved.

Phil Kellman

Stefano Soatto

Richard Korf

Adnan Darwiche, Committee Chair

University of California, Los Angeles

2005

To my parents

For their love and continual support

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acknowledgments

As I complete the writing of this dissertation, I would like to take this opportunity to express my gratitude towards the many people who I am indebted to.

First and foremost, I would like to thank my technical advisor, Adnan Darwiche. I started working for him after taking his class on probabilistic reasoning. From day one, he has offered many fresh insights on how I should conduct my research work, and provided countless feedback on my results. He has also provided guidance in the development of the SamIam project, which I implemented my results on, and is now one of the best Bayesian network graphical tools. He has not only helped me in my research, but also in writing comprehensible and well-motivated technical papers, and giving clear presentations that are appropriate to the audience. He is an excellent teacher, researcher, and mentor, and I am grateful to his advice, encouragement and patience under his tutelage.

I would also like to thank the members of my doctoral committee: Richard Korf, Stefano Soatto, and Phil Kellman. Their comments are valuable in the preparation of this dissertation. Rich has pointed out many mistakes and confusing sentences in the earlier versions of this dissertation, Stafano has given suggestions in terms of mathematical notations, and Phil has used his knowledge of human perception to help improve the graphical interface of SamIam. In particular, I would also like to thank Rich for being an instructor who care much about students' learning, both when I took the class on artificial intelligence during my undergraduate years and when I was a teaching assistant of the same class a short while ago. My interest in the material of the class I took from him was a big factor in me choosing the field of artificial intelligence in graduate school.

I must also acknowledge Judea Pearl, who has offered great advice on my work throughout the years. My research is firmly based on the many foundations of probabilistic reasoning he has built in his brilliant career, and none of my work would be possible without them. Moreover, he is a great teacher in the classroom (I very much enjoyed his class on causality), and a great humanitarian outside.

My research on belief revision has been significantly enhanced by comments and discussions with Carl Wagner, Judea Pearl, and the late Richard Jeffrey. I was fortunate enough to communicate with Richard only a few months before his death. His kind words on my initial paper on the topic were an encouragement to me, and he also helped me by pointing out other relevant work in the field. Carl's investigation on the commutativity of iterated revisions, and Judea's discussion of the interpretation of evidential statements, also inspire me to pursue the work further and complete the big picture of the process of belief revision.

I have also benefited from suggestions from Rina Dechter of the University of California at Irvine, Russ Greiner of the University of Alberta, and Wotjek Przytula of HRL. My work on tuning Bayesian network parameters came from a problem originating from Wotjek, and his many inputs provided the impetus for us to develop SamIam, and pointers for us to improve it afterwards.

My colleagues at the UCLA Automated Reasoning Group, including David Allen, Keith Cascio, Mark Chavira, Arthur Choi, Jinbo Huang, and James Park, have all asked interesting questions and offered helpful comments on my research, and I have enjoyed their company in our offices and on our trips to meetings and conferences. Most of us have participated in the SamIam program, and I commend all the developers and testers for a job well done. I would especially like to thank James for writing the inference engine, David for developing the graphical user interface with me in its early stage, and Keith for being in charge of all programming issues of the project. Without Keith's diligent maintenance of the code repository and his ability to produce new codes and debug as quickly as possible, most of my research results would not have been implemented.

Members of the UCLA Cognitive Systems Laboratory, including Chen Avin, Blai Bonet, Carlos Brito, Mark Hopkins, and Jin Tian, have also given useful remarks on my research when attending the seminars I have given throughout the years. The members of our group have spent a lot of time with them during seminars, classes, and conferences, and we have all got along very well. I would also like to thank Brenda Ng of Harvard University. She has been great fun to hang out with at the conferences, and we have shared our thoughts and problems (research related and others) by e-mails throughout the years. Brenda, I am looking forward to you graduating soon!

My years as a graduate student would not have been possible without the financial support I have received from the UCLA Computer Science Department during my stints as a graduate student researcher and a teaching assistant, and the grants from DiMI, MURI, NSF, and the Air Force. I offer my gratefulness to them for easing my financial burden.

I will soon receive my third degree from the UCLA Computer Science Department. I have learned a lot from its faculty through the classes and seminars I attended, and its staff has provided considerable help in all administrative matters. I would especially like to thank Verra Morgan, our Graduate Student Affairs Officer. She has solved many of my potentially disastrous problems, helped me navigate through bureaucracy, and genuinely cared about my well-being as a student, despite having hundreds of students to attend to.

On a personal note, I would like to thank all my friends who have helped, supported, or just shared laughs with me throughout my life. They include: the friends I have back in Hong Kong, who continue to keep in touch with me and welcome me when I go back to visit; the friends I make during my undergraduate years, who have all graduated but remain great friends with me; my brothers and sisters of the UCLA Chinese Christian Fellowship, who have treated me as one of them despite my “advanced” age; and my brothers and sisters of the Chinese United Methodist Church at Los Angeles, who have offered me spiritual and emotional support for the last two years. Unfortunately (or fortunately), the list of people is too long for me to write here!

More than ten years ago, when I was a secondary school student in Hong Kong, an alumnus named James Tong, who is a professor of the UCLA Political Science Department, came to visit. He encouraged some of us to pursue university studies in the United States, and recommended me to come to UCLA. He has been instrumental in introducing me and many other students to the opportunity of studying at universities such as UCLA, and finding financial support for many of them. He also enlisted the help of many kind people, including Fred Luk, Norma and Howard Lee, and Barbara and Gerald Lee (and others), who offered hospitality by inviting us to their homes, and eased our transition to a new environment. Therefore, I would like to express my appreciation to all of them.

Last but certainly the most important, I owe a great debt to my family members: my brother Kong, my father, and my mother. My brother has been a great companion to me, my father has worked tirelessly to financially support the whole family, including my studies, and my mother taught me many things during my childhood, has worked hard with my father, and has taken great care of me even to this day. Their love and continual support on all levels of my life are priceless.

Finally, as I will soon leave UCLA after studying here for ten years, I must say I have enjoyed every moment of it, and I will remain a Bruin forever!

Vita

March 30, 1978	Born in Jiangxi Province, China
1998	B.Sc., Computer Science and Engineering, University of California, Los Angeles
2000–2005	Graduate Student Researcher, Department of Computer Science, University of California, Los Angeles
2001	M.Sc., Computer Science, University of California, Los Angeles
2002	Dimitris N. Chorafas Foundation Prize, in recognition of work on sensitivity analysis
Fall 2004	Teaching Assistant of the class *Fundamentals of Artificial Intelligence*, Department of Computer Science, University of California, Los Angeles

Publications

Hei Chan and Adnan Darwiche. “When Do Numbers Really Matter?” In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 65–74, 2001.

Hei Chan and Adnan Darwiche. “A Distance Measure for Bounding Probabilistic Belief Change.” In *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI)*, pp. 539–545, 2002.

Hei Chan and Adnan Darwiche. “When Do Numbers Really Matter?” *Journal of Artificial Intelligence Research*, **17**:265–287, 2002.

Hei Chan and Adnan Darwiche. “On the Revision of Probabilistic Beliefs Using Uncertain Evidence.” In *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 99–105, 2003.

Hei Chan and Adnan Darwiche. “Reasoning About Bayesian Network Classifiers.” In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 107–115, 2003.

Hei Chan and Adnan Darwiche. “Sensitivity Analysis in Bayesian Networks: From Single to Multiple Parameters.” In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 67–75, 2004.

Hei Chan and Adnan Darwiche. “A Distance Measure for Bounding Probabilistic Belief Change.” *International Journal of Approximate Reasoning*, **38**:149–174, 2005.

Hei Chan and Adnan Darwiche. “On the Revision of Probabilistic Beliefs Using Uncertain Evidence.” *Artificial Intelligence*, **163**:67–90, 2005.

Hei Chan and Adnan Darwiche, 2005. “Sensitivity Analysis in Markov Networks.” In *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1300–1305, 2005.

Abstract of the Dissertation

Sensitivity Analysis of Probabilistic Graphical Models

by

Hei Chan

Doctor of Philosophy in Computer Science

University of California, Los Angeles, 2005

Professor Adnan Darwiche, Chair

Probabilistic belief systems are used in artificial intelligence to model uncertainty. A popular framework for realizing probabilistic belief systems is to use graphical models, such as Bayesian networks and Markov networks. The topic of sensitivity analysis is concerned broadly with the relationships between local beliefs, such as network parameters, and global beliefs, such as values of probabilistic queries. Sensitivity analysis is crucial to probabilistic belief systems because we often need to revise our state of belief to incorporate new probabilistic information in the form of local belief changes. This work focuses on sensitivity analysis of probabilistic graphical models, by addressing central research problems such as the assessment of global belief changes due to local belief changes, the identification of local belief changes that induce certain global belief changes, and the quantifying of belief changes in general. Our results can be divided into the following parts. First, we develop procedures and complexity results for tuning Bayesian or Markov network parameters (single or multiple) to ensure certain query constraints. Second, we provide network-independent bounds on changes in query values due to arbitrary changes in Bayesian or Markov network parameters. Third, we propose a new distance measure for quantifying probabilistic belief changes, and use it to provide guarantees on global belief changes in Bayesian or Markov networks. Fourth, we provide algorithms and complexity results on the sensitivity of decisions induced by Bayesian networks. Finally, we discuss the philosophical topic of belief revision. Many of our results have been implemented in a program called SamIam (Sensitivity Analysis, Modeling, Inference and More), a graphical Bayesian network tool developed by the UCLA Automated Reasoning Group.

CHAPTER 1

Introduction

Probabilistic belief systems are used in artificial intelligence to model uncertainty. A popular framework for realizing probabilistic belief systems is to use graphical models, such as Bayesian networks and Markov networks, which capture uncertain knowledge naturally and efficiently, and are widely used in various fields such as medicine and diagnosis-and-repair modules. A probabilistic state of belief, in terms of a joint probability distribution, can be induced from a complete specification of the graphical model, consisting of a network structure that specifies the conditional independence relationships in the domain, and a set of network parameters that quantify the degrees of influence between variables.

Given a probabilistic belief system, we can make a distinction between the *local beliefs* and the *global beliefs* captured by the state of belief. For example, the network parameters, which measure probabilities or compatibilities between variables within localized frames such as cause-and-effect relationships, are viewed as local beliefs. They constitute the building blocks for representing the state of belief, and are estimated by experts or from statistical data in order to construct the probabilistic belief system. On the other hand, given a state of belief, we can perform inference and compute probability values of query propositions on the global scale of the domain. These probabilistic queries are viewed as global beliefs, which are the conclusions drawn from the probabilistic belief system, and provide useful answers under various scenarios. Therefore, the goal of constructing a probabilistic belief system is to infer global beliefs that faithfully correspond to the domain by accurately estimating local beliefs.

The topic of sensitivity analysis is concerned broadly with how the outputs of a system depends upon its input parameters. In the case of a probabilistic belief system, we can interpret the local beliefs and global beliefs as its inputs and outputs. This work is a comprehensive study of the theory and applications involved in the sensitivity analysis of probabilistic belief systems, with the focus mostly on the graphical models of Bayesian networks and Markov networks, by examining the relationships between local beliefs and global beliefs. In particular, it addresses these central research problems:

- The assessment of global belief changes due to local belief changes, such as the impact on values of certain queries due to perturbations in network parameters, or when some localized evidence is obtained;
- The identification of local belief changes that induce certain global belief changes, such as the set of network parameter changes that can enforce a certain query constraint;
- The quantifying of belief changes in general, such that changes in local and global beliefs can be measured and compared meaningfully and efficiently.

Moreover, this work also discusses the philosophical topic of belief revision, and how it is related to sensitivity analysis.

In this chapter, we will first present the fundamentals of probabilistic belief systems, including the syntax and semantics of the graphical models of Bayesian networks and Markov networks. We will then introduce the concepts of local and global belief changes, and the topic of sensitivity analysis, and explain why sensitivity analysis is important to probabilistic belief systems. We will also present the central research problems concerning sensitivity analysis, a brief summary of the results we have obtained, and the applications facilitated by these results.

1.1 Probabilistic Belief Systems

In artificial intelligence, we often have to deal with uncertainty. This may be due to a lack of understanding of the environment, incomplete information, and/or inaccurate results of sensors and tests. For example, in medical diagnosis, we must take into account the facts that the medical theory we have may not be complete, we do not have complete information about the patient, and the tests we run may be inaccurate.

To represent uncertain knowledge, we must be able to provide degrees of belief in propositions. A common way to do this is to use probabilities, which can summarize the uncertainty that comes from our ignorance. For example, given a patient with a certain disease, we may believe that there is a probability of .8 that a certain symptom will appear. This probability value can be derived from statistical data or some general rules, and it summarizes all cases where the disease and the symptom may or may not be connected. The use of probabilities to model uncertainty gives rise to probabilistic belief systems.

To construct a probabilistic belief system, we first have to define a set of random variables, which are used to represent the pieces of information we are interested in or may acquire knowledge about, such as whether the patient has a certain disease, or the result of a blood test. Each variable has a domain of possible values it can take on.[1] For example, the result of a blood test may take on either of two values: positive or negative. An instantiation of all variables in

[1] We will only deal with discrete values in this work.

the probabilistic belief system is called a *world* (or atomic event). Each world describes a unique and complete specification of the domain that we are concerned with.

To model uncertainty in a probabilistic belief system, we need to define a *joint probability distribution*, which assigns probabilities to all possible worlds.[2] The joint probability distribution captures our state of belief, and from it we can compute the probability of any statement using basic probability theory.

Throughout this work, we will use the following convention. Random variables are denoted by upper-case letters (A) and their values are denoted by lower-case letters (a). Sets of random variables are denoted by bold-face upper-case letters ($\mathbf{A}$) and their instantiations are denoted by bold-face lower-case letters ($\mathbf{a}$). For a binary random variable A with values *true* and *false*, we use a to denote the event $A = \textit{true}$ and $\bar{a}$ to denote the event $A = \textit{false}$.

The following example illustrates a probabilistic belief system.

Example 1.1 *Mr. Holmes has just installed a new security alarm at his house, and he would like to know the probability of the alarm being triggered on any given day, and how the alarm may be affected by a burglar breaking into his house or the occurrence of an earthquake. The manufacturer of the alarm provides him a joint probability distribution, shown in Table 1.1. In this probabilistic belief system, there are three variables: A, which represents whether the alarm of Mr. Holmes' house is triggered (values a and $\bar{a}$); B, which represents whether there is a burglary at his house (values b and $\bar{b}$); and E, which represents whether there is an earthquake (values e and $\bar{e}$). The joint probability distribution consists of the probabilities of the eight possible worlds, which sum up to* 1*. For example,*

[2] Because the worlds are mutually exclusive and collectively exhaustive, due to the axioms of probability, the sum of the probabilities of all worlds must be 1.

A	B	E	$Pr(\omega)$
a	b	e	.0000019
a	b	$\bar{e}$	.00093812
a	$\bar{b}$	e	.00057942
a	$\bar{b}$	$\bar{e}$	.000997002
$\bar{a}$	b	e	.0000001
$\bar{a}$	b	$\bar{e}$	.00005988
$\bar{a}$	$\bar{b}$	e	.00141858
$\bar{a}$	$\bar{b}$	$\bar{e}$	.996004998

Table 1.1: The joint probability distribution defined by the probabilistic belief system in Example 1.1.

the probability that the alarm is triggered, a burglar breaks into his house, and there is no earthquake on any given day, is $Pr(a, b, \bar{e}) = .00093812$. *From this joint probability distribution, we can also answer various queries. For example, to compute the probability of the alarm being triggered on any given day,* $Pr(a)$, *we add up the probabilities of all worlds* ω *which are consistent with the instantiation* $A = a$ *(denoted by* $\omega \sim a$*),*[3] *giving us* $Pr(a) = \sum_{\omega \sim a} Pr(\omega) \approx .0025$. *We can also easily compute conditional queries, such as the probability that there is a burglary at his house given the alarm is not triggered,* $Pr(b \mid \bar{a})$, *by first obtaining both* $Pr(b, \bar{a})$ *and* $Pr(\bar{a})$ *by adding up the probabilities of the consistent worlds, then computing their ratio, giving us* $Pr(b \mid \bar{a}) = Pr(b, \bar{a})/Pr(\bar{a}) \approx .00006$.

[3] *Two instantiations are consistent iff they do not assign different values to the same variable.*

1.2 Probabilistic Graphical Models

In Example 1.1, we define a probabilistic belief system by directly specifying the joint probability distribution. There are two problems when we use this method in real life. First, the size of the distribution grows exponentially with the number of variables, making it impractical to directly specify the entire distribution when given even a moderate number of variables. Second, it is usually quite unnatural and difficult to estimate directly the probabilities of all worlds unless we are given a large amount of statistical data. Therefore, we need a more natural and efficient method of specifying the joint probability distribution.

To do this, we need to exploit the presence of conditional independence relationships between the variables. For example, from the joint probability distribution shown in Table 1.1, we find that the probability of having a burglary (or not) is not affected by the presence (or absence) of earthquake, and vice versa:[4]

$$\begin{aligned}
Pr(b) &= Pr(b \mid e) &&= Pr(b \mid \bar{e}); \\
Pr(\bar{b}) &= Pr(\bar{b} \mid e) &&= Pr(\bar{b} \mid \bar{e}); \\
Pr(e) &= Pr(e \mid b) &&= Pr(e \mid \bar{b}); \\
Pr(\bar{e}) &= Pr(\bar{e} \mid b) &&= Pr(\bar{e} \mid \bar{b}).
\end{aligned}$$

This means burglary and earthquake are independent causes of the alarm being triggered. Conditional independence relationships between variables are common in real-world scenarios. For example, in the medical domain, a disease is caused by only a small number of risk factors that are often independent of each other, and the appearance of a symptom is dependent only on whether the patient has a certain number of diseases.

Conditional independence relationships can be captured naturally and efficiently by graphical models, using a graph that connects variables that are di-

[4] We ignore the possibility of looting when earthquakes occur.

rectly related with each other. In order to induce the joint probability distribution from the graph, we now only need to specify a set of parameters that quantify the degrees of dependence between the related variables. This significantly reduces the number of parameter values we need to directly specify from exponential to polynomial in the number of variables. Moreover, these parameter values can be easily estimated by experts using their knowledge or local information with regard to the dependence between the variables.

In this section, we will introduce two popular probabilistic graphical models: Bayesian networks and Markov networks. In both models, there are two components, a qualitative part and a quantitative part. In the qualitative part, a graph called the *network structure* is used to represent the interactions between variables, such that variables with direct interactions are connected by edges, and conditional independence relationships can be inferred from the network structure. In the quantitative part, a set of *network parameters* are used to quantify the degrees of dependence between related variables. The joint probability distribution can then be induced from the two components, although the process of computing the distribution differs between Bayesian networks and Markov networks.

1.2.1 Bayesian Networks

A *Bayesian network* (or belief network) $\mathcal{B} = (G, \Theta)$ consists of two parts: the network structure in terms of a *directed acyclic graph*, and the parametrization in terms of a set of *conditional probability tables (CPTs)*.

In the directed acyclic graph G, each random variable is represented by a node, and there is a directed edge from one node to another if there is a direct influence of the former variable on the latter variable.[5] The absence of an edge between two variables means any potential interaction between them is indirect and conditional upon other variables. Therefore, the network structure encodes the conditional independence relationships between variables. A test called *d-separation* can be used to test if a variable X is conditionally independent of another variable Y given a third variable Z [Jen01, Pea88]. For variable X, we define its *parents* $\mathbf{U}$ as the set of variables that have outgoing directed edges pointing to X.

In the parametrization Θ, a conditional probability table (CPT) is specified for every variable X in the Bayesian network, which quantifies our belief in the conditional probability of every value of X given every instantiation of its parents $\mathbf{U}$. The CPT of X, $\Theta_{X|\mathbf{U}}$, assigns a number to the network parameter $\theta_{x|\mathbf{u}}$ for every value x of X and every instantiation $\mathbf{u}$ of $\mathbf{U}$, such that $\theta_{x|\mathbf{u}} = Pr(x \mid \mathbf{u})$. Due to the axioms of probability, the probabilities in the conditional probability distribution $\Theta_{X|\mathbf{u}}$ must sum to 1 for every $\mathbf{u}$, i.e., $\sum_x \theta_{x|\mathbf{u}} = 1$.

The structure and the CPTs of a Bayesian network $\mathcal{B}$ induce a joint probability distribution Pr. The probability of a world, i.e., an instantiation $\mathbf{x}$ of all variables in the Bayesian network $\mathbf{X}$, is the product of all network parameters $\theta_{x|\mathbf{u}}$ where $\{x, \mathbf{u}\}$ is consistent with $\mathbf{x}$:

$$Pr(\mathbf{x}) \stackrel{\text{def}}{=} \prod_{\{x,\mathbf{u}\}\sim\mathbf{x}} \theta_{x|\mathbf{u}}. \tag{1.1}$$

We now present an example of a Bayesian network.

Example 1.2 *Instead of directly specifying the joint probability distribution as in Example 1.1, the manufacturer of the alarm now decides to use a Bayesian*

[5] The direction of the edge usually implies that the influence from the former variable to the latter variable is causal, although this is not always true.

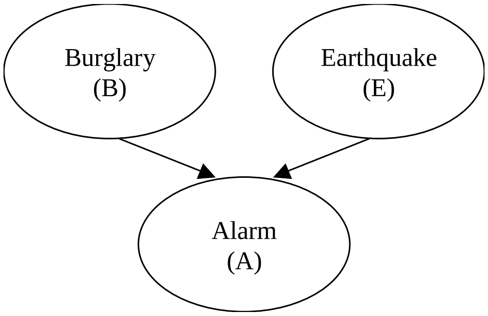


Figure 1.1: The structure of the Bayesian network Alarm.

network, called Alarm*, to model the uncertainty knowledge. After looking at the joint probability distribution shown in Table 1.1, it concludes that burglaries and earthquakes are direct and independent causes of whether an alarm will be triggered. Therefore, we can use the network structure shown in Figure 1.1, with directed edges from B to A and from E to A, to represent the conditional independence relationships between the variables. The CPTs of the Bayesian network* Alarm *are shown in Table 1.2. The CPTs of B and E specify the prior probability distribution of these two variables, and the CPT of A specifies the conditional probabilities of the possible values of A given different instantiations of B and E. Using Equation 1.1, we can easily verify that the joint probability distribution induced by this Bayesian network is the same as the one shown in Table 1.1. For example, the probability of $a, \bar{b}, \bar{e}$ is:*

$$\begin{aligned} Pr(a, \bar{b}, \bar{e}) &= \theta_{a|\bar{b},\bar{e}} \cdot \theta_{\bar{b}} \cdot \theta_{\bar{e}} \\ &= .001 \times .999 \times .998 \\ &= .000997002. \end{aligned}$$

B	Θ_B
b	.001
$\bar{b}$	.999

E	Θ_E
e	.002
$\bar{e}$	.998

B	E	A	$\Theta_{A\|B,E}$
b	e	a	.95
b	e	$\bar{a}$	.05
b	$\bar{e}$	a	.94
b	$\bar{e}$	$\bar{a}$	.06
$\bar{b}$	e	a	.29
$\bar{b}$	e	$\bar{a}$	.71
$\bar{b}$	$\bar{e}$	a	.001
$\bar{b}$	$\bar{e}$	$\bar{a}$	.999

Table 1.2: The CPTs of the Bayesian network Alarm.

1.2.2 Markov Networks

A *Markov network* $\mathcal{M} = (G, \Theta)$ consists of two parts: the network structure in terms of an *undirected graph*, and the parametrization in terms of a set of *clique tables*.

In the undirected graph G, each random variable is represented by a node, and there is an undirected edge connecting two nodes if there is a direct interaction between them. The absence of an edge between two variables means any potential interaction between them is indirect and conditional upon other variables. In the network structure, we define a *clique* $\mathbf{C}$ as a maximal set of variables where every pair of variables in the set $\mathbf{C}$ is connected by an edge.

In the parametrization Θ, a clique table is specified for every clique $\mathbf{C}$ in the Markov network, which quantifies the degrees of interactions between the clique variables. The clique table of $\mathbf{C}$, $\Theta_{\mathbf{C}}$, assigns a non-negative number to the network parameter $\theta_{\mathbf{c}}$ for every instantiation $\mathbf{c}$ of $\mathbf{C}$, such that it measures the relative degree of compatibility associated with $\mathbf{c}$.[6]

The structure and the clique tables of a Markov network $\mathcal{M}$ induce a joint probability distribution Pr. If $\mathbf{X}$ is the set of all variables in the Markov network, the joint potential ψ over $\mathbf{X}$ is defined as:

$$\psi(\mathbf{x}) \overset{\text{def}}{=} \prod_{\mathbf{c} \sim \mathbf{x}} \theta_{\mathbf{c}}. \tag{1.2}$$

That is, $\psi(\mathbf{x})$ is the product of all network parameters $\theta_{\mathbf{c}}$ where $\mathbf{c}$ is consistent with $\mathbf{x}$. The joint probability distribution Pr induced by the Markov network is then defined as the normalized joint potential over $\mathbf{X}$:

$$Pr(\mathbf{x}) \overset{\text{def}}{=} \frac{\psi(\mathbf{x})}{\sum_{\mathbf{x}} \psi(\mathbf{x})} = \zeta\psi(\mathbf{x}), \tag{1.3}$$

where $\zeta = 1/\sum_{\mathbf{x}} \psi(\mathbf{x})$ is the normalizing constant. From Equation 1.3, we can easily verify that the specific parameter values in a clique table are not important, but their ratios are. This is a major departure from Bayesian networks, where specific parameter values in CPTs are important when computing the joint probability distribution.

We now present an example of a Markov network.

Example 1.3 *Consider four individuals $\{A_1, A_2, B_1, B_2\}$, and a contagious disease that may be transmitted between them through their interactions. To model the transmission of the disease between the individuals, we use a Markov network, called* Disease*, where the presence or absence of the disease in each individual is represented by a binary variable of the same name. Its network structure is shown in Figure 1.2. Every pair of variables $\{A_i, B_j\}$ is connected by an edge, representing the fact that these pairs of individuals are in direct contact with each*

[6] Note that Markov network parameters are not probabilities, and may be estimated on an ad hoc basis [GG84] or by statistical models. It can be difficult to assign meanings and numbers to Markov network parameters intuitively [Pea88, pp. 107–108].

A_i	B_j	Θ_{A_i,B_j}
a_i	b_j	1
a_i	$\bar{b_j}$	2
$\bar{a_i}$	b_j	3
$\bar{a_i}$	$\bar{b_j}$	4

Table 1.3: The clique table of $\{A_i, B_j\}$ of the Markov network Disease.

other. The pairs $\{A_1, A_2\}$ and $\{B_1, B_2\}$ are not connected by an edge, meaning they do not interact directly with each other, although the disease can still be transmitted between them indirectly through the other two individuals. There are four cliques in this graph: $\{A_1, B_1\}$, $\{A_1, B_2\}$, $\{A_2, B_1\}$, and $\{A_2, B_2\}$. For every clique $\{A_i, B_j\}$, the same clique table Θ_{A_i,B_j} is used to specify the parameter values, and is shown in Table 1.3. The parameter values indicate the level of compatibility between the health of individuals who are in direct contact with each other. For example, it can be seen that both A_i and B_j not having the disease is four times more compatible than both A_i and B_j having the disease, since $\theta_{\bar{a_i},\bar{b_j}} = 4\theta_{a_i,b_j}$. Using Equation 1.3, we can compute the probability of any world. For example, given the normalizing constant $\zeta = 1/\sum_{\mathbf{x}} \psi(\mathbf{x}) = 1/836$, the probability of $a_1, \bar{a_2}, b_1, \bar{b_2}$ is:

$$\begin{aligned} Pr(a_1, \bar{a_2}, b_1, \bar{b_2}) &= \zeta \cdot \theta_{a_1,b_1} \cdot \theta_{a_1,\bar{b_2}} \cdot \theta_{\bar{a_2},b_1} \cdot \theta_{\bar{a_2},\bar{b_2}} \\ &= \frac{1}{836} \times 1 \times 2 \times 3 \times 4 \\ &\approx .0287. \end{aligned}$$

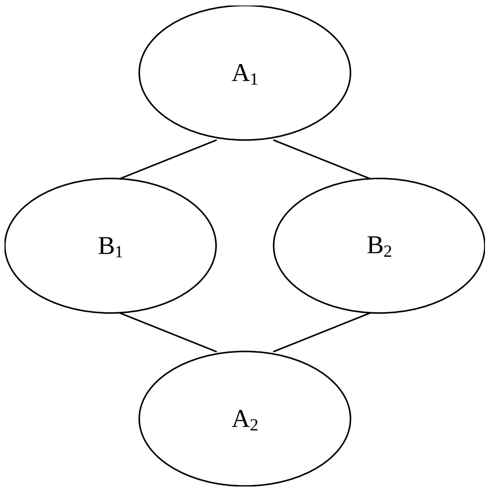


Figure 1.2: The structure of the Markov network Disease.

1.3 Local Belief Changes

As we are using a probabilistic belief system to model an uncertain environment, we often receive new probabilistic information that differs from our current state of belief. This can be due to more accurate sensory inputs, a better understanding of the domain, or simply changes in the modeled environment. The new probabilistic information that we need to incorporate is usually in the form of changes in local beliefs, which are intuitive and central to the construction of the probabilistic belief system. Examples of local beliefs include the probability distribution of a random variable, or the conditional probability specified by a Bayesian network parameter. In order to incorporate the new probabilistic information, we have to revise our state of belief according to the local belief change, and as a result, our global beliefs, such as the values of probabilistic queries, will be changed.

We now show two examples of local belief changes in probabilistic belief systems: soft evidence and Bayesian network parameter changes.

1.3.1 Soft Evidence

Suppose we are given a joint probability distribution, which we can use to compute the probability distribution of some variable X. We are now given some new probabilistic information, and after that we conclude the probability distribution of X should differ from the one computed from the current joint probability distribution. This piece of *soft evidence* (or uncertain evidence) is an example of local belief changes.[7] We must now adopt a new joint probability distribution that incorporates this piece of soft evidence.

For example, consider again the probabilistic belief system given in Example 1.1. Due to the result of a survey among customers, the manufacturer of the alarm now wants to change the probability of the alarm being triggered on any given day, $Pr(a)$, from the current value of .0025 to a new value of .005. To do this, the joint probability distribution shown in Table 1.1 must be changed to incorporate this piece of soft evidence.

The modification of a state of belief due to soft evidence has been addressed by the topic of *belief revision* [G88, Jef65, Jef92], which deals with two issues. The first issue is how we should specify soft evidence. As we will see later, the method of specifying soft evidence can be quite important. The second issue is how we should choose the new joint probability distribution. As there are infinitely many joint probability distributions that satisfy the soft evidence, our task is to commit to a principle that chooses the distribution that is the "best", or "closest" to the initial distribution, so that our state of belief is disturbed the least by the revision. Afterwards, sensitivity analysis can be used to provide guarantees on the change in global beliefs due to the revision process.

[7] This is in contrast to hard evidence, where we know for certain that a variable has taken on some value.

1.3.2 Bayesian Network Parameter Changes

Similar to general probabilistic belief systems, graphical models such as Bayesian networks also often have to be updated due to new probabilistic information. If we assume that the mechanisms of influence and dependence between variables remain the same, we will continue to use the same network structure, but change some of the network parameters to incorporate the new information. This means we often have to find changes in the network parameters that can help us model the new environment. For example, updating the reliability of a sensor translates to changing parameters in the CPT of the variable that represents the sensor.

Parameter changes are also an integral part of building a Bayesian network. Consider for example that we are constructing a Bayesian network to faithfully reflect a domain of interest, such as medicine. First, the network structure is developed from the knowledge of the dependence between risk factors, diseases, and symptoms. Next, parameters are estimated by non-experts using a combination of statistical data and qualitative influences available from textbook materials. Finally, medical experts are brought in to evaluate the network and fine-tune its parameters. One method of evaluation is to pose diagnostic scenarios to the network, and compare the results of such queries to those predicted by the experts. For example, given some set of symptoms $\mathbf{e}$, and two potential diagnoses y and z, the current network may give us the conclusion that the ratio $Pr(y \mid \mathbf{e})/Pr(z \mid \mathbf{e})$ is 2, while an expert may believe that the ratio should be no less than 4. We must now apply necessary parameter changes in order to satisfy the values given by the experts. This refining of network parameters by posing query constraints can also be classified as local belief changes [CPO99].

The first task we need to perform here is to efficiently find parameter changes that we can apply such that the new Bayesian network satisfy the constraints posed by the experts. Otherwise, the experts may have to spend a huge amount of time tuning each network parameter, and it is often the case that many parameters are irrelevant to satisfying the constraints.

The parameter changes we aim to find can be of different types. The simplest type is a single parameter change, where we change only one network parameter at a time (plus the co-varying parameters that must be changed in order to satisfy the sum-to-one constraint). This type of parameter changes is the easiest to compute and can act as indications of which parameters are more relevant to the query, yet may be too constraining to be applicable.

The second type of parameter changes is a single CPT change, where all parameters in a single CPT are changed. Single CPT changes are more intuitive and useful than single parameter changes, and do not require much more complexity in computation, as we will show later. Because multiple parameters are involved, the whole solution space of parameter changes can be too complicated to report. Instead, we should report a particular point in the solution space, and the problem becomes how we should find this particular parameter change.

The third type of parameter changes is a multiple CPT change, where all parameters in multiple CPTs are changed. This type of change is the hardest to find as it involves computation with higher-order complexity and more complicated procedures. Therefore, it is not realistic to use this procedure for every combination of CPTs. Instead, we would like to identify a process to find subsets of CPTs that should give meaningful results to the process of sensitivity analysis.

Finally, even though we have assumed that the network structure remains the same, changes in the network structure can sometimes be treated as network parameter changes. For example, deleting an edge from a Bayesian network is the same as changing the parameters in a CPT such that new conditional independence relationships between variables are reflected by the new parameter values.

The second task we need to perform is to choose among various parameter changes that can enforce the constraints. These parameter changes can be in the form of single parameter changes or multiple parameter changes. Usually, the experts will adopt parameter changes that do not disturb the state of belief much from the initial distribution induced by the Bayesian network.

1.4 Global Belief Changes

From the examples in Section 1.3, we can see that we often need to revise our state of belief in order to incorporate local belief changes, and obtain a new joint probability distribution different from the initial one. Given the initial and new joint probability distributions, we can assess how the global beliefs captured have changed as a result. For example, we may ask how much the value of a certain probabilistic query has changed. Alternatively, we may focus on a subset of queries, such as those where the evidence is the same, or those of the same event conditioned on different instantiations of the same evidence variables.

Sometimes, instead of how the exact value of a probabilistic query changes, we may be interested in the sensitivity of decisions made from the query value. For example, we may diagnose a certain disease given a set of symptoms if the probability of the disease under that scenario is more than .5, or we may choose a treatment from various choices such that the probability of recovery is the greatest. In this case, our interest in global belief changes shifts from dealing with continuous probability values to discrete decision values. When comparing the initial and new joint probability distributions, we are now interested in whether the new decisions made are different from the initial ones, or remain the same despite the local belief changes.

Therefore, depending on our scope of interest, the set of global belief changes we want to focus on can be quite different. In the most general case, the focus of our investigation is on how the value of any query $Pr(\alpha \mid \beta)$, where α and β are arbitrary events, differs under the initial and new joint probability distributions. Any guarantees that can be computed are applicable to arbitrary queries, and the results can usually be found using the minimum amount of computation.

1.5 Sensitivity Analysis of Probabilistic Belief Systems

The goal of sensitivity analysis of probabilistic belief systems is to investigate the impact of local belief changes on global beliefs. We now go through the various elements of sensitivity analysis that we will develop in this work.

1.5.1 From Local to Global Belief Changes

If we are given some new probabilistic information, and we are committed to a certain method of applying local belief changes, an interesting question that arises is: what would be the impact on global beliefs if we apply some local belief change? We answer this question by looking at how the value of any probabilistic query, $Pr(\alpha \mid \beta)$, where α and β are arbitrary events, changes after applying the local belief change.

This analysis can be done in two different contexts. If we are given a specific probabilistic belief system and a specific query, we can always simply compute the result before and after the local belief change. However, this only works for specific events α and β, as we cannot compute the results of all queries as there are too many of them.

Another approach we can take is to perform a general analysis of the sensitivity of any probabilistic query to a local belief change, even if we do not know the details of the probabilistic belief system. For example, we may know that we have changed a parameter in some Bayesian network from its current value of .02 to a new value of .03, but know neither its network structure nor its CPTs. Given that the current value of some probabilistic query is .4, what can we guarantee about the new value of this query after the parameter change?

The difference between the two approaches is that when we are given more details about the probabilistic belief system or the class of queries we are interested in, we can get more accurate guarantees on the query changes, but we may need more computation. On the other hand, a general analysis of query sensitivity is useful when we only want to know a rough estimate of the impact of a local belief change. A major goal of this work is to provide a comprehensive answer of this problem, especially for Bayesian networks.

1.5.2 From Global to Local Belief Changes

Instead of inducing global belief changes from local belief changes, we can work in the opposite direction, and ask what local belief changes we may apply if we want certain global beliefs to change by a certain amount. For example, as we pointed out in Section 1.3 about the updating of Bayesian networks, we often need to find parameter changes that we can apply to ensure that a Bayesian network satisfies some query constraints.

We can again perform this analysis in two different contexts. We may know all the information about the probabilistic belief system, and obtain exact answers to our question. Alternatively, we can perform a general analysis, and obtain bounded answers to our question with less computation. A major goal of this work is to provide accounts of this analysis in both contexts.

1.5.3 Quantifying Local and Global Belief Changes

To further understand the relationships between local and global belief changes, we first need to settle a major issue. How do we measure belief changes, either local or global?

To emphasize the significance of quantifying belief changes, we go back to the two examples of local belief changes in Section 1.3. Let us first look at the case where we want to incorporate some new soft evidence. As we are faced with infinitely many choices of new joint probability distributions that we can adopt, we want to adopt one that is the "closest" to the current joint probability distribution. This calls for a measure between two probability distributions, which must quantify the amount of difference between the two distributions, in terms of the degree of changes in global beliefs. We can then adopt a new joint probability distribution that minimizes this measure from the current joint probability distribution.

To compute this measure between two probability distributions, we must examine the probabilities of every world under the two distributions. We can use a measure that is computed using a weighted sum, such that worlds with higher probabilities are assigned heavier weights. KL-divergence is an example of this type of measure [KL51]. However, it can be argued that unlikely scenarios are as important as likely scenarios when capturing global belief changes. For example, the scenario of an accident happening, which is unlikely, is crucial in the modeling of a disaster management system. Therefore, it can be worthwhile to consider the change in every conditional query, no matter how unlikely the evidence may be, when capturing global belief changes.

After choosing a measure, if we decide to commit to a principle of belief revision that minimizes this measure, our goal then is to compute the global measure between the initial and new joint probability distributions efficiently by only examining the local belief changes. Therefore, we must first quantify the local belief changes, then provide a formula that computes the global measure from this local measure. The significance of this process is that it can help us understand the relationships between local and global belief changes, as discussed previously in this section. Moreover, this local measure can also help us define the notion of evidence strength. For example, we are often faced with several alternative pieces of soft evidence, and we may want to choose one that gives us the "minimal" change.

The same principle applies when we are given several candidate Bayesian network parameter changes. Because the joint probability distribution is uniquely induced by the parameters, the measure between the initial and new distributions induced can be determined from the parameter change chosen. The formula computing the global measure from the local measure can be used for quantifying parameter changes, and is useful when we want to adopt a parameter change such that the new Bayesian network is the "closest" to the initial Bayesian network.

The measure we propose for quantifying belief changes is a distance measure between two probability distributions that we will introduce later. We will show that this distance measure is both useful because it gives us the ability to bound any query change between two distributions, and easily computable because it allows the conversion between local and global measures for both soft evidence and Bayesian network parameter changes.

1.6 Applications

We now present a few applications of sensitivity analysis of probabilistic belief systems. First of all, as we have repeatedly shown, sensitivity analysis is integral to understanding the relationships between local and global belief changes. The ability to infer global belief changes from local belief changes is quite useful because often after a local belief change, we do not want to redo inference completely as it can be expensive. Instead, we would like to compute an estimate of new query results from the initial query results and the local belief change without redoing the entire inference.

Our results can also help experts build better probabilistic belief systems. As experts have to estimate a large number of probabilities, such as network parameter values, it is often difficult and expensive to estimate all the probabilities with great accuracy. With sensitivity analysis, we can identify probabilities where a small change can greatly affect certain global beliefs, and those where even a large change does not affect these global beliefs much. This can help direct the experts to probability values that must be accurately estimated.

Understanding of sensitivity analysis is also useful in a number of areas, including model debugging and system design. In model debugging, we want to identify parameters that are relevant to certain queries, or to find parameter changes that would be necessary to enforce certain sanity checks on the values of probabilistic queries. In system design, sensitivity analysis can be used to choose false-positive and false-negative rates for sensors and tests, in order to ensure the quality of an information system based on a Bayesian network.

Moreover, our results can be applied in the domain of approximate reasoning. For example, after learning the parameter values of a Bayesian network from statistical data, we may want to round them off to a certain degree of accuracy. Using our results on sensitivity analysis, we can provide robustness assurances of the query results after the approximation.

We can also often approximate a probabilistic belief system by a simpler one that can help us save computational resources. For example, a Bayesian network that is highly connected and contains many cycles is often computationally too expensive to perform inference on. Instead, we can delete edges from the network structure, and obtain another Bayesian network with a simpler network structure, such that inference can be performed in less time. Even though the network structure has been changed, we can consider this as a local belief change in the form of parameter changes, by interpreting the edge deletions as changes in the CPTs of the child nodes where new conditional independence relationships are introduced. Therefore, the process of edge deletion can be interpreted as an approximation of our state of belief. Previous work on edge deletion has focused mainly on the reduction of complexity without providing guarantees on the query results [Kjae94, van97]. Using our sensitivity analysis results, we can compute the bound on query changes due to the approximation, and can provide guarantees on the real query results from the approximated query results.

Finally, by providing a measure that quantifies belief changes, we can help humans visualize belief changes and make better decisions based on them. Many graphical Bayesian network tools do a good job of providing users with visual cues indicating the current state of belief, but have not paid any attention to the change in the state of belief due to new probabilistic information. However, it has been shown that users of these tools have difficulty visualizing belief changes because of the large amount of information usually on display, the transient nature of the different states of belief, and the unfamiliarity of probability theory.

The measure we provide can thus be a great help for users for many purposes, such as gauging the strength of some new evidence, or comparing the impact of a parameter change on different query values. We have experimented with providing numerical and visual indicators of belief changes based on this measure in a graphical Bayesian network tool called SamIam [ACC].

1.7 Overview

The following chapters are laid out as follows:

- In Chapter 2, we start our work on sensitivity analysis of Bayesian networks, as we solve the problem of tuning single parameters to enforce a query constraint in a given Bayesian network. We provide a procedure to do so, present its complexity results, and its implementation with a graphical user interface.

- In Chapter 3, we expand the question of sensitivity analysis to arbitrary Bayesian networks in general, as we investigate the global impact of parameter changes, by first bounding the derivative of a query with respect to a parameter, then bounding the change in a query due to an arbitrary parameter change. We also use the results for an intuitive understanding of when parameter changes do or do not matter in terms of query robustness.

- In Chapter 4, we discuss how we can quantify belief changes between probabilistic states of belief, and aim to bound the impact on global belief changes due to local belief changes in general probabilistic belief systems. To do this, we propose a distance measure that allows us to quantify belief differences between two probabilistic distributions. We compare this measure with existing measures, and show its applications to Bayesian networks.

- In Chapter 5, we expand our work on tuning Bayesian network parameters to enforce a query constraint from the domain of single parameters to multiple parameters, and discuss the additional work that is needed to solve the problem of finding multiple parameter changes that enforce query constraints in a given Bayesian network.

- In Chapter 6, we turn our attention to the sensitivity of decisions induced by Bayesian networks. In particular, we provide an algorithm that converts naive Bayes classifiers, which are simple in structure but commonly used for classification purposes, into logical representations, which allows reasoning for sensitivity analysis purposes in time polynomial in the sizes of the representations.

- In Chapter 7, we turn our focus of sensitivity analysis to Markov networks, and compare the similarities and differences between sensitivity analysis of Markov networks and Bayesian networks.

- In Chapter 8, we discuss the philosophical topic of belief revision. We look into two existing methods of belief revision given soft evidence, Jeffrey's rule and Pearl's method of virtual evidence, and use the distance measure proposed to bound the global belief change. We also discuss the similarities and differences between the two methods, in the principle used for belief revision, the specification of the soft evidence, and the interpretation of evidential statements.

- In Chapter 9, we summarize the contributions of this work, and present some final remarks regarding possible future directions of the research of sensitivity analysis of probabilistic graphical models.

- Proofs of all theorems are included in Appendix A.

CHAPTER 2

Tuning Single Bayesian Network Parameters

Sensitivity analysis of Bayesian networks is broadly concerned with understanding the relationships between local network parameters and global conclusions drawn based on the network. In this chapter, we will first discuss the process of finding single parameter changes that can help enforce query constraints in a given Bayesian network [CD01, CD02b]. This process is integral to building and debugging Bayesian networks, which we illustrate by the following example.

Suppose we are given a Bayesian network, called Fire, which is built by experts to model the possible scenario of whether there is a fire in a building. Its structure is shown in Figure 2.1 and its CPTs are shown in Table 2.1.[1] The experts now test the network by posing possible queries and comparing the results with those expected by the experts. For example, we may have evidence $\mathbf{e} = \{\mathit{Report} = \mathit{true}, \mathit{Smoke} = \mathit{false}\}$, i.e., the fire department receives a report that people are evacuating the building, but no smoke is observed. This evidence should make the occurrence of tampering more likely than the occurrence of fire, and the given Bayesian network does indeed reflect this with $Pr(\mathit{Tampering} = \mathit{true} \mid \mathbf{e}) = .501$ and $Pr(\mathit{Fire} = \mathit{true} \mid \mathbf{e}) = .0294$. However, the experts believe the probability of the alarm having been tampered should be no less than .65 given this evidence. Naturally, we must change some of the network parameters to satisfy this query

[1] This Bayesian network is distributed with the evaluation version of the commercial Hugin Graphical User Interface by Hugin Expert.

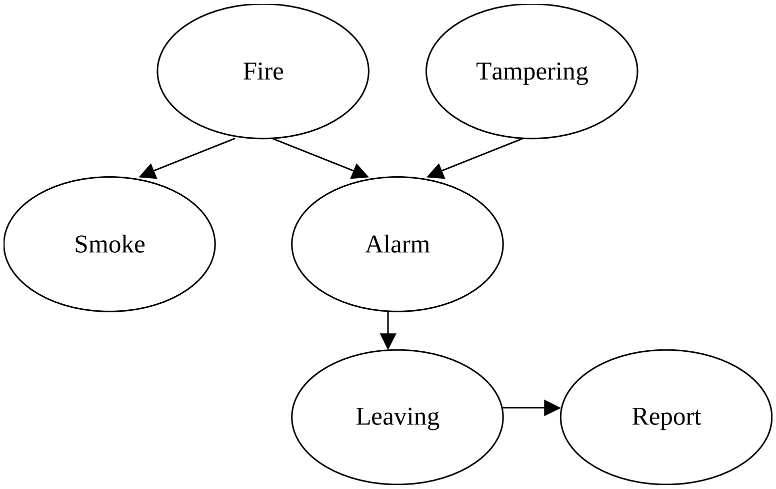


Figure 2.1: The structure of the Bayesian network Fire.

constraint, if we believe the network structure is indeed correct.

Suppose now the experts attempt to enforce the query constraint by changing only one network parameter.[2] The experts find out that either of the following two changes can be made to satisfy the constraint:

1. Increase the prior probability of alarm tampering, $Pr(\mathit{Tampering} = \mathit{true})$, from its current value .02 to $\geq$ .0364;

2. Decrease the probability of a false report given there is no evacuation, $Pr(\mathit{Report} = \mathit{true} \mid \mathit{Leaving} = \mathit{false})$, from its current value .01 to $\leq$ .00471.

The experts also find out that the parameters in the CPTs of variables *Fire*, *Smoke*, *Alarm*, and *Leaving* are irrelevant to satisfying the query constraint, i.e., no matter how much we change any single parameter in these CPTs, we would not be able to enforce the desired constraint.

[2] Because the probabilities of mutually exclusive and exhaustive events must sum to 1, the experts also have to change other parameters in the same CPT. Later, we will introduce the concept of meta-parameters to deal with this problem.

Fire	$\Theta_{X\|\mathbf{U}}$
true	.01
false	.99

Tampering	$\Theta_{X\|\mathbf{U}}$
true	.02
false	.98

Fire	*Smoke*	$\Theta_{X\|\mathbf{U}}$
true	*true*	.9
true	*false*	.1
false	*true*	.01
false	*false*	.99

Fire	*Tampering*	*Alarm*	$\Theta_{X\|\mathbf{U}}$
true	*true*	*true*	.5
true	*true*	*false*	.5
true	*false*	*true*	.99
true	*false*	*false*	.01
false	*true*	*true*	.85
false	*true*	*false*	.15
false	*false*	*true*	.0001
false	*false*	*false*	.9999

Alarm	*Leaving*	$\Theta_{X\|\mathbf{U}}$
true	*true*	.88
true	*false*	.12
false	*true*	.001
false	*false*	.999

Leaving	*Report*	$\Theta_{X\|\mathbf{U}}$
true	*true*	.75
true	*false*	.25
false	*true*	.01
false	*false*	.99

Table 2.1: The CPTs of the Bayesian network Fire.

Notice that here we are only interested in *single parameter changes*. However, it is also possible that we can change a combination of parameters to satisfy the constraint, which we will look at in a later chapter.

As the number of parameters increases with the number of variables in the Bayesian network, it is difficult for experts to manually find parameter changes that can enforce a query constraint. Therefore, we would like to find an efficient procedure to automatically find these parameter changes. We will first discuss this topic, then present an automated program we have developed that implements this procedure, called SamIam [ACC]

2.1 Procedure and Complexity of Tuning Single Parameters

In general, given a Bayesian network, we want to efficiently find parameter changes that can be made to satisfy constraints on probabilistic queries. Here are some of the most common types of query constraints, given some value κ:

$$\begin{aligned} Pr(y \mid \mathbf{e}) &\geq \kappa; && (2.1)\\ Pr(y \mid \mathbf{e}) &\leq \kappa; && (2.2)\\ Pr(y \mid \mathbf{e}) - Pr(z \mid \mathbf{e}) &\geq \kappa; && (2.3)\\ \frac{Pr(y \mid \mathbf{e})}{Pr(z \mid \mathbf{e})} &\geq \kappa. && (2.4) \end{aligned}$$

Here, evidence $\mathbf{e}$ is an instantiation of variables $\mathbf{E}$, and events y and z are values of the variables Y and Z respectively, with $Y, Z \notin \mathbf{E}$. These types of constraints often arise when we debug Bayesian networks. For example, if we want to make event y more likely than event z, given evidence $\mathbf{e}$, we can specify the constraint, $Pr(y \mid \mathbf{e}) - Pr(z \mid \mathbf{e}) \geq 0$. We can also make event y at least twice as likely as event z, given evidence $\mathbf{e}$, by specifying the constraint, $Pr(y \mid \mathbf{e})/Pr(z \mid \mathbf{e}) \geq 2$.

We now proceed to introduce a procedure that efficiently finds single parameter changes that can enforce a single query constraint.

2.1.1 Parameters of Binary Variables

We first consider the parameters in the CPT of a binary variable X, with two values x and $\bar{x}$, meaning there are two parameters $\theta_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}}$ for every parent instantiation $\mathbf{u}$. Because we must have $\theta_{x|\mathbf{u}} + \theta_{\bar{x}|\mathbf{u}} = 1$, it is not meaningful to change only $\theta_{x|\mathbf{u}}$ or $\theta_{\bar{x}|\mathbf{u}}$ without changing the other. Therefore, for every parent instantiation $\mathbf{u}$, we introduce a *meta-parameter* $\tau_{x|\mathbf{u}}$, and assign $\theta_{x|\mathbf{u}} = \tau_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}} = 1 - \tau_{x|\mathbf{u}}$. Our goal is then to determine the amount of change that must be applied to $\tau_{x|\mathbf{u}}$, which would lead to complementary changes in $\theta_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}}$, that can enforce the query constraint.

First we observe that the probability of an instantiation $\mathbf{e}$, $Pr(\mathbf{e})$, is a linear function of any parameter $\theta_{x|\mathbf{u}}$ in a Bayesian network [CGH97, RBK95]. In fact, the probability is linear in any meta-parameter $\tau_{x|\mathbf{u}}$, giving us the following theorem.

Theorem 2.1 *The derivative of $Pr(\mathbf{e})$ with respect to $\tau_{x|\mathbf{u}}$, which we define as $\pi^{\mathbf{e}}_{x|\mathbf{u}}$, is given by:*

$$\pi^{\mathbf{e}}_{x|\mathbf{u}} \stackrel{\mathrm{def}}{=} \frac{\partial Pr(\mathbf{e})}{\partial \tau_{x|\mathbf{u}}} = \frac{Pr(\mathbf{e}, x, \mathbf{u})}{\theta_{x|\mathbf{u}}} - \frac{Pr(\mathbf{e}, \bar{x}, \mathbf{u})}{\theta_{\bar{x}|\mathbf{u}}}, \tag{2.5}$$

if $\theta_{x|\mathbf{u}} \neq 0$ and $\theta_{\bar{x}|\mathbf{u}} \neq 0$.[3] *Since $Pr(\mathbf{e})$ is linear in $\tau_{x|\mathbf{u}}$, if we apply a change of $\Delta\tau_{x|\mathbf{u}}$ to $\tau_{x|\mathbf{u}}$, the change in $Pr(\mathbf{e})$ is given by:*

$$\Delta Pr(\mathbf{e}) = \pi^{\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}}. \tag{2.6}$$

[3] *If either of the parameters is zero, we can use the differential approach to compute this derivative [Dar03].*

In Equation 2.5, $\pi^{\mathbf{e}}_{x|\mathbf{u}}$ is a constant in terms of both $\theta_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}}$, and consequently also a constant in terms of $\tau_{x|\mathbf{u}}$.[4] The constant $\pi^{\mathbf{e}}_{x|\mathbf{u}}$, as well as $\pi^{y,\mathbf{e}}_{x|\mathbf{u}}$ and $\pi^{z,\mathbf{e}}_{x|\mathbf{u}}$, when applicable, is crucial to the procedure of finding the necessary change in $\tau_{x|\mathbf{u}}$ to enforce the query constraint.

We are now ready to find the single parameter changes that can enforce the query constraint, given that it is not satisfied by the current Bayesian network, which induces the probability distribution pr. For example, to enforce Inequality 2.1, it suffices to ensure that $Pr(y, \mathbf{e}) \geq \kappa \cdot Pr(\mathbf{e})$, or equivalently:

$$pr(y, \mathbf{e}) + \Delta Pr(y, \mathbf{e}) \geq \kappa \left(pr(\mathbf{e}) + \Delta Pr(\mathbf{e}) \right),$$

where $pr(\mathbf{e})$ and $pr(y, \mathbf{e})$ are the current probabilities of $\mathbf{e}$ and $y, \mathbf{e}$ respectively. If we apply a change of $\Delta\tau_{x|\mathbf{u}}$ to $\tau_{x|\mathbf{u}}$, from Equation 2.6, we have:

$$pr(y, \mathbf{e}) + \pi^{y,\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \geq \kappa \left(pr(\mathbf{e}) + \pi^{\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \right).$$

Rearranging the terms, we get the following result.

Corollary 2.1 *To satisfy Inequality 2.1, $Pr(y \mid \mathbf{e}) \geq \kappa$, we need to change $\tau_{x|\mathbf{u}}$ by $\Delta\tau_{x|\mathbf{u}}$, such that:*

$$pr(y, \mathbf{e}) - \kappa \cdot pr(\mathbf{e}) \geq \Delta\tau_{x|\mathbf{u}} \left(-\pi^{y,\mathbf{e}}_{x|\mathbf{u}} + \kappa \cdot \pi^{\mathbf{e}}_{x|\mathbf{u}} \right),$$

where $pr(\mathbf{e})$ and $pr(y, \mathbf{e})$ are the current probabilities of $\mathbf{e}$ and $y, \mathbf{e}$ respectively, and the constants $\pi^{\mathbf{e}}_{x|\mathbf{u}}$ and $\pi^{y,\mathbf{e}}_{x|\mathbf{u}}$ are defined by Equation 2.5.

Similarly, the solution to Inequality 2.2 is as follows.

[4] The closed form of $\pi^{\mathbf{e}}_{x|\mathbf{u}}$ is $Pr(\mathbf{u})(Pr(\mathbf{e} \mid x, \mathbf{u}) - Pr(\mathbf{e} \mid \bar{x}, \mathbf{u}))$, which is constant in terms of both $\theta_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}}$.

Corollary 2.2 *To satisfy Inequality 2.2, $Pr(y \mid \mathbf{e}) \leq \kappa$, we need to change $\tau_{x|\mathbf{u}}$ by $\Delta\tau_{x|\mathbf{u}}$, such that:*

$$pr(y, \mathbf{e}) - \kappa \cdot pr(\mathbf{e}) \leq \Delta\tau_{x|\mathbf{u}} \left(-\pi^{y,\mathbf{e}}_{x|\mathbf{u}} + \kappa \cdot \pi^{\mathbf{e}}_{x|\mathbf{u}} \right),$$

where $pr(\mathbf{e})$ and $pr(y, \mathbf{e})$ are the current probabilities of $\mathbf{e}$ and $y, \mathbf{e}$ respectively, and the constants $\pi^{\mathbf{e}}_{x|\mathbf{u}}$ and $\pi^{y,\mathbf{e}}_{x|\mathbf{u}}$ are defined by Equation 2.5.

To enforce Inequality 2.3, it suffices to ensure that $Pr(y, \mathbf{e}) - Pr(z, \mathbf{e}) \geq \kappa \cdot Pr(\mathbf{e})$. Therefore, we need to solve for $\Delta\tau_{x|\mathbf{u}}$ in the following inequality:

$$pr(y, \mathbf{e}) + \pi^{y,\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} - pr(z, \mathbf{e}) - \pi^{z,\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \geq \kappa \left(pr(\mathbf{e}) + \pi^{\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \right).$$

Rearranging the terms, we get the following result.

Corollary 2.3 *To satisfy Inequality 2.3, $Pr(y \mid \mathbf{e}) - Pr(z \mid \mathbf{e}) \geq \kappa$, we need to change $\tau_{x|\mathbf{u}}$ by $\Delta\tau_{x|\mathbf{u}}$, such that:*

$$pr(y, \mathbf{e}) - pr(z, \mathbf{e}) - \kappa \cdot pr(\mathbf{e}) \geq \Delta\tau_{x|\mathbf{u}} \left(-\pi^{y,\mathbf{e}}_{x|\mathbf{u}} + \pi^{z,\mathbf{e}}_{x|\mathbf{u}} + \kappa \cdot \pi^{\mathbf{e}}_{x|\mathbf{u}} \right),$$

where $pr(\mathbf{e})$, $pr(y, \mathbf{e})$, and $pr(z, \mathbf{e})$ are the current probabilities of $\mathbf{e}$, $y, \mathbf{e}$, and $z, \mathbf{e}$ respectively, and the constants $\pi^{\mathbf{e}}_{x|\mathbf{u}}$, $\pi^{y,\mathbf{e}}_{x|\mathbf{u}}$, and $\pi^{z,\mathbf{e}}_{x|\mathbf{u}}$ are defined by Equation 2.5.

To enforce Inequality 2.4, we need to solve for $\Delta\tau_{x|\mathbf{u}}$ in the following inequality:

$$pr(y, \mathbf{e}) + \pi^{y,\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \geq \kappa \left(pr(z, \mathbf{e}) + \pi^{z,\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \right).$$

Rearranging the terms, we get the following result.

Corollary 2.4 *To satisfy Inequality 2.4, $Pr(y \mid \mathbf{e}) / Pr(z \mid \mathbf{e}) \geq \kappa$, we need to change $\tau_{x|\mathbf{u}}$ by $\Delta\tau_{x|\mathbf{u}}$, such that:*

$$pr(y, \mathbf{e}) - \kappa \cdot pr(z, \mathbf{e}) \geq \Delta\tau_{x|\mathbf{u}} \left(-\pi^{y,\mathbf{e}}_{x|\mathbf{u}} + \kappa \cdot \pi^{z,\mathbf{e}}_{x|\mathbf{u}} \right),$$

where $pr(y, \mathbf{e})$ and $pr(z, \mathbf{e})$ are the current probabilities of $y, \mathbf{e}$ and $z, \mathbf{e}$ respectively, and the constants $\pi^{y,\mathbf{e}}_{x|\mathbf{u}}$ and $\pi^{z,\mathbf{e}}_{x|\mathbf{u}}$ are defined by Equation 2.5.

The solutions of $\Delta\tau_{x|\mathbf{u}}$ in Corollaries 2.1 to 2.4 are always in one of the following two forms:

- $\Delta\tau_{x|\mathbf{u}} \leq \delta$, for some computed $\delta < 0$, in which case the new value of $\tau_{x|\mathbf{u}}$ must be in the interval $[0, p+\delta]$, where p is the current value of $\tau_{x|\mathbf{u}}$;
- $\Delta\tau_{x|\mathbf{u}} \geq \delta$, for some computed $\delta > 0$, in which case the new value of $\tau_{x|\mathbf{u}}$ must be in the interval $[p+\delta, 1]$, where p is the current value of $\tau_{x|\mathbf{u}}$.

Therefore, δ is the minimum amount of change in $\tau_{x|\mathbf{u}}$ that can enforce the query constraint. For many parameters, no solutions can be found, meaning there is no way we can change these parameters to enforce the desired query constraint.

The question now is how to find the solutions efficiently for all network parameters. Note that there are usually several parameter changes that can enforce the given constraint, and we are interested in identifying all such changes. To do so, we must be able to compute a number of probabilities under the current distribution *pr*. For example, we can easily solve for the amount of change $\Delta\tau_{x|\mathbf{u}}$ in Corollary 2.1, once we know the following probabilities: $pr(\mathbf{e})$, $pr(y, \mathbf{e})$, $pr(\mathbf{e}, x, \mathbf{u})$, $pr(\mathbf{e}, \bar{x}, \mathbf{u})$, $pr(y, \mathbf{e}, x, \mathbf{u})$, and $pr(y, \mathbf{e}, \bar{x}, \mathbf{u})$. This leads to the following complexity of a procedure that finds all solutions of single parameter changes.

Corollary 2.5 *If we have an algorithm that can compute $Pr(\mathbf{i}, x, \mathbf{u})$, for a given instantiation $\mathbf{i}$, and all family instantiations $x, \mathbf{u}$ of all variables X in a Bayesian network,*[5] *in $O(f)$ time, we can solve for the solution of $\Delta\tau_{x|\mathbf{u}}$ in Corollaries 2.1 to 2.4 for all meta-parameters $\tau_{x|\mathbf{u}}$ of the Bayesian network in $O(f)$ time.*

For example, to solve for $\Delta\tau_{x|\mathbf{u}}$ in Corollary 2.3 for all meta-parameters $\tau_{x|\mathbf{u}}$ of the Bayesian network, we need to run the algorithm that computes $Pr(\mathbf{i}, x, \mathbf{u})$ three times, first with $\mathbf{i} = \mathbf{e}$, then with $\mathbf{i} = y, \mathbf{e}$, and finally with $\mathbf{i} = z, \mathbf{e}$.

[5] *The family of a variable X is the set containing X and its parents $\mathbf{U}$.*

The join-tree algorithm [JLO90, SS86] and the differential approach [Dar03] can both compute $Pr(\mathbf{i}, x, \mathbf{u})$, for a given instantiation $\mathbf{i}$ and all family instantiations $x, \mathbf{u}$ of all variables X, in $O(n2^w)$ time, where n is the number of variables in the Bayesian network and w is the tree width, which is a quantity that depends on the connectivity of the network structure [Dar03]. Note that $O(n2^w)$ time is also the complexity needed to answer one of the simplest queries in Bayesian networks, that of computing the probability of evidence $Pr(e)$.

2.1.2 Parameters of Multi-Valued Variables

Our previous results can be easily extended to a multi-valued variable X, as long as we use a scheme to change the co-varying parameters when a parameter in the CPT of X is changed [Dar03, Kv00]. After the parameter $\theta_{x|\mathbf{u}}$ is changed, we need to change the other parameters in the same conditional probability distribution, i.e., $\theta_{x^*|\mathbf{u}}$ for all $x^* \neq x$, to ensure that $\theta_{x|\mathbf{u}} + \sum_{x^* \neq x} \theta_{x^*|\mathbf{u}} = 1$. The most common way to do this is to use the *proportional scheme*. In this scheme, we change the co-varying parameters such that the ratios between them remain the same. For example, given a variable X with three values $\{1, 2, 3\}$, the current values of the three parameters in a conditional probability distribution are $\theta_{X=1|\mathbf{u}} = .6$, $\theta_{X=2|\mathbf{u}} = .3$, and $\theta_{X=3|\mathbf{u}} = .1$. After $\theta_{X=1|\mathbf{u}}$ is changed to .8, we change the other two parameter values to $\theta_{X=2|\mathbf{u}} = .3(.2/.4) = .15$ and $\theta_{X=3|\mathbf{u}} = .1(.2/.4) = .05$, such that the ratio $\theta_{X=2|\mathbf{u}}/\theta_{X=3|\mathbf{u}}$ remains 3. We now formally define the meta-parameter $\tau_{x|\mathbf{u}}$ such that all co-varying parameters can be changed simultaneously using the proportional scheme.

Definition 2.1 *After the parameter $\theta_{x|\mathbf{u}}$ is changed, we can change the co-varying parameters using the proportional scheme, by introducing the meta-parameter*

$\tau_{x|\mathbf{u}}$*, and assigning:*

$$\begin{aligned}\theta_{x|\mathbf{u}} &= \tau_{x|\mathbf{u}}; \\ \theta_{x^*|\mathbf{u}} &= \frac{\vartheta_{x^*|\mathbf{u}}}{1-\vartheta_{x|\mathbf{u}}}\Big(1-\tau_{x|\mathbf{u}}\Big) \textit{ for all } x^* \neq x,\end{aligned}$$

where $\vartheta_{x|\mathbf{u}}$ *and* $\vartheta_{x^*|\mathbf{u}}$ *are the current parameter values of* $\theta_{x|\mathbf{u}}$ *and* $\theta_{x^*|\mathbf{u}}$ *respectively.*[6]

With this definition of meta-parameter $\tau_{x|\mathbf{u}}$, $Pr(\mathbf{e})$ is again linear in $\tau_{x|\mathbf{u}}$, and the derivative of $Pr(\mathbf{e})$ with respect to $\tau_{x|\mathbf{u}}$ is given by:

$$\pi^{\mathbf{e}}_{x|\mathbf{u}} \stackrel{\text{def}}{=} \frac{\partial Pr(\mathbf{e})}{\tau_{x|\mathbf{u}}} = \frac{Pr(\mathbf{e},x,\mathbf{u})}{\theta_{x|\mathbf{u}}} - \frac{\sum_{x^*\neq x} Pr(\mathbf{e},x^*,\mathbf{u})}{\sum_{x^*\neq x}\theta_{x^*|\mathbf{u}}}. \tag{2.7}$$

Equation 2.7 is similar to Equation 2.5, if we group all values $x^* \neq x$ into the pseudo-value $\bar{x}$, and thus we can easily solve for $\Delta\tau_{x|\mathbf{u}}$ in Corollaries 2.1 to 2.4 for multi-valued variables.

There are certainly other schemes for changing co-varying parameters of multi-valued variables that experts may want to use. For example, sometimes certain parameters, such as those that are assigned 0 values, should remain constant [WD00]. This can easily be done by using another set of equations to relate the meta-parameter $\tau_{x|\mathbf{u}}$ with the parameters $\theta_{x^*|\mathbf{u}}$ for all $x^* \neq x$. As long as $Pr(\mathbf{e})$ is linear in $\tau_{x|\mathbf{u}}$, we can compute $\pi^{\mathbf{e}}_{x|\mathbf{u}}$, which is a constant in terms of $\tau_{x|\mathbf{u}}$, and all of the previous results still hold.

[6] *If* $\vartheta_{x|\mathbf{u}} = 1$*, we define the proportional scheme as assigning* $\theta_{x|\mathbf{u}} = \tau_{x|\mathbf{u}}$ *and* $\theta_{x^*|\mathbf{u}} = (1-\tau_{x|\mathbf{u}})/(c-1)$ *for all* $x^* \neq x$*, where* c *is the cardinality of* X*.*

2.2 SamIam: A Tool for Tuning Bayesian Network Parameters

To automate the process of tuning Bayesian network parameters, we have developed a program called SamIam (Sensitivity Analysis, Modeling, Inference And More) [ACC]. The procedure of finding single parameter changes that can enforce a query constraint in a Bayesian network given in Section 2.1 has been implemented in SamIam, which includes two main components: a graphical user interface and a reasoning engine. Figure 2.2 shows a screenshot of SamIam performing sensitivity analysis. We now present an example to illustrate how SamIam can be used in practice.

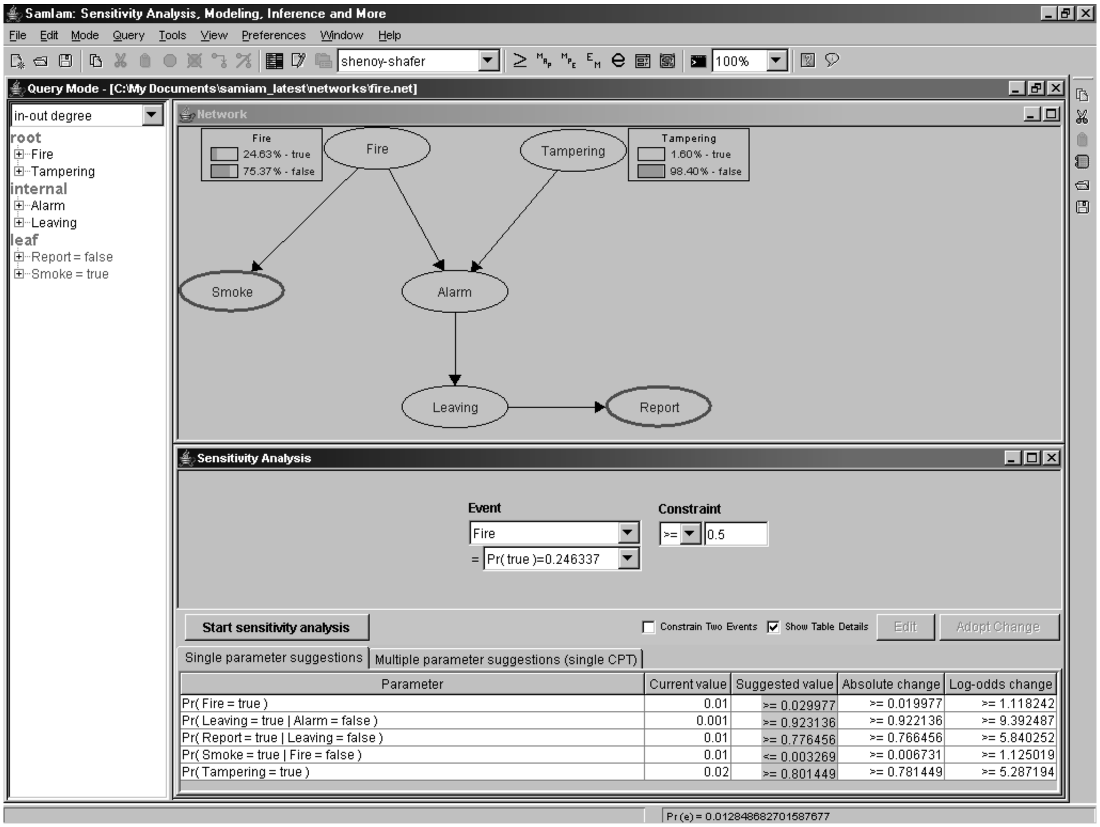
Example 2.1 *Consider again the Bayesian network* Fire. *We now set the evidence* $\mathbf{e} = \{Smoke = true, Report = false\}$*, i.e., smoke is observed, but there is no report of people evacuating the building. Current query results indicate that* $Pr(Fire = true \mid \mathbf{e}) = .246$ *and* $Pr(Tampering = true \mid \mathbf{e}) = .0160$*. However, we believe in this case that the posterior probability of having a fire should be no less than* $.5$*, and ask SamIam to find the single parameter changes that can enforce the query constraint* $Pr(Fire = true \mid \mathbf{e}) \geq .5$*. SamIam solves for the solution of Corollary 2.1 for every network parameter, and returns five suggestions of single parameter changes, as shown in Figure 2.2:*

1. *Increase* $Pr(Fire = true)$ *from* $.01$ *to* $\geq .0300$*;*

2. *Increase* $Pr(Leaving = true \mid Alarm = false)$ *from* $.001$ *to* $\geq .923$*;*

3. *Increase* $Pr(Report = true \mid Leaving = false)$ *from* $.01$ *to* $\geq .776$*;*

4. *Decrease* $Pr(Smoke = true \mid Fire = false)$ *from* $.01$ *to* $\leq .00327$*;*


SamIam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
shenoy-shafer
100%
Query Mode - [C:\My Documents\samiam_latest\networks\fire.net]
in-out degree
root
Fire
Tampering
internal
Alarm
Leaving
leaf
Report = false
Smoke = true
Network
Fire
24.63% - true
75.37% - false
Tampering
1.60% - true
98.40% - false
Fire
Tampering
Smoke
Alarm
Leaving
Report
Sensitivity Analysis
Start sensitivity analysis
Edit
Adopt Change
Single parameter suggestions
Multiple parameter suggestions (single CPT)

Event: Fire = Pr(true)=0.246337

Constraint: >= 0.5

☐ Constrain Two Events ☑ Show Table Details

Parameter	Current value	Suggested value	Absolute change	Log-odds change
Pr(Fire = true)	0.01	>= 0.029977	>= 0.019977	>= 1.118242
Pr(Leaving = true | Alarm = false)	0.001	>= 0.923136	>= 0.922136	>= 9.392487
Pr(Report = true | Leaving = false)	0.01	>= 0.776456	>= 0.766456	>= 5.840252
Pr(Smoke = true | Fire = false)	0.01	<= 0.003269	>= 0.006731	>= 1.125019
Pr(Tampering = true)	0.02	>= 0.801449	>= 0.781449	>= 5.287194

Pr (e) = 0.012848682701587677

Figure 2.2: A screenshot of SamIam returning suggestions of single parameter changes for enforcing a query constraint.

5. *Increase Pr(Tampering = true) from .02 to ≥ .801.*

Clearly, three of these parameter changes can be ruled out based on qualitative considerations, and the only sensible parameter changes are either to increase the prior probability of having a fire, or to decrease the probability of observing smoke without having a fire.

This example and other similar ones suggest that finding such parameter changes and their magnitudes is inevitable for developing a faithful Bayesian network, yet it is not trivial for experts to accomplish this task by visual inspection of the Bayesian network, often due to its size and complexity. Sensitivity analysis tools such as SamIam can help facilitate this process by identifying important parameters that need to be fine-tuned in order to satisfy certain query constraints. Of course, if we are given multiple query constraints, we need to be cautious when implementing a suggestion made by SamIam to satisfy one of the query constraints, since applying this change may result in violating other query constraints. In this case, the parameter changes suggested by SamIam should be used to help experts focus their attention on the relevant parameters.

Moreover, the examples we have shown illustrate the need to develop more analytic tools to understand and explain the sensitivity of queries to certain parameter changes. There is also a need to reconcile the sensitivity of parameters exhibited by the examples with previous experimental studies demonstrating the robustness of probabilistic queries against small parameter changes in certain application areas, such as diagnosis [PHP96]. We will address these questions in Chapter 3.

CHAPTER 3

Network-Independent Sensitivity Analysis

In Chapter 2, we focused our study on the relationships between parameter changes and query changes given the specific details of the Bayesian network. In this chapter, we will instead perform a network-independent analysis of the relationships between parameter changes and query changes, which can be done without being given any information about the Bayesian network [CD01, CD02b].

One of the central issues for sensitivity analysis of Bayesian networks is the assessment of the impact of parameter changes on probabilistic queries of interest. Common wisdom has it that small perturbations in the parameter values of a Bayesian network do not matter much for the results of probabilistic queries. However, we have encountered scenarios in which queries are quite sensitive to small variations in certain network parameters, i.e., small variations in these network parameters can lead to significant changes in computed queries. For example, the absolute change in a query can be much larger than the absolute change in a network parameter. Moreover, we will show later an example where an infinitesimal change in a network parameter leads to a change of .5 to a query. We will also show examples in which the relative change in a query is larger than the relative change in a network parameter. One may wonder then whether there is a different method of measuring probabilistic change, other than absolute or relative, which gives a non-trivial bound on the change in a probabilistic query in terms of the change in a network parameter.

To answer this and other related questions, we will conduct an analytic study of the *derivative of a probabilistic query* $Pr(y \mid \mathbf{e})$ *with respect to a meta-parameter* $\tau_{x|\mathbf{u}}$, which was introduced in Chapter 2 and enables us to change co-varying parameters simultaneously. Our study leads us to three main results:

1. A bound on the partial derivative $\partial Pr(y \mid \mathbf{e})/\partial \tau_{x|\mathbf{u}}$ in terms of $Pr(y \mid \mathbf{e})$ and $Pr(x \mid \mathbf{u})$ only, which is independent of any other aspect of the given Bayesian network;

2. A bound on the sensitivity of a query to an *infinitesimal change* in a Bayesian network parameter;

3. A bound on the sensitivity of a query to an *arbitrary change* in a Bayesian network parameter.

The last bound in particular shows that the amount of change in a probabilistic query can be bounded in terms of the amount of change in a network parameter, as long as the amount of change is measured in the *relative change in odds*. This result has a number of practical implications. First, it can relieve experts from having to be too precise when specifying certain parameters subjectively. Next, it can be important for approximate inference algorithms that pre-process network parameters to eliminate small distinctions between a set of parameters in order to increase the efficiency of inference [Poo98]. Finally, it can be used to show that automated reasoning systems based on Bayesian networks are robust, and hence, suitable for real-world applications [PHP96]. We will then explore the implications of the above bounds, where we provide an analytic explanation of why certain parameter changes do not matter.

3.1 Bounding the Derivative of a Query With Respect To a Parameter

Our starting point in understanding the sensitivity of a query $Pr(y \mid \mathbf{e})$ to changes in a parameter is to analyze the derivative of the query with respect to the parameter. Because we cannot change only one parameter without changing the other parameters in the same conditional probability distribution, we again have to use the meta-parameter $\tau_{x|\mathbf{u}}$, which was introduced in the last chapter to simultaneously change all co-varying parameters. If X is a binary variable, we assign $\theta_{x|\mathbf{u}} = \tau_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}} = 1 - \tau_{x|\mathbf{u}}$. If X is a multi-valued variable, we assume that we use the proportional scheme to change co-varying parameters, as defined in Definition 2.1.

We are now ready to tackle the problem by analyzing the partial derivative $\partial Pr(y \mid \mathbf{e})/\partial\tau_{x|\mathbf{u}}$. The following theorem provides a simple bound on this partial derivative, in terms of only $Pr(y \mid \mathbf{e})$ and $Pr(x \mid \mathbf{u}) = \tau_{x|\mathbf{u}}$, the current values of the query and the meta-parameter. We can then use this bound to study the effect of parameter changes on probabilistic queries.

Theorem 3.1 *The derivative of $Pr(y \mid \mathbf{e})$ with respect to $\tau_{x|\mathbf{u}}$ is bounded by:*[1]

$$\left|\frac{\partial Pr(y \mid \mathbf{e})}{\partial \tau_{x|\mathbf{u}}}\right| \leq \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})\,(1 - Pr(x \mid \mathbf{u}))}.$$

The bound given in Theorem 3.1 is tight, and we will show later an example where this bound is exactly assumed. The main point to note about this bound is that it is independent of the specific details of the given Bayesian network. Note that we have an exact closed form of the partial derivative $\partial Pr(y \mid \mathbf{e})/\partial\tau_{x|\mathbf{u}}$

[1] *This theorem and all results that follow in this chapter requires that $\tau_{x|\mathbf{u}} \neq 0$ and $\tau_{x|\mathbf{u}} \neq 1$, since we can only use the expression in Equation 2.5 under these conditions.*

[Dar03, GGS97], but that form includes terms that are specific to the given Bayesian network.

Figure 3.1 plots the upper bound given in Theorem 3.1 against $Pr(x \mid \mathbf{u})$ and $Pr(y \mid \mathbf{e})$. We can make two observations about this plot. For extreme values of $Pr(x \mid \mathbf{u})$, the bound approaches infinity, and thus a small absolute change in the meta-parameter $\tau_{x|\mathbf{u}}$ can have a big impact on the query $Pr(y \mid \mathbf{e})$. On the other hand, the bound approaches 0 for extreme values of the query $Pr(y \mid \mathbf{e})$. Therefore, a small absolute change in the meta-parameter $\tau_{x|\mathbf{u}}$ can only have a small effect on the absolute change in the query $Pr(y \mid \mathbf{e})$

One of the implications of Theorem 3.1 is that if we have a Bayesian network where the queries of interest have extreme values, such queries are robust against small absolute changes in the network parameters. This of course assumes that robustness is understood to be a small absolute change in the value of the given query. Interestingly enough, if y is a disease that is diagnosed by finding $\mathbf{e}$, i.e., the probability $Pr(y \mid \mathbf{e})$ is quite high, it is not surprising that this query would be robust against small perturbations to network parameters. This seems to explain some of the previous results where robustness have been confirmed for queries where $Pr(y \mid \mathbf{e}) \geq .9$ [PHP96].

Another implication of Theorem 3.1 is that we have to be careful when changing parameters that are extreme. Such parameters are potentially very influential and we must handle them with care.

Therefore, the worst situation from a robustness viewpoint materializes if we have to deal with extreme parameters and non-extreme queries. In such a case, the queries can be very sensitive to small variations in the parameters. We illustrate this by the following example.

Example 3.1 *Consider a Bayesian network with three binary variables, X, Y,*

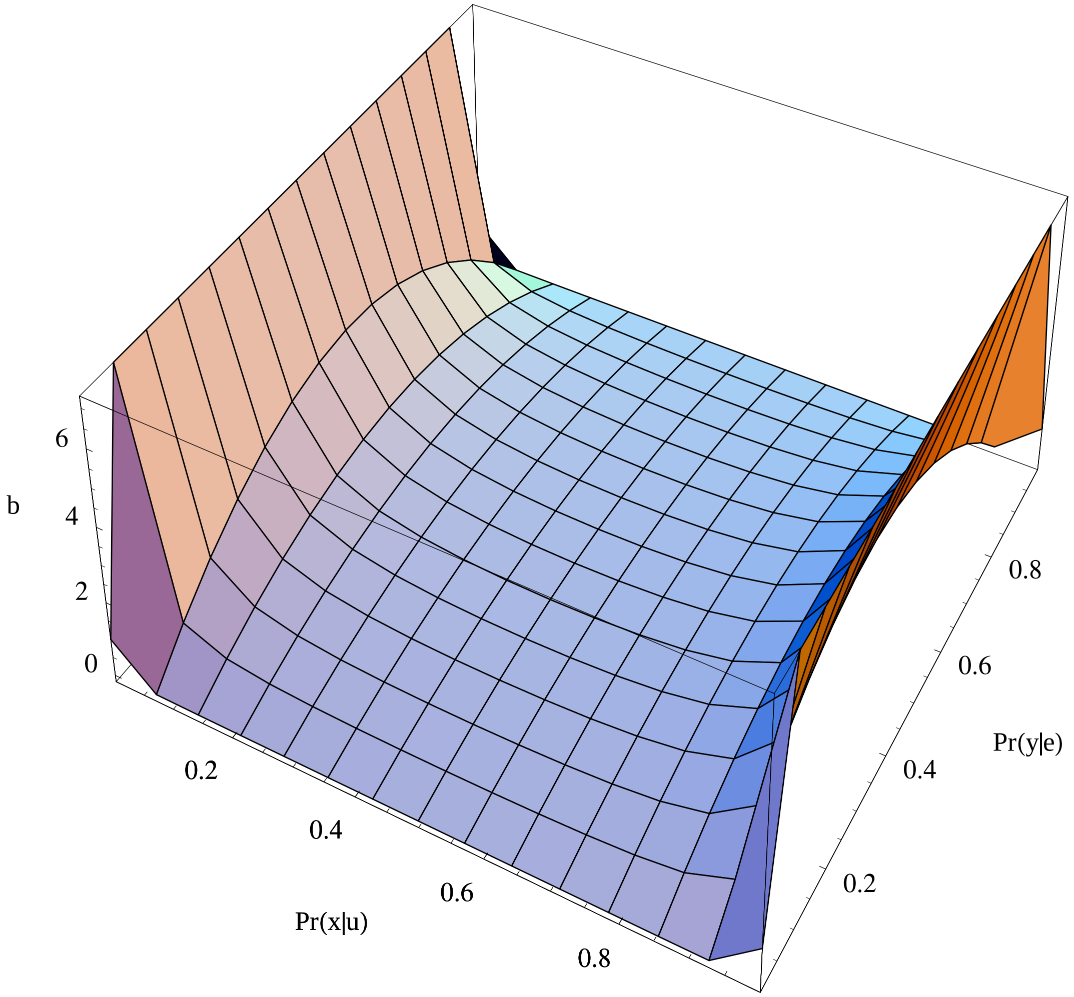


Figure 3.1: The plot of the upper bound on the partial derivative, $b = \partial Pr(y \mid \mathbf{e})/\partial \tau_{x|\mathbf{u}}$, against $Pr(x \mid \mathbf{u})$ and $Pr(y \mid \mathbf{e})$, as given in Theorem 3.1.

and E. There are two binary root nodes, X and Y, with respective parameters $\{\theta_x, \theta_{\bar{x}}\}$ and $\{\theta_y, \theta_{\bar{y}}\}$ in their CPTs. The node E is a deterministic binary node whose parents are X and Y, where $E = e$ iff $X = Y$. This dictates the following parameters in the CPT of E: $Pr(e \mid x, y) = 1$, $Pr(e \mid x, \bar{y}) = 0$ and $Pr(e \mid \bar{x}, y) = 0$, $Pr(e \mid \bar{x}, \bar{y}) = 1$. The conditional probability $Pr(y \mid e)$ can be expressed using the parameters of X and Y:

$$Pr(y \mid e) = \frac{\theta_x \theta_y}{\theta_x \theta_y + \theta_{\bar{x}} \theta_{\bar{y}}}.$$

Since $\partial\theta_x / \partial\tau_x = 1$ and $\partial\theta_{\bar{x}} / \partial\tau_x = -1$, the derivative of $Pr(y \mid e)$ with respect to τ_x is given by:

$$\begin{aligned} \frac{\partial Pr(y \mid e)}{\partial \tau_x} &= \frac{(\theta_x \theta_y + \theta_{\bar{x}} \theta_{\bar{y}})\, \theta_y - \theta_x \theta_y\, (\theta_y - \theta_{\bar{y}})}{(\theta_x \theta_y + \theta_{\bar{x}} \theta_{\bar{y}})^2} \\ &= \frac{\theta_y \theta_{\bar{y}}}{(\theta_x \theta_y + \theta_{\bar{x}} \theta_{\bar{y}})^2}. \end{aligned}$$

This is equal to the upper bound given in Theorem 3.1:

$$\begin{aligned} \frac{Pr(y \mid e)\,(1 - Pr(y \mid e))}{Pr(x)\,(1 - Pr(x))} &= \frac{(\theta_x \theta_y)\,(\theta_{\bar{x}} \theta_{\bar{y}})}{\theta_x \theta_{\bar{x}}\,(\theta_x \theta_y + \theta_{\bar{x}} \theta_{\bar{y}})^2} \\ &= \frac{\theta_y \theta_{\bar{y}}}{(\theta_x \theta_y + \theta_{\bar{x}} \theta_{\bar{y}})^2}. \end{aligned}$$

If we now set $\theta_x = \theta_{\bar{y}}$, the derivative becomes:

$$\frac{\partial Pr(y \mid e)}{\partial \tau_x} = \frac{1}{4\theta_x \theta_{\bar{x}}},$$

and as θ_x or $\theta_{\bar{x}}$ approaches 0, the derivative approaches infinity. Finally, if we set $\theta_x = \theta_{\bar{y}} = \epsilon$, where ϵ is very close to 0, we have $Pr(y \mid e) = .5$. However, if we change τ_x from ϵ to 0, but keep θ_y and $\theta_{\bar{y}}$ constant at $1 - \epsilon$ and ϵ, we get the new result $Pr(y \mid e) = 0$.

Example 3.1 illustrates three points. First, it shows that the bound given in Theorem 3.1 is tight, i.e., we can construct a Bayesian network that exactly assumes the bound. Second, it gives an example network where the partial derivative $\partial Pr(y \mid \mathbf{e})/\partial \tau_{x|\mathbf{u}}$ tends to infinity, and thus it cannot be bounded by any constant. Third, it shows that applying an infinitesimal absolute change to a meta-parameter, as we did by changing τ_x from ϵ to 0, can induce a non-infinitesimal absolute change in some query, as $Pr(y \mid e)$ changes from .5 to 0.

3.2 Bounding Query Changes Due To Parameter Changes

In this section, we will use the bound on the partial derivative $\partial Pr(y \mid \mathbf{e})/\partial \tau_{x|\mathbf{u}}$, given in Theorem 3.1, to bound the changes in a query due to changes in a meta-parameter. We first consider the case where we apply an infinitesimal change to a meta-parameter. Example 3.1 shows that an infinitesimal absolute change in a meta-parameter can induce a non-infinitesimal absolute change in some query. The following theorem, however, shows that this is not possible if we consider a relative notion of change.

Theorem 3.2 *Given meta-parameter $\tau_{x|\mathbf{u}} \leq .5$,*[2] *let $\Delta\tau_{x|\mathbf{u}}$ be an infinitesimal change applied to $\tau_{x|\mathbf{u}}$, leading to a change of $\Delta Pr(y \mid \mathbf{e})$ to the query $Pr(y \mid \mathbf{e})$. The relative change in the query $Pr(y \mid \mathbf{e})$ is bounded by:*

$$\left|\frac{\Delta Pr(y \mid \mathbf{e})}{Pr(y \mid \mathbf{e})}\right| \leq 2\left|\frac{\Delta\tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}}\right|.$$

For a function $f(p)$, the quantity:

$$\lim_{p \to p_0} \frac{\frac{f(p)-f(p_0)}{f(p_0)}}{\frac{p-p_0}{p_0}},$$

[2] *For a binary variable X, if $\tau_{x|\mathbf{u}} > .5$, we can instead choose the meta-parameter $\tau_{\bar{x}|\mathbf{u}}$ without loss of generality.*

is typically known as the *sensitivity* of f to p at $p = p_0$. Therefore, Theorem 3.2 shows that the sensitivity of $Pr(y \mid \mathbf{e})$ to $\tau_{x|\mathbf{u}}$ is bounded.

To illustrate Theorem 3.2, consider again Example 3.1. The change of τ_x from ϵ to 0 amounts to a relative change of $|-\epsilon/\epsilon| = 1$. The change of $Pr(y \mid e)$ from .5 to 0 amounts to a relative change of $|-.5/.5| = 1$. Therefore, the relative change in the query is not as great from this viewpoint.[3]

The relative change in $Pr(y \mid \mathbf{e})$ may be greater than double the relative change in $\tau_{x|\mathbf{u}}$ for non-infinitesimal changes, since the partial derivative $\partial Pr(y \mid \mathbf{e})/\partial \tau_{x|\mathbf{u}}$ depends on the value of $\tau_{x|\mathbf{u}}$ [Dar03, Jen99]. Consider again the Bayesian network given in Example 3.1. If we set $\theta_x = .5$ and $\theta_y = .01$, we obtain the result $Pr(y \mid e) = .01$. If we now increase τ_x to .6, amounting to a relative change of 20%, we get the new result $Pr(y \mid e) = .0149$, amounting to a relative change of 49%, which is more than double of the relative change in τ_x.

The question now becomes: if we change a meta-parameter $\tau_{x|\mathbf{u}}$ by an arbitrary (non-infinitesimal) amount, what can we say about the corresponding change in the query $Pr(y \mid \mathbf{e})$? Before giving the answer, we first define the notion of *odds*.

Definition 3.1 *The odds of conditional event $\alpha \mid \beta$ under distribution Pr is given by:*[4]

$$O(\alpha \mid \beta) \overset{\text{def}}{=} \frac{Pr(\alpha \mid \beta)}{Pr(\bar{\alpha} \mid \beta)} = \frac{Pr(\alpha \mid \beta)}{1 - Pr(\alpha \mid \beta)}.$$

We now present the following theorem which bounds the amount of query change due to an arbitrary parameter change.

[3] If we consider the meta-parameter $\tau_{\bar{x}} = 1-\epsilon$ instead, the relative change in $\tau_{\bar{x}}$ then amounts to $\epsilon/(1-\epsilon)$. However, Theorem 3.2 is not applicable in this case with ϵ close to 0, since the theorem requires that the chosen meta-parameter to be no greater than .5.

[4] *Of course, we must have $Pr(\beta) \neq 0$ for the odds to be defined.*

Theorem 3.3 *Assume that we apply an arbitrary change to the meta-parameter* $\tau_{x|\mathbf{u}}$*. The initial odds of* $x \mid \mathbf{u}$ *and* $y \mid \mathbf{e}$ *are denoted by* $O(x \mid \mathbf{u})$ *and* $O(y \mid \mathbf{e})$ *respectively, and the new odds of* $x \mid \mathbf{u}$ *and* $y \mid \mathbf{e}$ *after the parameter change are denoted by* $O'(x \mid \mathbf{u})$ *and* $O'(y \mid \mathbf{e})$ *respectively. If the change in* $\tau_{x|\mathbf{u}}$ *is positive, we have:*

$$\frac{O(x \mid \mathbf{u})}{O'(x \mid \mathbf{u})} \leq \frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \leq \frac{O'(x \mid \mathbf{u})}{O(x \mid \mathbf{u})},$$

or if it is negative, we have:

$$\frac{O'(x \mid \mathbf{u})}{O(x \mid \mathbf{u})} \leq \frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \leq \frac{O(x \mid \mathbf{u})}{O'(x \mid \mathbf{u})}.$$

Combining both results, we have:

$$\left|\ln O'(y \mid \mathbf{e}) - \ln O(y \mid \mathbf{e})\right| \leq \left|\ln O'(x \mid \mathbf{u}) - \ln O(x \mid \mathbf{u})\right|.$$

Theorem 3.3 states that the relative change in the odds of $y \mid \mathbf{e}$ is bounded by the relative change in the odds of $x \mid \mathbf{u}$, or equivalently, the log-odds change in any query is bounded by the log-odds change in any parameter. Note that *this result makes no assumptions whatsoever about the structure or the CPTs of the given Bayesian network.* We now illustrate this using an example.

Example 3.2 *Consider again Example 2.1. Given evidence* $\mathbf{e} = \{\mathit{Smoke} = \mathit{true}, \mathit{Report} = \mathit{false}\}$*, we decide to increase the posterior probability* $Pr(\mathit{Fire} = \mathit{true} \mid \mathbf{e})$ *from* $.246$ *to* $.5$*. The log-odds change in this query is* $|\ln O'(\mathit{Fire} = \mathit{true} \mid \mathbf{e}) - \ln O(\mathit{Fire} = \mathit{true} \mid \mathbf{e})| = 1.12$*. There were five single parameter changes suggested by SamIam, as shown in Figure 2.2, and for the minimum amount of each of these parameter changes, we calculate the log-odds change,* $|\Delta \ln O(x \mid \mathbf{u})| = |\ln O'(x \mid \mathbf{u}) - \ln O(x \mid \mathbf{u})|$*:*

1. *Increase* $Pr(\mathit{Fire} = \mathit{true})$ *from* $.01$ *to* $.0300$*:* $|\Delta \ln O(x \mid \mathbf{u})| = 1.12$*;*

2. *Increase* $Pr(\mathit{Leaving} = \mathit{true} \mid \mathit{Alarm} = \mathit{false})$ *from* $.01$ *to* $.923$*:* $|\Delta \ln O(x \mid \mathbf{u})| = 9.39$*;*

3. *Increase* $Pr(\mathit{Report} = \mathit{true} \mid \mathit{Leaving} = \mathit{false})$ *from* $.01$ *to* $.776$*:* $|\Delta \ln O(x \mid \mathbf{u})| = 5.84$*;*

4. *Decrease* $Pr(\mathit{Smoke} = \mathit{true} \mid \mathit{Fire} = \mathit{false})$ *from* $.01$ *to* $.00327$*:* $|\Delta \ln O(x \mid \mathbf{u})| = 1.13$*;*

5. *Increase* $Pr(\mathit{Tampering} = \mathit{true})$ *from* $.02$ *to* $.801$*:* $|\Delta \ln O(x \mid \mathbf{u})| = 5.29$*.*

Therefore, we can see that all suggested parameter changes satisfy Theorem 3.3, i.e., the log-odds change in the query value is bounded by the log-odds change in the parameter value.

An interesting special case of Theorem 3.3 is when X is a root node and $X = Y$. From basic probability theory, we have:

$$O(x \mid \mathbf{e}) = O(x)\frac{Pr(\mathbf{e} \mid x)}{Pr(\mathbf{e} \mid \bar{x})}.$$

As the ratio $Pr(\mathbf{e} \mid x)/Pr(\mathbf{e} \mid \bar{x})$ is independent of $Pr(x)$, the ratio $O(x \mid \mathbf{e})/O(x)$ is also independent of this prior probability. Therefore, we can conclude that:

$$\frac{O'(x \mid \mathbf{e})}{O(x \mid \mathbf{e})} = \frac{O'(x)}{O(x)}. \tag{3.1}$$

This means we can find the exact amount of change needed for a meta-parameter τ_x in order to induce a particular change in the query $Pr(x \mid \mathbf{e})$. There is no need to use the more expensive procedure given in Section 2.1 in this case. For example, consider again Example 3.2. We can easily compute the new prior probability $Pr'(\mathit{Fire} = \mathit{true})$ that enforces the desired constraint using Equation 3.1:

$$\frac{\frac{.5}{.5}}{\frac{.246}{.754}} = \frac{\frac{Pr'(\mathit{Fire}=\mathit{true})}{1-Pr'(\mathit{Fire}=\mathit{true})}}{\frac{.01}{.99}},$$

giving us $Pr'(\mathit{Fire} = \mathit{true}) = .0300$, which is equal to the result we obtained using SamIam. Both the changes in $Pr(\mathit{Fire} = \mathit{true})$ and $Pr(\mathit{Fire} = \mathit{true} \mid \mathbf{e})$ amount to a log-odds change of 1.12.

Theorem 3.3 has a number of implications. First, given a particular query $Pr(y \mid \mathbf{e})$ and a meta-parameter $\tau_{x|\mathbf{u}}$, it can be used to bound the effect that a change in $\tau_{x|\mathbf{u}}$ will have on the query $Pr(y \mid \mathbf{e})$. Suppose we follow the first suggestion by SamIam in Example 3.2, and increase $Pr(\mathit{Fire} = \mathit{true})$ from .01 to .0300. Consequently, the log-odds change in any query will be bounded by the log-odds change in this network parameter. For example, currently we have the query value $Pr(\mathit{Tampering} = \mathit{true} \mid \mathbf{e}) = .016$. We can find the range of the new query value $Pr'(\mathit{Tampering} = \mathit{true} \mid \mathbf{e})$ using Theorem 3.3:

$$\left| \ln \frac{Pr'(\mathit{Tampering} = \mathit{true} \mid \mathbf{e})}{1 - Pr'(\mathit{Tampering} = \mathit{true} \mid \mathbf{e})} - \ln \frac{.016}{.984} \right| \leq \left| \ln \frac{.0300}{.9700} - \ln \frac{.01}{.99} \right|,$$

giving us the bound $.00528 \leq Pr'(\mathit{Tampering} = \mathit{true} \mid \mathbf{e}) \leq .0474$. The exact new query value $Pr'(\mathit{Tampering} = \mathit{true} \mid \mathbf{e})$, obtained by inference, is .0233, which is within the computed bounds.

Second, Theorem 3.3 can be used to efficiently approximate the solutions given in Corollaries 2.1 to 2.4, which compute single parameter changes for enforcing query constraints. Given a desirable change in the value of query $Pr(y \mid \mathbf{e})$, we can use Theorem 3.3 to immediately compute a lower bound on the minimum change in the meta-parameter $\tau_{x|\mathbf{u}}$ needed to induce the query change. This method can be applied in constant time and can serve as a preliminary suggestion, as the procedure given in Section 2.1 is much more expensive computationally.

Third, when SamIam suggests several parameter changes where each one of them can induce a desirable change on a given query, we may want to ask: which one of these changes should we adopt? The main principle applied in these situations is to adopt a "minimal" change. However, what is "minimal" in this case? As previous examples have revealed, a notion of minimality that is based on the amount of absolute change or relative change can be very misleading. Instead, according to Theorem 3.3, we should adopt the parameter change that minimizes the relative change in odds, as other queries can be shown to be robust against such a change in a precise sense.

For example, we are given two parameter changes, one from .1 to .15, and another from .4 to .45. Both these changes amount to the same absolute change of .05. However, the first change amounts to a log-odds change of .462, while the second change amounts to a log-odds change of .205. Therefore, two parameter changes that amount to the same absolute change can have different amounts of log-odds change.

On the other hand, two parameter changes that amount to the same relative change can also have different amounts of log-odds change. For example, we are given two parameter changes, one from .1 to .2, and another from .2 to .4. Both these changes double the initial parameter value. However, the first change amounts to a log-odds change of .811, while the second change amounts to a log-odds change of .981.

Finally, the results in this section can be used to obtain a better intuitive understanding of parameter changes that do or do not matter, a topic that we will discuss in the next section.

3.3 Parameter Changes That (Don't) Matter

We now return to a central question: when do changes in network parameters matter (or not matter)? As we mentioned earlier, there have been experimental studies investigating the robustness of Bayesian networks against parameter changes [PHP96]. However, we have also shown very simple and intuitive examples where networks can be very sensitive to small parameter changes. This calls for a better understanding of the effect of parameter changes on queries, so we can intuitively sort out situations in which such changes do or do not matter. Our goal in this section is to further develop such an understanding by looking more closely into some of the implications of Theorem 3.3. We start by highlighting the difference between this theorem and previous results on sensitivity analysis.

3.3.1 Network-Specific Sensitivity Analysis

One of the main differences between our results and other approaches to sensitivity analysis is that we do not need to know the Bayesian network, and hence, do not need to perform inference. To clarify this difference, we now look at the sensitivity function approach [vR01], which computes the *sensitivity function* that relates a query, $f(p) = Pr(y \mid \mathbf{e})$, and a meta-parameter, $p = \tau_{x|\mathbf{u}}$, in the form:

$$f(p) = \frac{\mu_1 p + \mu_0}{\nu_1 p + \nu_0}, \tag{3.2}$$

where μ_1, μ_0, ν_1, ν_0 are constants that depend on the given network and are computed by performing inference [vR01].[5]

As an example, consider again the Bayesian network Fire. We can express the query $Pr(\mathit{Fire} = \mathit{true} \mid \mathit{Smoke} = \mathit{true}, \mathit{Report} = \mathit{false})$ as a function of the parameter $p = Pr(\mathit{Smoke} = \mathit{true} \mid \mathit{Fire} = \mathit{false})$ using Equation 3.2. The sensitivity function is given by:

$$f(p) = \frac{.00317}{.968 \cdot p + .00317},$$

and we plot this function in Figure 3.2. We can see that at the current parameter value .01, the query value is .246, but if we decrease the parameter value to

[5] It can be easily verified that $\mu_1 = \pi^{y,\mathbf{e}}_{x|\mathbf{u}}$ and $\nu_1 = \pi^{\mathbf{e}}_{x|\mathbf{u}}$, as defined in Theorem 2.1.

.00327, the query value increases to .5, which is one of the parameter changes suggested by SamIam in Example 2.1.

However, from our work in Section 3.2, we can find a bound on the relationship between a query and a network parameter without doing inference on the Bayesian network (and without knowing the network). This is because Theorem 3.1 gives us the bound on the derivative of $f(p)$ with respect to p, and Theorem 3.3 gives us the bound on the change in $f(p)$ after an arbitrary change in p. For example, by changing the current parameter value from .01 to .00327, the new query value will be within the bounds of .0958 and .501. On the other hand, if we want the query value to increase from .246 to .5, we will have to decrease the parameter value from .01 to .00328 or less, or increase it to .0300 or more.

3.3.2 Assuring Query Robustness

One of the important issues we have yet to settle is: what does it mean for a parameter change to *not* matter? We can think of at least three definitions. First, the absolute change in the probability value $Pr(y \mid \mathbf{e})$ is small. Second, the relative change in the probability value $Pr(y \mid \mathbf{e})$ is small. Third, the relative change in the odds value $O(y \mid \mathbf{e})$ is small. The first notion is the most prevalent one in the literature, so we will adopt it in the rest of this section.

Suppose we have a Bayesian network for a diagnostic application and we are concerned about the robustness of the query $Pr(y \mid \mathbf{e})$ with respect to changes in network parameters. In this application, y is a particular disease and $\mathbf{e}$ is a set of observations that predicts the disease, where the current value of the query is $Pr(y \mid \mathbf{e}) = .9$. Let us define robustness in this case to be an absolute change of no more than .05 in the query value. We would like to know: what amount

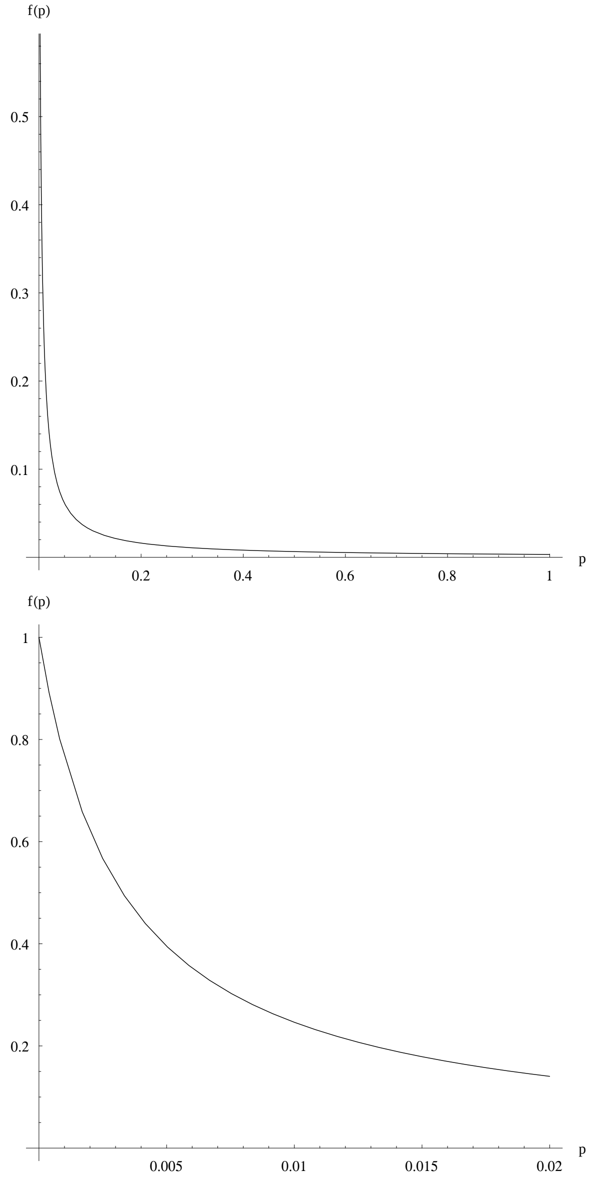


Figure 3.2: The plots of the sensitivity function $f(p) = Pr(\mathit{Fire} = \mathit{true} \mid \mathit{Smoke} = \mathit{true}, \mathit{Report} = \mathit{false})$ against the parameter value $p = Pr(\mathit{Smoke} = \mathit{true} \mid \mathit{Fire} = \mathit{false})$, in the Bayesian network Fire. The second plot magnifies the first plot for the region where p is between 0 and .02.

of change in a single network parameter is guaranteed to keep the query within the desirable range? We can use Theorem 3.3 easily to answer this question. If the current value of a parameter is p, and we are going to change its value to $p' = p + \delta$, and we want the value of the query to remain $\leq .95$, we must ensure that:

$$\left| \ln \frac{p+\delta}{1-(p+\delta)} - \ln \frac{p}{1-p} \right| \leq \left| \ln \frac{.95}{.05} - \ln \frac{.9}{.1} \right| = .747.$$

Similarly, if we want to ensure that the query remains $\geq .85$, we must ensure that:

$$\left| \ln \frac{p+\delta}{1-(p+\delta)} - \ln \frac{p}{1-p} \right| \leq \left| \ln \frac{.85}{.15} - \ln \frac{.9}{.1} \right| = .463.$$

Figure 3.3 plots δ as a function of p. The main observation we can make here is that the amount of permissible parameter change depends on the current parameter value p, with smaller absolute changes allowed for extreme values of p. It is also interesting to note that it is easier to guarantee that the query remains no more than .95 than to guarantee that it remains no less than .85. In general, it is more likely for a parameter change to reduce the value of a query that is close to 1 than to increase it by the same amount (and to increase the value of a query that is close to 0 than to reduce it by the same amount). Finally, if we are going to increase the parameter, a parameter value close to .4 allows the biggest absolute change. However, if we are going to decrease the parameter, a parameter value close to .6 allows the biggest absolute change.

Let us now repeat the same exercise but given that the current value of the query is $Pr(y \mid \mathbf{e}) = .6$, yet insisting on the same measure of robustness. We must now ensure that:

$$\left| \ln \frac{p+\delta}{1-(p+\delta)} - \ln \frac{p}{1-p} \right| \leq \left| \ln \frac{.65}{.35} - \ln \frac{.6}{.4} \right| = .214,$$

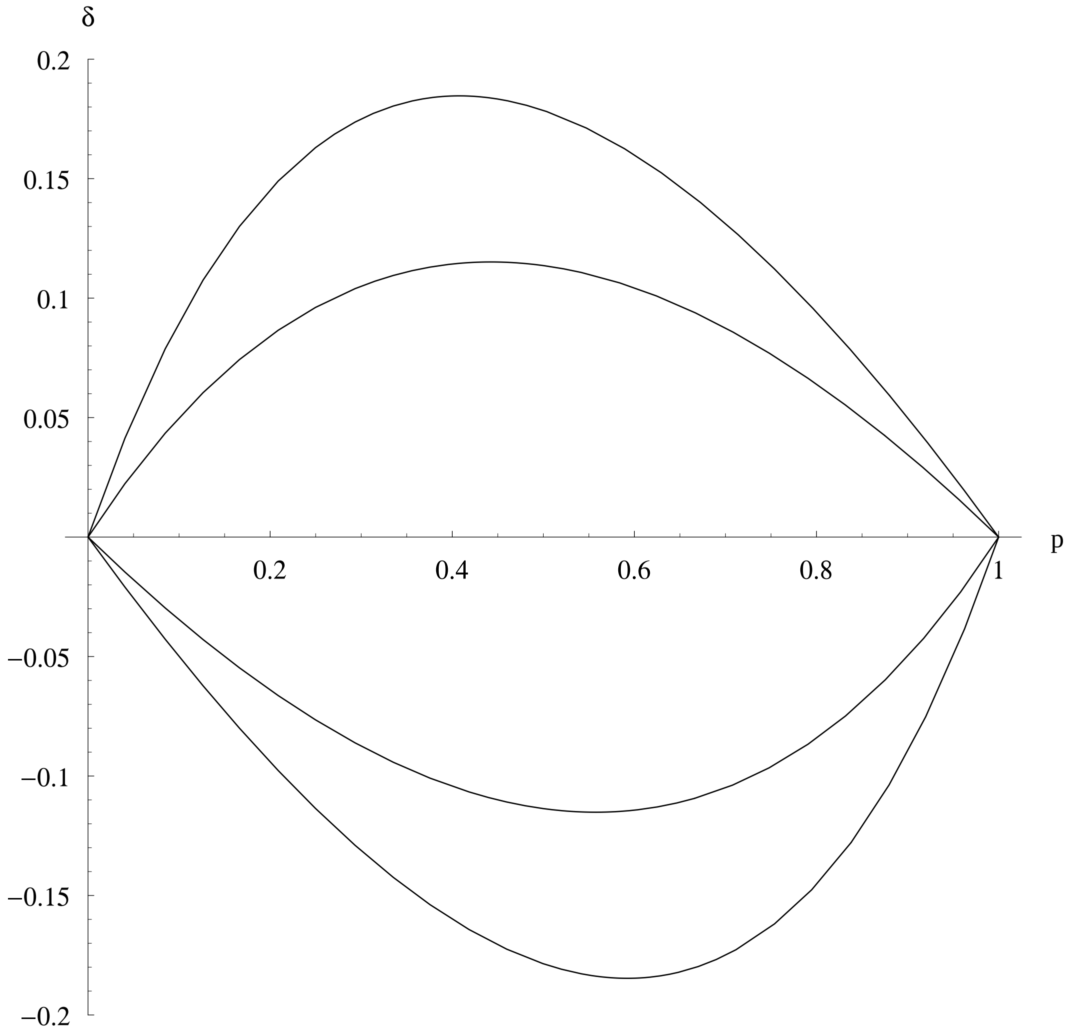


Figure 3.3: The plot of δ against p that would guarantee the query $Pr(y \mid \mathbf{e}) = .9$ to stay within the interval $[.85, .95]$, where p and $p' = p + \delta$ are the initial and new parameter values respectively. The outer envelope guarantees the query to remain no more than .95, while the inner envelope guarantees the query to remain no less than .85.

and:

$$\left|\ln \frac{p+\delta}{1-(p+\delta)} - \ln \frac{p}{1-p}\right| \leq \left|\ln \frac{.65}{.35} - \ln \frac{.6}{.4}\right| = .205.$$

Figure 3.4 plots δ as a function of p, and again, we observe that smaller absolute changes are allowed for extreme values of p. Another observation we emphasize here is that for the same parameter value p, the amount of permissible parameter change is now much smaller compared with the previous case, since the current value of the query is not as extreme. Therefore, this query is much less robust than the previous one.

Finally, Figure 3.5 plots the log-odds change, $|\Delta \ln O(\alpha \mid \beta)| = |\ln O'(\alpha \mid \beta) - \ln O(\alpha \mid \beta)|$, in the query value of an arbitrary conditional event $\alpha \mid \beta$, against its initial and new probability values, $p = Pr(\alpha \mid \beta)$ and $p' = p + \delta = Pr'(\alpha \mid \beta)$, and Figure 3.6 shows cross-sections of Figure 3.5 for three different values of p. These plots again explain analytically why for the same log-odds change, we can apply bigger absolute changes to non-extreme probabilities [Poo98, PHP96].

From Figure 3.6, we also notice that although the plot is symmetric for $p = .5$, it is not for either $p = .1$ or $p = .9$, i.e., the new parameter values of $p' = p+\delta$ and $p' = p - \delta$ give us different amounts of log-odds change. For example, changing the probability value from .1 to .05 amounts to a larger log-odds change than changing it from .1 to .15. We also notice that the plots for $p = .1$ and $p = .9$ are mirror images of each other. Therefore, the log-odds changes are the same for complementary changes in probabilities $Pr(\alpha \mid \beta)$ and $Pr(\bar{\alpha} \mid \beta)$.

We close this section by emphasizing that the results above identify parameter changes that guarantee keeping queries within certain ranges. However, if the Bayesian network has specific properties, such as a specific topology, it is possible for the query to be robust against parameter changes that are outside the identified bounds.

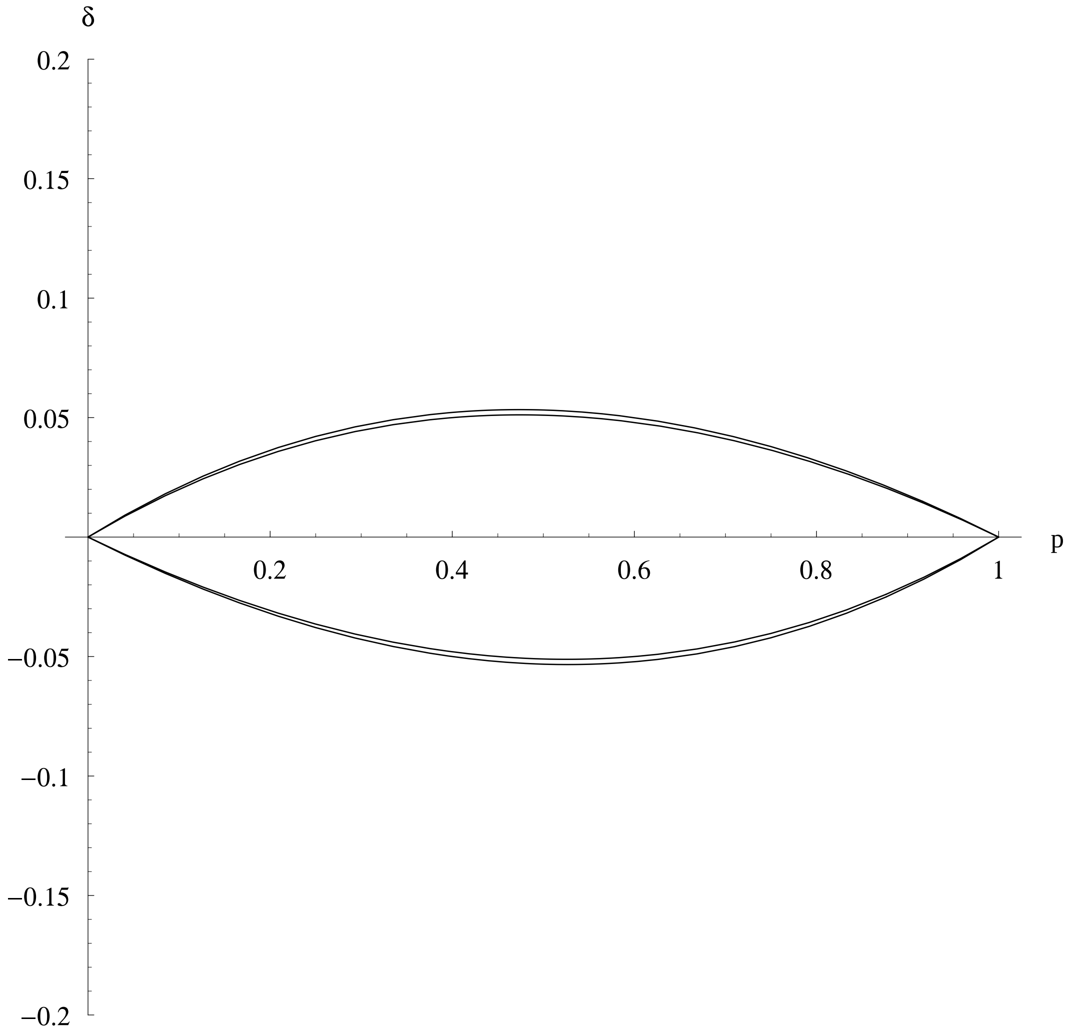


Figure 3.4: The plot of δ against p that would guarantee the query $Pr(y \mid \mathbf{e}) = .6$ to stay within the interval $[.55, .65]$, where p and $p' = p + \delta$ are the initial and new parameter values respectively. The outer envelope guarantees the query to remain no more than .65, while the inner envelope guarantees the query to remain no less than .55.

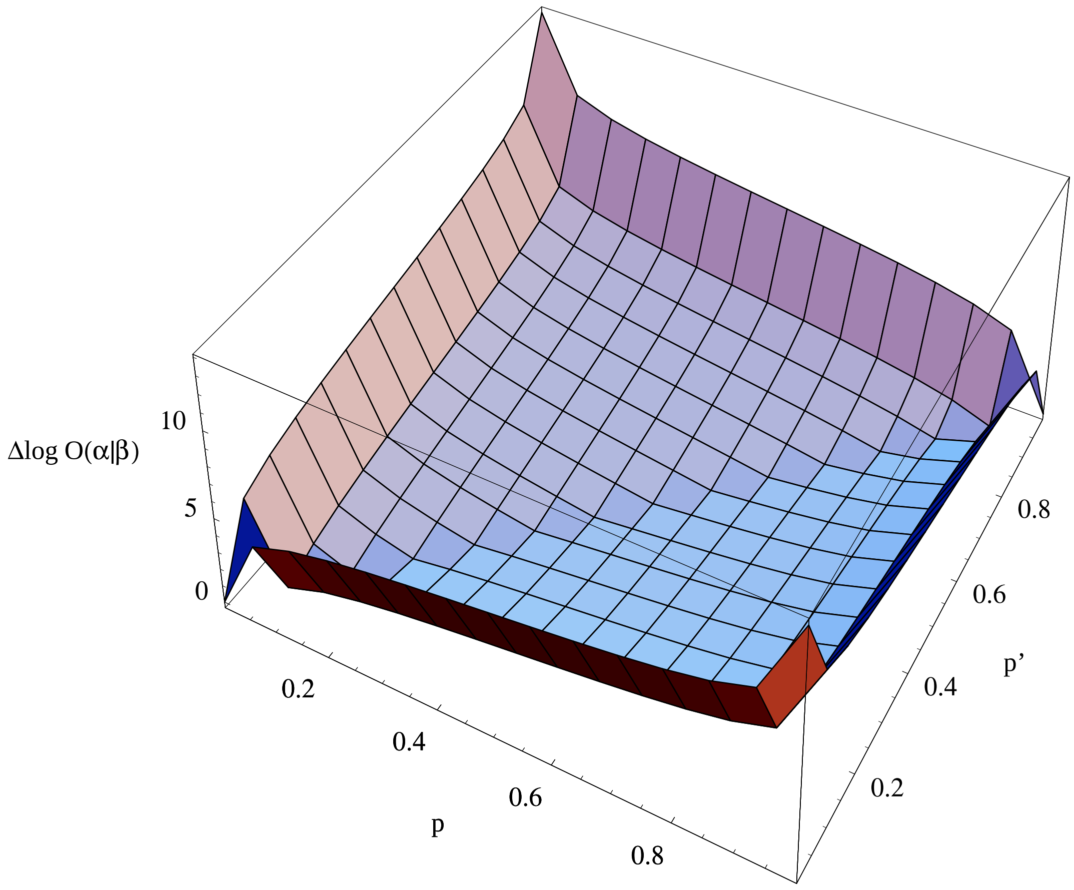


Figure 3.5: The plot of the log-odds change, $|\Delta \ln O(\alpha \mid \beta)| = |\ln O'(\alpha \mid \beta) - \ln O(\alpha \mid \beta)|$, against its initial and new probability values, $p = Pr(\alpha \mid \beta)$ and $p' = p + \delta = Pr'(\alpha \mid \beta)$ respectively.

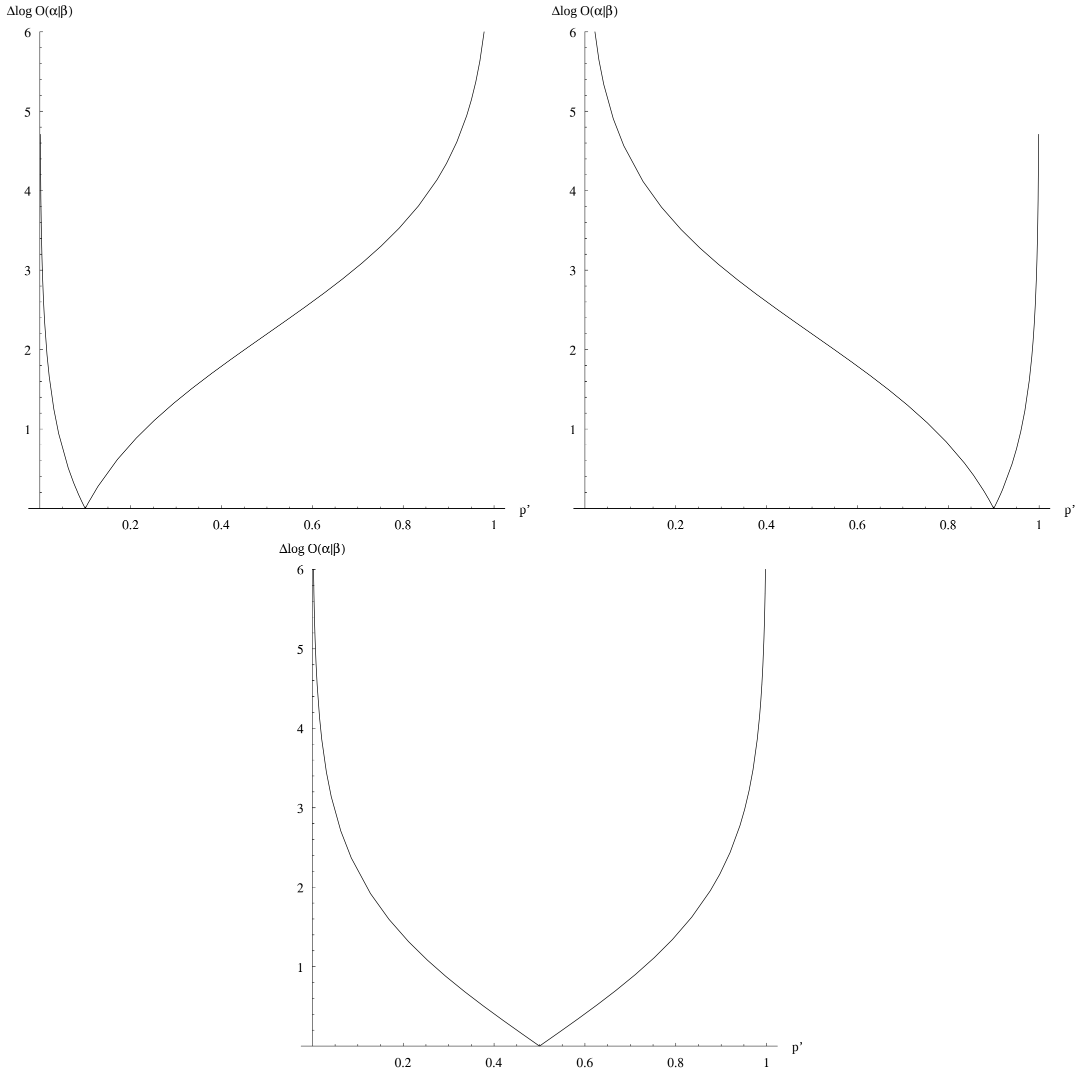


Figure 3.6: The plots of the log-odds change, $|\Delta \ln O(\alpha \mid \beta)| = |\ln O'(\alpha \mid \beta) - \ln O(\alpha \mid \beta)|$, against its new probability value $p' = p + \delta = Pr'(\alpha \mid \beta)$, for several initial probability values $p = Pr(\alpha \mid \beta)$: $p = .1$ (top left), $p = .9$ (top right), $p = .5$ (bottom).

CHAPTER 4

Quantifying Belief Changes

In this chapter, we will discuss the problem of quantifying belief changes. In particular, we want to quantify the global belief changes between two probabilistic states of belief. To achieve this, we propose a distance measure that allows us to bound the amount of global belief change that results from transforming one probabilistic state of belief into another [CD02a, CD05a]. Specifically, given a probability distribution Pr representing an initial state of belief, and a distribution Pr' representing a new state of belief, we define a distance measure that allows us to tightly bound belief changes as follows: $e^{-d} \leq O'(\alpha \mid \beta)/O(\alpha \mid \beta) \leq e^{d}$. Here, d is the proposed distance computed between Pr and Pr', α and β are arbitrary events, $O(\alpha \mid \beta)$ is the odds of conditional event $\alpha \mid \beta$ under Pr, and $O'(\alpha \mid \beta)$ is the odds of conditional event $\alpha \mid \beta$ under Pr'. We show a number of theoretical results about the proposed measure and then present two of its key applications.

On the theoretical side, we prove that our proposed measure satisfies the three properties of distance. We also contrast our distance measure with classical measures, including KL-divergence [KL51], where we present results on its ability to bound belief changes. Specifically, we show that the belief change between two states of belief can be unbounded, even when the KL-divergence between them tends to zero. We show, however, that KL-divergence can be used to bound the average change in beliefs as opposed to the worst-case change in beliefs.

On the practical side, we apply our distance measure to sensitivity analysis of Bayesian networks, an area that concerns itself with bounding global belief changes that result from applying a local perturbation to a Bayesian network [CGH97, CPO99, Dar03, Kv00, Las95, vR01], as discussed in Chapter 3. We show three key results here. First, we show that if Pr is the distribution induced by a Bayesian network $\mathcal{B}$, and if Pr' is the distribution induced by a Bayesian network $\mathcal{B}'$ that results from changing some conditional probability distribution in $\mathcal{B}$, the distance measure between Pr and Pr' can be computed locally by only examining the changed parameters. Second, we use our distance measure to provide a bound on the change in a query that results from a local CPT change, and show that this bound generalizes and provides more insights into the bound given in Theorem 3.3. Third, we use our distance measure to prove the optimality of the proportional scheme defined in Definition 2.1, which is a prevalent, but formally unjustified, technique in the literature on sensitivity analysis used to change the CPTs of multi-valued variables [Dar03, Kv00, Las95].

4.1 A Distance Measure for Bounding Probabilistic Belief Changes

Our proposed measure is defined between two probability distributions as follows.

Definition 4.1 *Let Pr and Pr' be two probability distributions over the same set of worlds ω. We define a measure $D(Pr, Pr')$ as follows:*

$$D(Pr, Pr') \overset{\text{def}}{=} \ln \max_{\omega} \frac{Pr'(\omega)}{Pr(\omega)} - \ln \min_{\omega} \frac{Pr'(\omega)}{Pr(\omega)},$$

where we also define, $0/0 \overset{\text{def}}{=} 1$ *and* $\infty/\infty \overset{\text{def}}{=} 1$.

We say that two probability distributions Pr and Pr' *have the same support*, if for every world ω, $Pr(\omega) = 0$ iff $Pr'(\omega) = 0$. Note that $D(Pr, Pr') = \infty$ iff two distributions Pr and Pr' do not have the same support.

Our first result on this measure is that it satisfies the three properties of distance, and hence, it is a *distance measure*.

Theorem 4.1 *Let Pr, Pr' and Pr'' be three probability distributions over the same set of worlds. The distance measure defined in Definition 4.1 satisfies these three properties:*

Positiveness *$D(Pr, Pr') \geq 0$, and $D(Pr, Pr') = 0$ iff $Pr = Pr'$;*

Symmetry *$D(Pr, Pr') = D(Pr', Pr)$;*

Triangle inequality *$D(Pr, Pr') + D(Pr', Pr'') \geq D(Pr, Pr'')$.*

Our interest in this distance measure stems from two reasons. First, it can be easily computed in a number of practical situations that we will discuss in later sections. Second, it allows us to bound the difference in beliefs captured by two probability distributions.

Theorem 4.2 *Let Pr and Pr' be two probability distributions over the same set of worlds, and α and β be arbitrary events. Given the distance measure $D(Pr, Pr')$ between Pr and Pr', we have the following bound:*

$$e^{-D(Pr,Pr')} \leq \frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} \leq e^{D(Pr,Pr')},$$

where $O(\alpha \mid \beta)$ and $O'(\alpha \mid \beta)$ are the odds of $\alpha \mid \beta$ under distributions Pr and Pr' respectively. The bound is tight in the sense that for every pair of distributions

Pr and Pr', there are events α and β such that:

$$\begin{aligned}\frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} &= e^{D(Pr,Pr')};\\ \frac{O'(\bar{\alpha} \mid \beta)}{O(\bar{\alpha} \mid \beta)} &= e^{-D(Pr,Pr')}.\end{aligned}$$

We can express the bound given in Theorem 4.2 in two other useful forms. First, we can use logarithms:

$$|\ln O'(\alpha \mid \beta) - \ln O(\alpha \mid \beta)| \leq D(Pr, Pr'). \tag{4.1}$$

Second, we can use probabilities instead of odds to express the bound on $Pr'(\alpha \mid \beta)$:

$$\frac{e^{-d}p}{(e^{-d}-1)\,p+1} \leq Pr'(\alpha \mid \beta) \leq \frac{e^{d}p}{(e^{d}-1)\,p+1}, \tag{4.2}$$

where $p = Pr(\alpha \mid \beta)$ and $d = D(Pr, Pr')$. The bounds on $Pr'(\alpha \mid \beta)$ are plotted against p for several values of d in Figure 4.1.

In the applications we will discuss next, Pr is a distribution that represents an initial state of belief, and Pr' is a distribution that represents a new state of belief, which results from applying some kind of local change to the initial state of belief. Examples include a change in some conditional belief or the incorporation of new soft evidence. Our goal is then to assess the global impact of such local belief changes. According to Theorem 4.2, if we are able to compute the distance measure $D(Pr, Pr')$, we can bound global belief changes in a very precise sense, by using Inequality 4.2 to compute the bound on any query $Pr'(\alpha \mid \beta)$, given its initial value $Pr(\alpha \mid \beta)$. We will later show two applications from sensitivity analysis and belief revision where our distance measure can be computed efficiently.

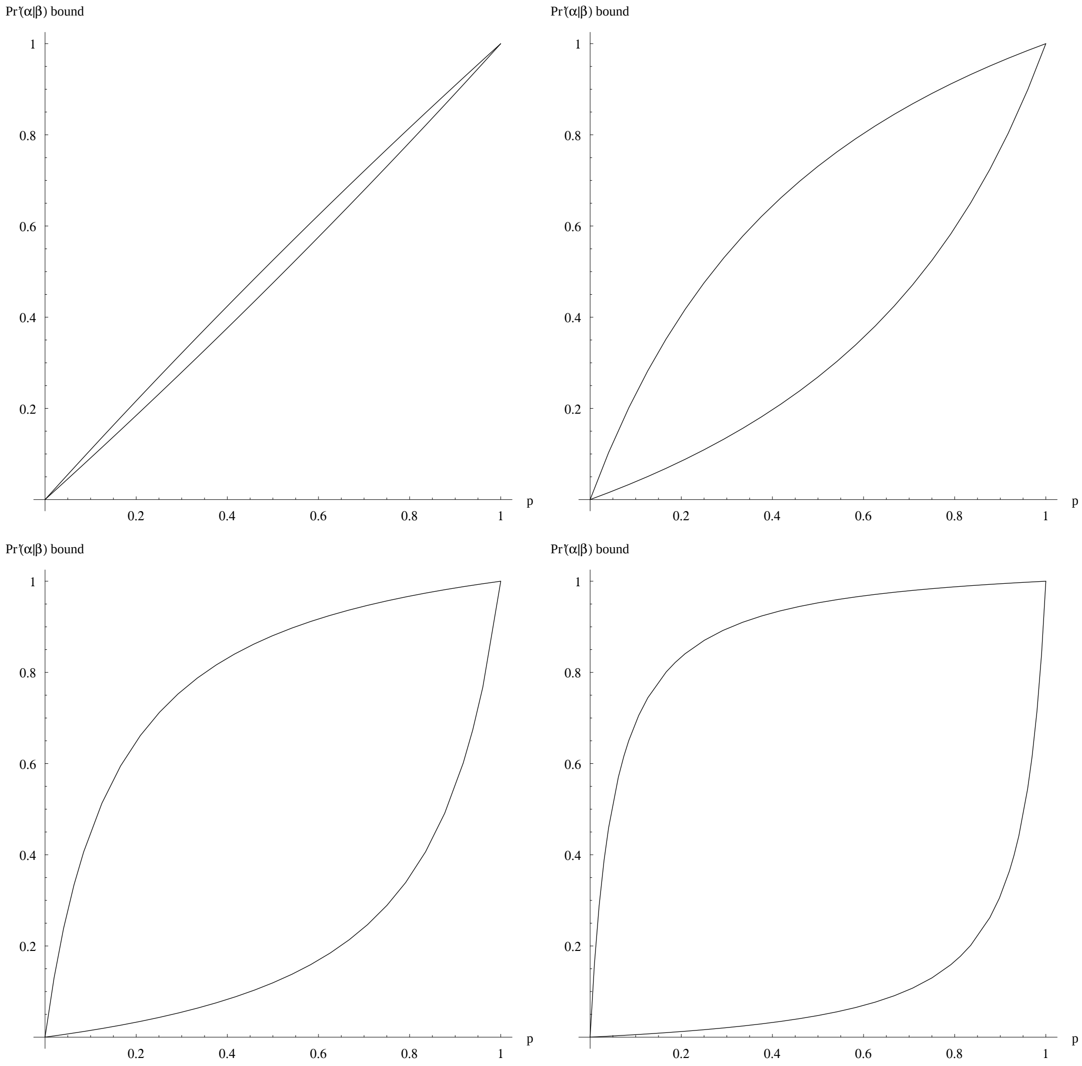


Figure 4.1: The plots of the bounds on $Pr'(\alpha \mid \beta)$ against $p = Pr(\alpha \mid \beta)$, for several values of the distance measure $d = D(Pr, Pr')$, as given by Inequality 4.2: $d = .1$ (top left), $d = 1$ (top right), $d = 2$ (bottom left), and $d = 3$ (bottom right).

4.2 Comparison with Existing Measures

Before we discuss the applications of our distance measure, we first need to settle a major question: can we bound belief changes in the sense given above using one of the classical probabilistic measures? We will show next that this is not possible using at least two of the most commonly used measures: KL-divergence and Euclidean distance. We will show, however, that KL-divergence can be used to provide an average-case bound on belief changes, and also provide a relationship between that bound and ours.

We start first with the Kullback-Leibler divergence, or *KL-divergence*, which is one of the most common measures for comparing probability distributions [KL51], defined as follows.

Definition 4.2 *Let Pr and Pr' be two probability distributions over the same set of worlds ω. The KL-divergence between Pr and Pr' is defined as:*

$$KL(Pr, Pr') \overset{\text{def}}{=} -\sum_{\omega} Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)}.$$

The first thing we note about KL-divergence is that it is incomparable with our distance measure.[1]

Example 4.1 *Consider the following distributions, Pr, Pr' and Pr'', over worlds ω_1, ω_2 and ω_3:*

$$\begin{array}{lll}
Pr(\omega_1) = .5, & Pr(\omega_2) = .25, & Pr(\omega_3) = .25; \\
Pr'(\omega_1) = .5, & Pr'(\omega_2) = .3, & Pr'(\omega_3) = .2; \\
Pr''(\omega_1) = .43, & Pr''(\omega_2) = .32, & Pr''(\omega_3) = .25.
\end{array}$$

[1] Note that KL-divergence is asymmetric, and is thus technically not a distance measure.

Computing the KL-divergence and our distance measure gives us:

$$\begin{aligned} KL(Pr, Pr') &= .0102, \quad & KL(Pr, Pr'') &= .0137; \\ D(Pr, Pr') &= .405, & D(Pr, Pr'') &= .398. \end{aligned}$$

Therefore, according to KL-divergence, Pr' is closer to Pr than Pr'', while according to our distance measure, Pr'' is closer to Pr than Pr'.

The following example shows that we can make the KL-divergence between two distributions arbitrarily close to 0, while keeping some odds ratio $O'(\alpha \mid \beta)/O(\alpha \mid \beta)$ arbitrarily close to some constant k. Therefore, KL-divergence cannot be used to bound belief changes as permitted by our distance measure.

Example 4.2 *Consider the following distributions, Pr and Pr', over worlds ω_1, ω_2 and ω_3:*

$$\begin{aligned} Pr(\omega_1) &= p, & Pr(\omega_2) &= q - p, & Pr(\omega_3) &= 1 - q; \\ Pr'(\omega_1) &= kp, & Pr'(\omega_2) &= q - kp, & Pr'(\omega_3) &= 1 - q; \end{aligned}$$

where $0 \leq p \leq q \leq 1$ and $0 \leq k \leq q/p$. The KL-divergence between Pr and Pr' is:

$$KL(Pr, Pr') = -p \ln k - (q - p) \ln \frac{q - kp}{q - p}.$$

Suppose we have events $\alpha = \omega_1$ and $\beta = \omega_1 \vee \omega_2$. The odds ratio of $\alpha \mid \beta$ between Pr and Pr' is:

$$\frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} = \frac{k\,(q - p)}{q - kp}.$$

We can see that as p approaches 0, the KL-divergence also approaches 0, while the odds ratio $O'(\alpha \mid \beta)/O(\alpha \mid \beta)$ approaches k.

In Example 4.2, we condition on event β, where $Pr(\beta) = q$ where q can be arbitrarily large. However, the probability of event α, which is p under Pr and

kp under to Pr', is very small as p approaches 0, meaning that although we have $Pr'(\alpha)/Pr(\alpha) = k$, this ratio is virtually ignored by KL-divergence because the term $-p \ln k$ is very small. More generally, the "contribution" of a world ω to KL-divergence is equal to $-Pr(\omega)\ln(Pr'(\omega)/Pr(\omega))$. Therefore, for a fixed ratio $Pr'(\omega)/Pr(\omega)$, this "contribution" becomes closer to 0 as $Pr(\omega)$ decreases, and becomes infinitesimal when $Pr(\omega)$ approaches 0.

Another popular measure to compare two probability distributions Pr and Pr' is the Euclidean distance, defined as:

$$Eucl(Pr, Pr') \stackrel{\text{def}}{=} \sum_{\omega} \sqrt{(Pr'(\omega) - Pr(\omega))^2}.$$

That means, when computing the Euclidean distance, we add up the squared differences between pairs of probability values. Therefore, this measure has the same problem as KL-divergence, as even if there is a large relative change in the probability of a world from Pr to Pr', it will be ignored if this probability is very small. Consequently, we cannot provide any guarantee on the ratio $O'(\alpha \mid \beta)/O(\alpha \mid \beta)$, no matter how small the Euclidean distance is (unless it is zero). To summarize, neither KL-divergence nor Euclidean distance can be used to provide guarantees similar to the one given in Theorem 4.2 using our distance measure.

Finally, we note that our distance measure is an improvement over computing the L-infinity metric:

$$L_\infty(\ln Pr, \ln Pr') \stackrel{\text{def}}{=} \max_{\omega} |\ln Pr'(\omega) - \ln Pr(\omega)|,$$

since our distance measure is computed from both the maximum and minimum values of $\ln Pr'(\omega) - \ln Pr(\omega)$. If $L_\infty(\ln Pr, \ln Pr') = l$, we can conclude that $-l \leq \ln Pr'(\omega) - \ln Pr(\omega) \leq l$ for any ω, and prove the following two guarantees. First, if α and β are arbitrary events, we have $e^{-2l} \leq O'(\alpha \mid \beta)/O(\alpha \mid \beta) \leq e^{2l}$. Second, our distance measure is bounded such that $D(Pr, Pr') \leq 2l$. However, the first guarantee is also a direct result of the second guarantee, and by computing the exact value of the distance measure $D(Pr, Pr')$, we can obtain a bound that is no worse than the bound obtained from $L_\infty(\ln Pr, \ln Pr')$. Therefore, our distance measure should always be preferred.

4.2.1 Worst-Case Bound vs. Average-Case Bound

Even though KL-divergence cannot be used to bound belief changes as discussed above, it can still be used to offer a bound on the average change in beliefs, as proved by the following theorem.

Theorem 4.3 *Let Pr and Pr' be two probability distributions over the same set of worlds. Let α and β be arbitrary events. We have:*

$$\begin{aligned} &KL(Pr, Pr') \\ &\geq \;\; -Pr(\beta)\left(Pr(\alpha \mid \beta) \ln \frac{Pr'(\alpha \mid \beta)}{Pr(\alpha \mid \beta)} + (1 - Pr(\alpha \mid \beta)) \ln \frac{1 - Pr'(\alpha \mid \beta)}{1 - Pr(\alpha \mid \beta)}\right), \end{aligned}$$

or alternatively, in odds:

$$KL(Pr, Pr') \geq Pr(\beta)\left(\ln \frac{O'(\alpha \mid \beta) + 1}{O(\alpha \mid \beta) + 1} - \frac{O(\alpha \mid \beta)}{O(\alpha \mid \beta) + 1} \ln \frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)}\right).$$

According to this theorem, KL-divergence can be used to provide a guarantee on the new odds value $O'(\alpha \mid \beta)$ in terms of the initial odds value $O(\alpha \mid \beta)$. However, the provided guarantee depends on the probability of β, where the quality of the guarantee degrades as this probability decreases. This echoes Example 4.2, where we can get a constant log-odds change even when the KL-divergence approaches 0, since the probability $Pr(\beta)$ also approaches 0. To give better insights into the guarantee offered by Theorem 4.3, we plot in Figure 4.2 the bounds on $Pr'(\alpha \mid \beta)$ against $p = Pr(\alpha \mid \beta)$ for different values of $KL(Pr, Pr')$ and $Pr(\beta)$.

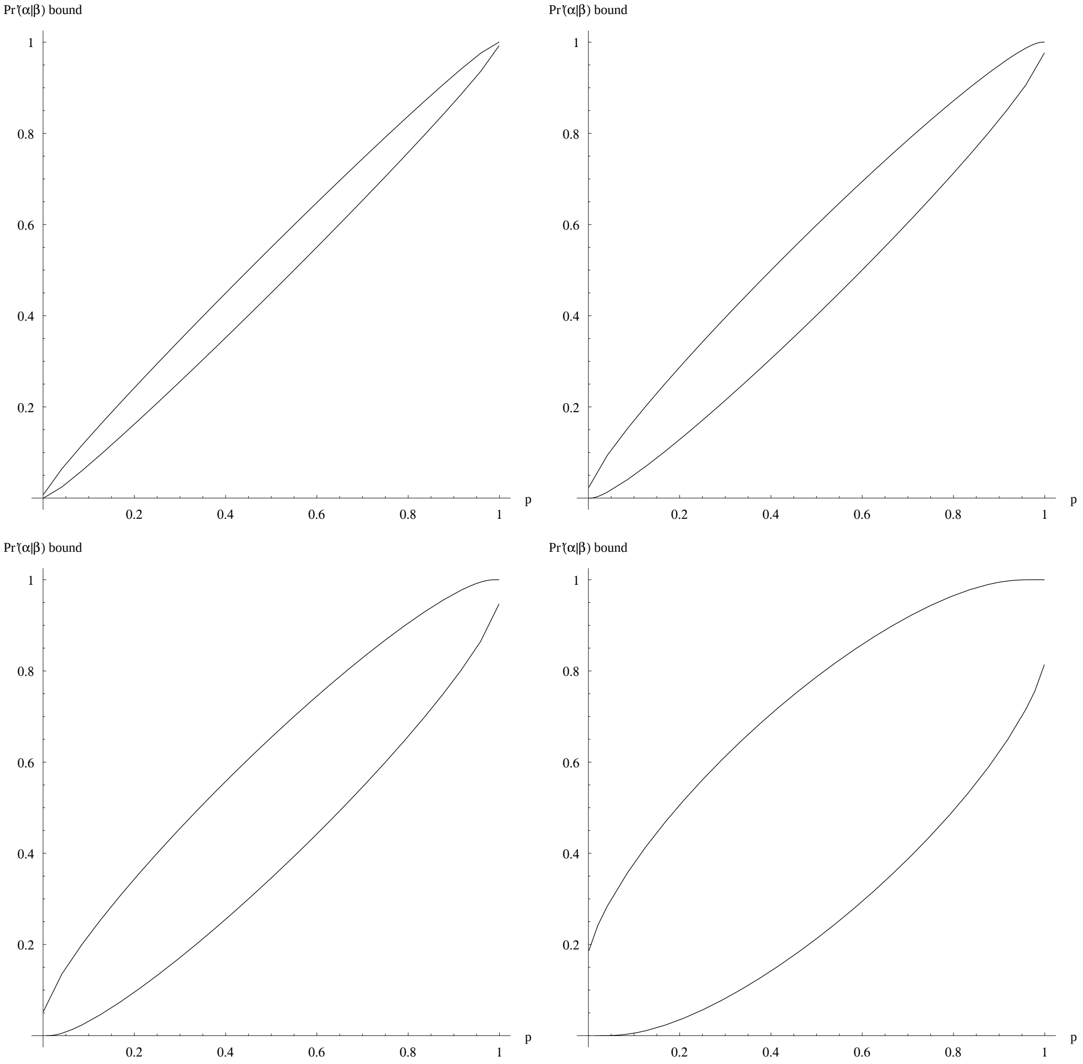


Figure 4.2: The plots of the bounds on $Pr'(\alpha \mid \beta)$ against $p = Pr(\alpha \mid \beta)$, for several values of $KL(Pr, Pr')$ and $Pr(\beta)$, as given in Theorem 4.3: $KL(Pr, Pr') = .005$ and $Pr(\beta) = 1$ (top left), $KL(Pr, Pr') = .02$ and $Pr(\beta) = 1$ (top right), $KL(Pr, Pr') = .005$ and $Pr(\beta) = .1$ (bottom left), and $KL(Pr, Pr') = .02$ and $Pr(\beta) = .1$ (bottom right).

We can also provide a bound on the change in the odds of $\alpha \mid \beta$ that is independent of the probability of β by taking an average over all possible probabilities of β, but that would be an average-case bound as opposed to the worst-case bound provided by our distance measure.

To further relate the bounds given by our distance measure and KL-divergence, we ask an interesting question: if we are given the value of our distance measure between two distributions, can we put a bound on the KL-divergence between them? The following theorem provides us the answer.

Theorem 4.4 *Given two distributions Pr and Pr', where $D(Pr, Pr') = d > 0$, we have:*

$$KL(Pr, Pr') \leq \frac{d}{e^d - 1} - 1 - \ln \frac{d}{e^d - 1}.$$

Figure 4.3 plots the bound on $KL(Pr, Pr')$ against d given in Theorem 4.4. We can see that the smaller d is, the more we can say about the KL-divergence as we get a tighter bound. If we view our distance measure as providing a worst-case bound on belief changes, and KL-divergence as providing an average-case bound on such changes, it is not surprising that we can say more about the average-case bound (KL-divergence) if the worst-case bound (our distance measure) is less dramatic.

We close this section with some further comments on the suitability of the average-case versus worst-case bounds. In general, average-case bounds have proved to be useful in learning algorithms, while worst-case bounds are more important in common-sense reasoning, where conditioning on unlikely events is not uncommon. However, even in learning algorithms, worst-case analysis can be useful if it gives us tight results.

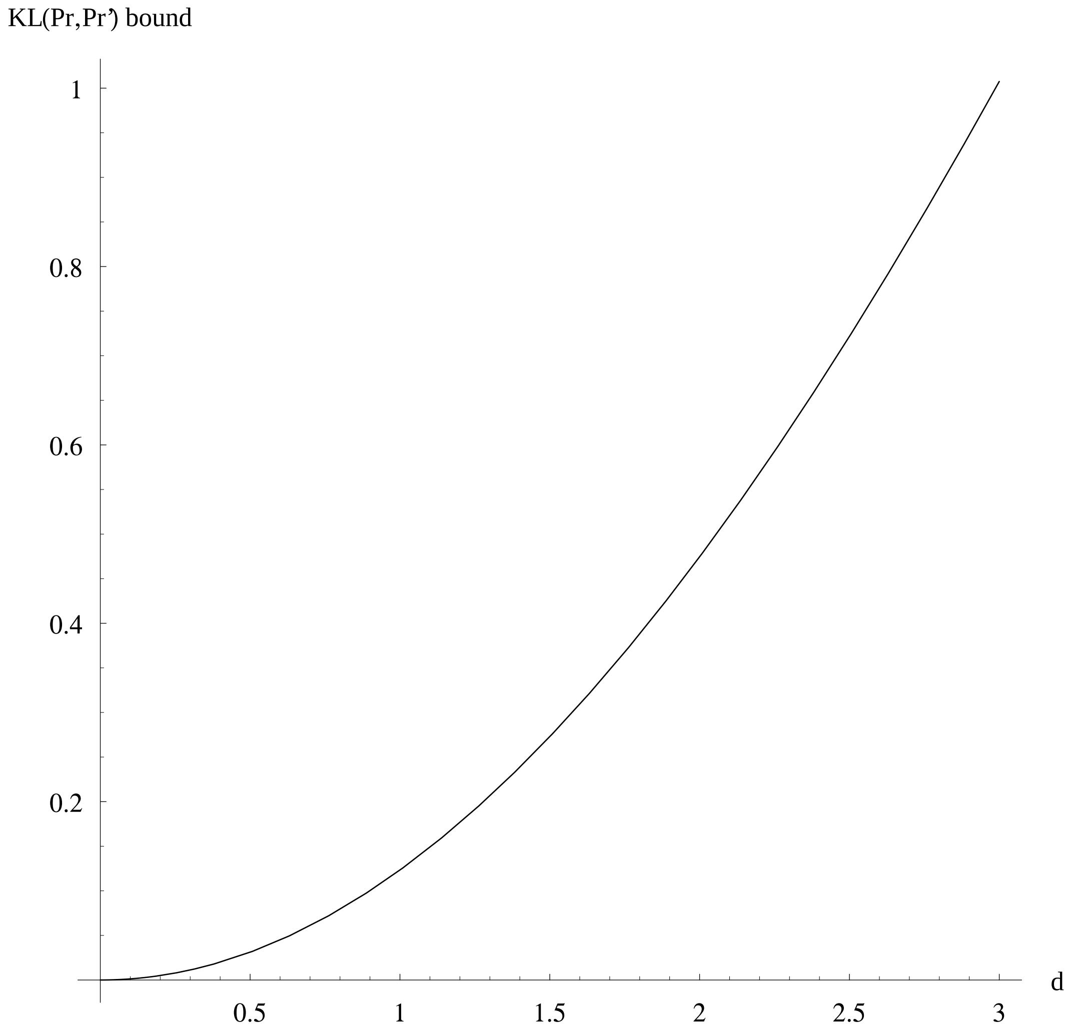


Figure 4.3: The plot of the bound on the KL-divergence $KL(Pr, Pr')$ against the distance measure $d = D(Pr, Pr')$, as given in Theorem 4.4.

4.2.2 Bounding Bayes Factors

Another useful term that can help us further understand our distance measure and its difference with KL-divergence is the *Bayes factor* [Goo50, Goo83, Jef92], defined as follows.

Definition 4.3 *If Pr and Pr' are two probability distributions over the same set of worlds, and γ_1 and γ_2 are arbitrary events, the Bayes factor, denoted by $B_{Pr',Pr}(\gamma_1 : \gamma_2)$, is defined as the following ratio:*

$$B_{Pr',Pr}(\gamma_1 : \gamma_2) \stackrel{\text{def}}{=} \frac{\frac{Pr'(\gamma_1)}{Pr'(\gamma_2)}}{\frac{Pr(\gamma_1)}{Pr(\gamma_2)}}.$$

Instead of using the definition given in Definition 4.1, our distance measure can also be expressed using the Bayes factor:

$$D(Pr, Pr') = \ln \max_{\omega_i,\omega_j} B_{Pr',Pr}(\omega_i : \omega_j).$$

Therefore, our distance measure can be regarded as the logarithm of the maximum Bayes factor of any two worlds between the two distributions. Consequently, it can be used to bound the Bayes factor of any two events between the two distributions.

Corollary 4.1 *If Pr and Pr' are two probability distributions over the same set of worlds, and γ_1 and γ_2 are arbitrary events, we have:*

$$e^{-D(Pr,Pr')} \le B_{Pr',Pr}(\gamma_1 : \gamma_2) \le e^{D(Pr,Pr')}.$$

Theorem 4.2 is a special case of Corollary 4.1, when we substitute the following terms: $\gamma_1 = \alpha \mid \beta$ and $\gamma_2 = \bar{\alpha} \mid \beta$.

While our distance measure provides us a worst-case bound of Bayes factors, the KL-divergence can be perceived as an average-case bound of Bayes factors, as shown by the following theorem.

Theorem 4.5 *Let α be an arbitrary event, and $\gamma_1, \ldots, \gamma_n$ be a set of mutually exclusive and exhaustive events. We have:*

$$0 \leq \sum_i Pr(\gamma_i) \ln B_{Pr',Pr}(\alpha : \gamma_i) - \ln \frac{Pr'(\alpha)}{Pr(\alpha)} \leq KL(Pr, Pr').$$

If instead of an arbitrary partition $\gamma_1, \ldots, \gamma_n$, we are given the set of worlds ω, we get the following equality relation:

$$KL(Pr, Pr') = \sum_\omega Pr(\omega) \ln B_{Pr',Pr}(\alpha : \omega) - \ln \frac{Pr'(\alpha)}{Pr(\alpha)}.$$

Therefore, Theorem 4.5 gives us a bound on the weighted sum of the logarithms of the Bayes factors using KL-divergence.

4.3 Application to Bayesian Networks

We now consider a major application of our distance measure to sensitivity analysis of Bayesian networks [CGH97, CPO99, Dar03, Kv00, Las95, vR01], as discussed in Chapter 3, by asking the following questions with respect to Bayesian networks: what can we say about the global effect of changing some parameter $\theta_{x|\mathbf{u}}$ to a new value $\theta'_{x|\mathbf{u}}$? What is the effect of this local parameter change on the value of some arbitrary query $Pr(\alpha \mid \beta)$?

Theorem 3.3 provided a partial answer, for the case where variable X is binary (or X is multi-valued and we use the proportional scheme defined in Definition 2.1), α is the value y of some variable Y, β is the instantiation $\mathbf{e}$ of some variables $\mathbf{E}$, and neither $\theta_{x|\mathbf{u}}$ nor $\theta'_{x|\mathbf{u}}$ is extreme, i.e., equal to 0 or 1. It provided a formalization of a number of intuitions on the sensitivity of probabilistic queries to changes in network parameters. We now show how our distance measure can be used to derive a generalization of the bound given in Theorem 3.3, without any of the previously mentioned restrictions.

Suppose we are given two Bayesian networks that differ on only a single conditional probability distribution in the CPT of a variable, and the two networks induce probability distributions Pr and Pr' respectively. If we are able to compute the distance between Pr and Pr', $D(Pr, Pr')$, we can then use Theorem 4.2 to provide a guarantee on the global effect of the local CPT change. As it turns out, the distance can be computed locally as shown by the following theorem.

Theorem 4.6 *Let $\mathcal{B}$ and $\mathcal{B}'$ be Bayesian networks that induce distributions Pr and Pr' respectively, and let X be a variable with parents $\mathbf{U}$ in networks $\mathcal{B}$ and $\mathcal{B}'$. If $\mathcal{B}'$ is obtained from $\mathcal{B}$ by changing the conditional probability distribution of variable X given parent instantiation $\mathbf{u}$ from $\Theta_{X|\mathbf{u}}$ to $\Theta'_{X|\mathbf{u}}$, i.e., we change parameter $\theta_{x|\mathbf{u}}$ to $\theta'_{x|\mathbf{u}}$ for every value x, and if $Pr(\mathbf{u}) > 0$, the distance measure between Pr and Pr' is:*[2]

$$\begin{aligned} D(Pr, Pr') &= D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}}) \\ &= \ln \max_x \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \min_x \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}}. \end{aligned}$$

Theorem 4.6 shows that the distance between the global distributions induced by networks $\mathcal{B}$ and $\mathcal{B}'$ is exactly the distance between the local conditional distributions $\Theta_{X|\mathbf{u}}$ and $\Theta'_{X|\mathbf{u}}$, assuming that all other parameters in $\mathcal{B}$ and $\mathcal{B}'$ are the same. It is of great practical importance as it allows us to invoke Theorem 4.2 to provide a generalized sensitivity analysis formula for Bayesian networks.

Corollary 4.2 *Let $\mathcal{B}$ and $\mathcal{B}'$ be Bayesian networks that induce distributions Pr and Pr' respectively, and X be a variable with parents $\mathbf{U}$ in networks $\mathcal{B}$ and $\mathcal{B}'$. If $\mathcal{B}'$ is obtained from $\mathcal{B}$ by changing the conditional probability distribution of X*

[2] *If $Pr(\mathbf{u}) = 0$, $D(Pr, Pr') = 0$.*

given parent instantiation $\mathbf{u}$ *from* $\Theta_{X|\mathbf{u}}$ *to* $\Theta'_{X|\mathbf{u}}$*, i.e., we change parameter* $\theta_{x|\mathbf{u}}$ *to* $\theta'_{x|\mathbf{u}}$ *for every value* x*, and if* $Pr(\mathbf{u}) > 0$*, we have the following bound:*

$$e^{-D(\Theta_{X|\mathbf{u}},\Theta'_{X|\mathbf{u}})} \leq \frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} \leq e^{D(\Theta_{X|\mathbf{u}},\Theta'_{X|\mathbf{u}})}.$$

Theorem 3.3 is a special case of Corollary 4.2, where variable X is binary (or X multi-valued and we use the proportional scheme defined in Definition 2.1). In this case, the distance $D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}})$ is equal to:

$$\begin{aligned} D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}}) &= \left| \ln \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \frac{\theta'_{\bar{x}|\mathbf{u}}}{\theta_{\bar{x}|\mathbf{u}}} \right| \\ &= \left| \ln \frac{\theta'_{x|\mathbf{u}}}{\theta'_{\bar{x}|\mathbf{u}}} - \ln \frac{\theta_{x|\mathbf{u}}}{\theta_{\bar{x}|\mathbf{u}}} \right|. \end{aligned}$$

We have therefore generalized our previous results on sensitivity analysis to arbitrary events and general Bayesian networks. We have also relaxed the condition that neither $\theta_{x|\mathbf{u}}$ nor $\theta'_{x|\mathbf{u}}$ can be extreme.

Under the same circumstances, we can also compute the KL-divergence between two Bayesian networks that differ on only a single conditional probability distribution:

$$KL(Pr, Pr') = Pr(\mathbf{u}) KL(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}}). \tag{4.3}$$

We can see from Equation 4.3 that to compute the KL-divergence between the two Bayesian networks, we need to know the probability $Pr(\mathbf{u})$, the probability of the parent instantiation. Therefore, the KL-divergence cannot be obtained locally as it would require a global computation to obtain $Pr(\mathbf{u})$. This is not necessary when we compute our distance measure as given in Theorem 4.6. Therefore, we have two differences between our distance measure and KL-divergence in this regard. The first difference is semantical in which our distance measure can be used to bound worst-case belief changes, while KL-divergence can be used to bound average-case belief changes. The second difference is computational where our distance measure can be computed in constant time under local changes, while the KL-divergence cannot.

Finally, we close this chapter with another application of our distance measure, which is to prove the optimality of the proportional scheme defined in Definition 2.1. This scheme has been used in all approaches to tuning Bayesian network parameters that we are familiar with [Dar03, Kv00, Las95], yet without justification. As it turns out, we can use our distance measure to prove the optimality of this scheme in a very precise sense.

Theorem 4.7 *When changing a parameter $\theta_{x|\mathbf{u}}$ to $\theta'_{x|\mathbf{u}}$ for a multi-valued variable X, the proportional scheme defined in Definition 2.1, which sets $\theta'_{x^*|\mathbf{u}} = (1 - \theta'_{x|\mathbf{u}})(\theta_{x^*|\mathbf{u}}/(1-\theta_{x|\mathbf{u}}))$ for all $x^* \neq x$, leads to the smallest distance between the initial and new distributions of X given $\mathbf{u}$, which is given by:*

$$\begin{aligned} D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}}) &= \left| \ln \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \frac{1-\theta'_{x|\mathbf{u}}}{1-\theta_{x|\mathbf{u}}} \right| \\ &= \left| \ln \frac{\theta'_{x|\mathbf{u}}}{1-\theta'_{x|\mathbf{u}}} - \ln \frac{\theta_{x|\mathbf{u}}}{1-\theta_{x|\mathbf{u}}} \right| . \end{aligned}$$

Therefore, Theorem 4.7 justifies the use of the proportional scheme on the grounds that it leads to the tightest bound on the amount of the belief change.

CHAPTER 5

Tuning Multiple Bayesian Network Parameters

In Chapter 2, we introduced a procedure that tunes single Bayesian network parameters to enforce a certain query constraint. Although single parameter changes are easy to visualize and compute, they are only a subset of possible parameter changes. Generally, we are interested in changing a set of parameters in a Bayesian network simultaneously to enforce the query constraint. In this chapter, we will expand our work on tuning Bayesian network parameters from the domain of single parameters to multiple parameters [CD04].

To facilitate this, we need to understand the relationship between a joint probability and the set of network parameters [CJK00, Dar03]. A common case involves changing all parameters in the CPT of a single variable. We first show how to find such changes, with little extra computation beyond that needed for finding single parameter changes. This is significant because multiple parameter changes can be more meaningful, and may disturb the probability distribution less significantly than single parameter changes. Practically speaking, this new technique allows us to change both the false-positive and false-negative rates of a certain information source, which can allow the enforcement of certain constraints that cannot be enforced by only changing either the false-positive or the false-negative rate.

As expected, the solution space of multiple parameter changes is a region in the s-dimensional space, where s is the number of parameters involved. For example, the solution space is a half-plane in the two-dimensional space if we change both the false-positive and false-negative rates of a sensor. However, the solution space is difficult to visualize and present to users. Therefore, we would like to identify and report a particular point in the solution space, i.e., a specific amount of change in each of the parameters involved. The key question now is: which point in the solution space should we report? Theoretically, we want to report the point that minimizes model disturbance, which can be quantified using the distance measure given in Definition 4.1, computed between the initial distribution and the new distribution after the parameters have been changed. Practically, we will introduce a simple numerical procedure which finds a solution close to this optimal point.

Instead of just changing multiple parameters in a single CPT, we may also want to change multiple parameters in a subset of CPTs. For example, given a number of sensors with the same reliability, the CPTs of these variables should always be the same, meaning we need to change the parameters in all of these CPTs at the same time. At the end of this chapter, we outline some preliminary work on finding multiple CPT changes to enforce a query constraint.

5.1 Tuning Parameters in a Single CPT

In Section 2.1, we presented a procedure that efficiently finds single parameter changes to enforce a query constraint. This time, instead of only changing a single parameter, we are allowed to change all parameters in the CPT of a single variable X. Similar to what we did in Section 2.1, we introduce a meta-parameter $\tau_{x|\mathbf{u}}$ for every parent instantiation $\mathbf{u}$, such that all parameters in the conditional probability distribution $\Theta_{X|\mathbf{u}}$ can be changed accordingly. If X is binary, we assign $\theta_{x|\mathbf{u}} = \tau_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}} = 1 - \tau_{x|\mathbf{u}}$. If X is multi-valued, we use the proportional scheme to change co-varying parameters, as defined in Definition 2.1. We now introduce a procedure that efficiently finds sufficient parameter changes in a single CPT to enforce a query constraint. We will use Inequality 2.1, $Pr(y \mid \mathbf{e}) \geq \kappa$, as the constraint we would like to enforce.

5.1.1 Finding Sufficient Single CPT Changes

We first reiterate that the probability of an instantiation $\mathbf{e}$, $Pr(\mathbf{e})$, is a linear function in any meta-parameter $\tau_{x|\mathbf{u}}$, and the derivative of $Pr(\mathbf{e})$ with respect to $\tau_{x|\mathbf{u}}$, which we designated as $\pi^{\mathbf{e}}_{x|\mathbf{u}}$, is given by Equation 2.5 if X is binary, or Equation 2.7 if X is multi-valued and we use the proportional scheme to change co-varying parameters.[1] Moreover, we also note that two meta-parameters in the CPT of X, $\tau_{x|\mathbf{u}}$ and $\tau_{x|\mathbf{u}^*}$ where $\mathbf{u} \neq \mathbf{u}^*$, are never multiplied together in the expression of $Pr(\mathbf{e})$, i.e., $\partial^2 Pr(\mathbf{e})/\partial\tau_{x|\mathbf{u}}\partial\tau_{x|\mathbf{u}^*} = 0$ for every pair of parent instantiations $\mathbf{u}$ and $\mathbf{u}^*$ [Dar03].

Therefore, if for every $\tau_{x|\mathbf{u}}$, we apply a change of $\Delta\tau_{x|\mathbf{u}}$, the change in $Pr(\mathbf{e})$ is given by:

$$\Delta Pr(\mathbf{e}) = \sum_{\mathbf{u}} \pi^{\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}}. \tag{5.1}$$

To enforce Inequality 2.1, it suffices to ensure that $Pr(y, \mathbf{e}) \geq \kappa \cdot Pr(\mathbf{e})$, or equivalently:

$$pr(y, \mathbf{e}) + \Delta Pr(y, \mathbf{e}) \geq \kappa \left(pr(\mathbf{e}) + \Delta Pr(\mathbf{e})\right),$$

where $pr(\mathbf{e})$ and $pr(y, \mathbf{e})$ are the current probabilities of $\mathbf{e}$ and $y, \mathbf{e}$ respectively. If for every $\tau_{x|\mathbf{u}}$, we apply a change of $\Delta\tau_{x|\mathbf{u}}$, from Equation 5.1, we have:

$$pr(y, \mathbf{e}) + \sum_{\mathbf{u}} \pi^{y,\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \geq \kappa \left(pr(\mathbf{e}) + \sum_{\mathbf{u}} \pi^{\mathbf{e}}_{x|\mathbf{u}} \Delta\tau_{x|\mathbf{u}}\right).$$

Rearranging the terms, we get the following result.

[1] For simplicity of presentation, we will assume X is binary from now on.

Corollary 5.1 *To satisfy Inequality 2.1, $Pr(y \mid \mathbf{e}) \geq \kappa$, for every $\tau_{x|\mathbf{u}}$, we need to change it by $\Delta\tau_{x|\mathbf{u}}$, such that:*

$$pr(y, \mathbf{e}) - \kappa \cdot pr(\mathbf{e}) \geq \sum_{\mathbf{u}} \Delta\tau_{x|\mathbf{u}} \left(-\pi^{y,\mathbf{e}}_{x|\mathbf{u}} + \kappa \cdot \pi^{\mathbf{e}}_{x|\mathbf{u}} \right),$$

where $pr(\mathbf{e})$ and $pr(y, \mathbf{e})$ are the current probabilities of $\mathbf{e}$ and $y, \mathbf{e}$ respectively, and the constants $\pi^{\mathbf{e}}_{x|\mathbf{u}}$ and $\pi^{y,\mathbf{e}}_{x|\mathbf{u}}$ are defined by Equation 2.5.

Therefore, to enforce the query constraint by changing all parameters in the CPT of X, we need to solve for possible combinations of $\tau_{x|\mathbf{u}}$ in Corollary 5.1. The solution space can be found by solving for the equality condition, and it is in the shape of a half-space due to the linearity of the terms.

To find the solution space of single CPT changes for all CPTs in a Bayesian network, we need to compute the derivatives $\partial Pr(y, \mathbf{e})/\partial\tau_{x|\mathbf{u}}$ and $\partial Pr(\mathbf{e})/\partial\tau_{x|\mathbf{u}}$ for all network parameters. They can be computed using the join-tree algorithm [JLO90, SS86] or the differential approach [Dar03]. The complexity of this computation is $O(n2^w)$, where n is the number of variables in the Bayesian network and w is the tree width, which is a quantity that depends on the connectivity of the network structure [Dar03]. This complexity is the same as that of computing the probability of evidence $Pr(\mathbf{e})$. Moreover, this means that the procedure of finding single CPT changes has the same complexity as the previous procedure of finding single parameter changes, as given in Corollary 2.5.

We now present an example of finding sufficient single CPT changes.

Example 5.1 *Consider again the Bayesian network* Fire. *We now set the evidence $\mathbf{e} = \{Report = true, Smoke = false\}$, i.e., smoke is not observed, but there is report of people evacuating the building. The current Bayesian network gives us the posterior probabilities $Pr(Tampering = true \mid \mathbf{e}) = .50$. However,* *we would like this query value to be at least* .65*, and we want to change the CPT of Report to enforce this constraint. Assume that we apply changes to the parameters* $Pr(\textit{Report} = \textit{false} \mid \textit{Leaving} = \textit{true})$*, i.e., the probability of not receiving an evacuation report when there is an evacuation (false-negative), and* $Pr(\textit{Report} = \textit{true} \mid \textit{Leaving} = \textit{false})$*, i.e., the probability of receiving an evacuation report when there is no evacuation (false-positive), respectively. Their current values are* .25 *and* .01 *respectively, and the changes applied to them are denoted by* δ_1 *and* δ_2*. Corollary 5.1 gives us the inequality that specifies the solution space:*

$$-.00329 \geq .00390 \cdot \delta_1 + .622 \cdot \delta_2.$$

The solution space is plotted in Figure 5.1. The line indicates the set of points where the equality condition $Pr(\textit{Tampering} = \textit{true} \mid \mathbf{e}) = .65$ *holds and is the boundary of the solution space, while the solution space is the region below the line. Therefore, we can ensure that* $Pr(\textit{Tampering} = \textit{true} \mid \mathbf{e}) \geq .65$ *by applying any parameter change in this region.*

5.1.2 Approximating Optimal Single CPT Changes

We now address the second problem of interest in this section, which is how to find a solution in Corollary 5.1 that minimizes the distance measure between the initial and new distributions, defined in Definition 4.1. A solution close to optimal in this sense can be found using a simple numerical procedure, which is based on the following observations.

The first observation is that if the new distribution Pr' is obtained from the initial distribution Pr by changing only one CPT, the distance measure between Pr and Pr', $D(Pr, Pr')$, can be computed from the local information about this CPT, using the following result.

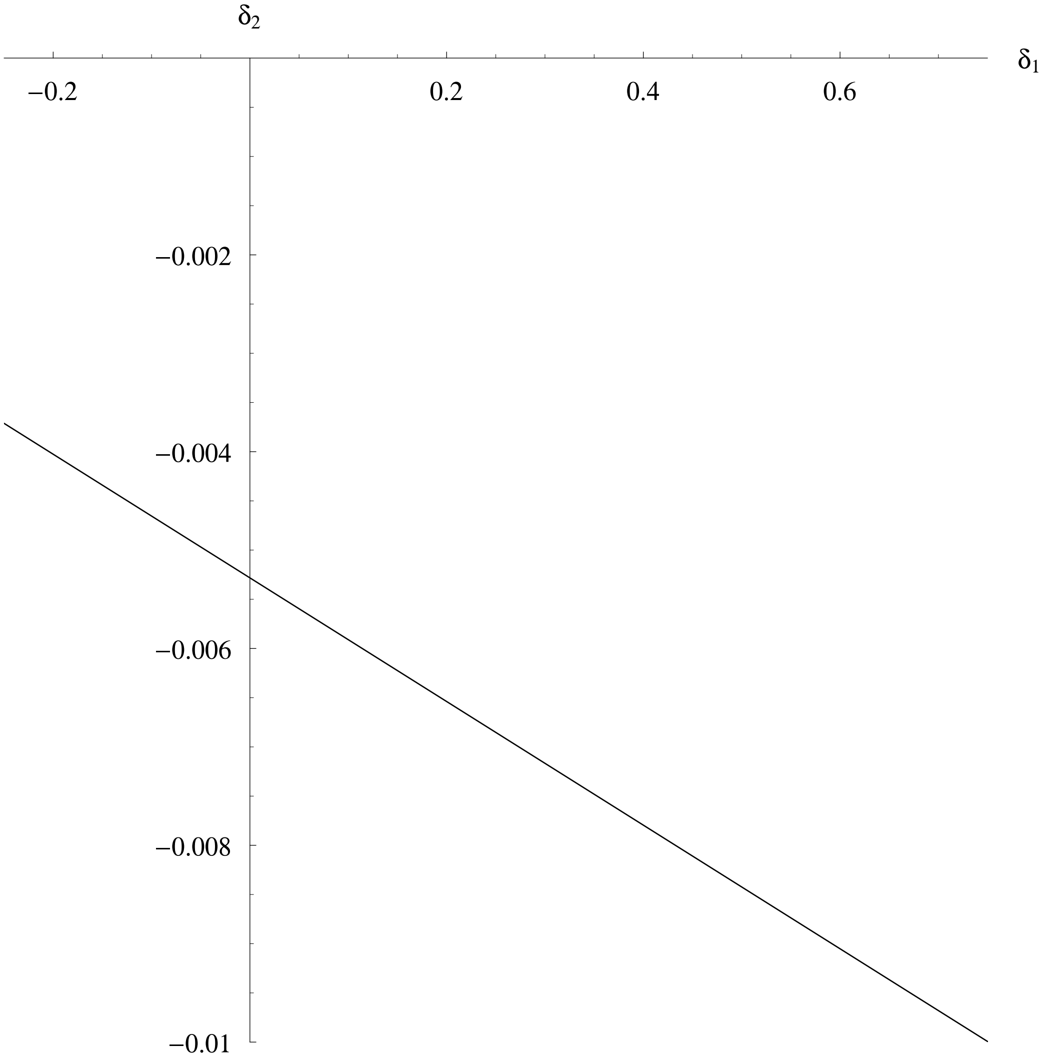


Figure 5.1: The plot of the solution space of the single CPT changes given in Example 5.1. The solution space is the region below the line, which is the boundary of the solution space.

Corollary 5.2 *Let $\mathcal{B}$ and $\mathcal{B}'$ be Bayesian networks that induce distributions Pr and Pr' respectively, and X be a variable with parents $\mathbf{U}$ in networks $\mathcal{B}$ and $\mathcal{B}'$. If $\mathcal{B}'$ is obtained from $\mathcal{B}$ by changing the CPT of X from $\Theta_{X|\mathbf{U}}$ to $\Theta'_{X|\mathbf{U}}$, i.e., we change parameter $\theta_{x|\mathbf{u}}$ to $\theta'_{x|\mathbf{u}}$ for every value x and every parent instantiation $\mathbf{u}$, and if $Pr(\mathbf{u}) > 0$ for every $\mathbf{u}$, the distance measure between Pr and Pr' is:*[2]

$$D(Pr, Pr') = \ln \max_{x,\mathbf{u}} \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \min_{x,\mathbf{u}} \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}}.$$

We will loosen up the definition of the distance measure given in Definition 4.1, and define this value as $D(\Theta_{X|\mathbf{U}}, \Theta'_{X|\mathbf{U}})$, since the previous definition only accepts probability distributions and not CPTs.[3]

The second observation is that we must be able to find an optimal solution on the boundary of the solution space, i.e., the line where $Pr(y \mid \mathbf{e}) = \kappa$, since if there is an optimal solution where $Pr(y \mid \mathbf{e}) > \kappa$, we can always decrease the absolute change in some parameter to satisfy the equality condition, and the distance measure will not increase.

Finally, we observe that the distance measure $D(\Theta_{X|\mathbf{U}}, \Theta'_{X|\mathbf{U}})$ can be approximated by the following term:[4]

$$\max_{Pr(\mathbf{u})>0} \left| \ln O'(x \mid \mathbf{u}) - \ln O(x \mid \mathbf{u}) \right|. \tag{5.2}$$

Note that this term is simply the maximum log-odds change in any meta-parameter $\tau_{x|\mathbf{u}}$. Moreover, for any solution that satisfies $Pr(y \mid \mathbf{e}) = \kappa$, it follows that the solution that minimizes the term in Equation 5.2 is the one where the log-odds

[2] *If $Pr(\mathbf{u}) = 0$ for some $\mathbf{u}$, we can leave the parameters $\theta_{x|\mathbf{u}}$ out when computing the distance measure.*

[3] *In many fields, the ratio between the smallest and largest possible values of a changeable quantity is often called the dynamic range. Therefore, $D(\Theta_{X|\mathbf{U}}, \Theta'_{X|\mathbf{U}})$ is simply the dynamic range of the log-odds changes in the parameter values.*

[4] Previously, we claimed $D(\Theta_{X|\mathbf{U}}, \Theta'_{X|\mathbf{U}})$ is equal to the term in Equation 5.2 [CD04], which turns out to be incorrect. We regret this error.

changes in all parameters in the CPT is the same. This is because to obtain another solution on the line, we must increase the log-odds change in one parameter and decrease it in another, thereby producing a larger maximum log-odds change.

Given the above observations, we can now search for an approximate optimal single CPT parameter change among the solutions in Corollary 5.1, using the following procedure:

1. Pick all meta-parameters $\tau_{x|\mathbf{u}}$ in the CPT of X where the terms $\kappa_{x|\mathbf{u}} = -\pi^{y,\mathbf{e}}_{x|\mathbf{u}} + \kappa \cdot \pi^{\mathbf{e}}_{x|\mathbf{u}}$ are non-zero, and categorize them according to whether this term is positive or negative.

2. Choose a certain amount of log-odds change, and apply it to each meta-parameter $\tau_{x|\mathbf{u}}$, such that whether a parameter is increased or decreased depends on the term $\kappa_{x|\mathbf{u}}$.

3. If $Pr(y \mid \mathbf{e}) = \kappa$ within an acceptable degree of error, return the current point as our solution. Otherwise, try a larger log-odds change if $Pr(y \mid \mathbf{e}) < \kappa$, or a smaller one if $Pr(y \mid \mathbf{e}) > \kappa$. The new amount of log-odds change applied should be determined numerically by the last computed query value of $Pr(y \mid \mathbf{e})$ for a fast rate of convergence.

This numerical procedure is implemented in SamIam [ACC].[5] We now illustrate this procedure with an example.

Example 5.2 *Consider again the solution space given in Example 5.1. Figure 5.2 shows the solution space of single CPT changes, and a curve originating*

[5] To find the optimal solution, we can perform local search starting from the solution we obtained from this procedure to find the point which minimizes the real distance measure $D(\Theta_{X|\mathbf{U}}, \Theta'_{X|\mathbf{U}})$.

from the origin where the log-odds changes in the two parameters involved are the same for every point on the curve (for the case where both parameters are decreased because this gives us the best solution). The intersection of this curve and the boundary of the solution space gives us a solution which minimizes the term in Equation 5.2, which is a near-optimal solution in terms of minimizing the distance measure. To find this solution, we only need to move on the curve using a numerical procedure, until we are at the boundary of the solution space. The solution found is $\delta_1 = -.100$ *and* $\delta_2 = -.00467$*. Therefore, this solution gives us the new parameter values* $Pr(\mathit{Report} = \mathit{false} \mid \mathit{Leaving} = \mathit{true}) = .150$ *and* $Pr(\mathit{Report} = \mathit{true} \mid \mathit{Leaving} = \mathit{false}) = .00533$.

5.1.3 Single CPT Changes vs. Single Parameter Changes

We now proceed to compare applying single CPT changes with applying single parameter for sensitivity analysis purposes. As we have shown, finding both types of changes require the same computation complexity, for computing the partial derivatives of joint probabilities with respect to all parameters. However, solutions of single CPT changes are harder to visualize and present, and it takes a little more time to find a near-optimal solution using the numerical method we proposed.

However, it is advantageous to apply single CPT changes instead of single parameter changes to a Bayesian network in order to satisfy a query constraint. First, single CPT changes are more meaningful and intuitive than single parameter changes. For example, given a sensor in a network, single parameter changes allow us to change only the false-positive or false-negative rate of this sensor, while single CPT changes allow us to change both rates.

Second, for some variable in the network, there may exist single CPT changes,

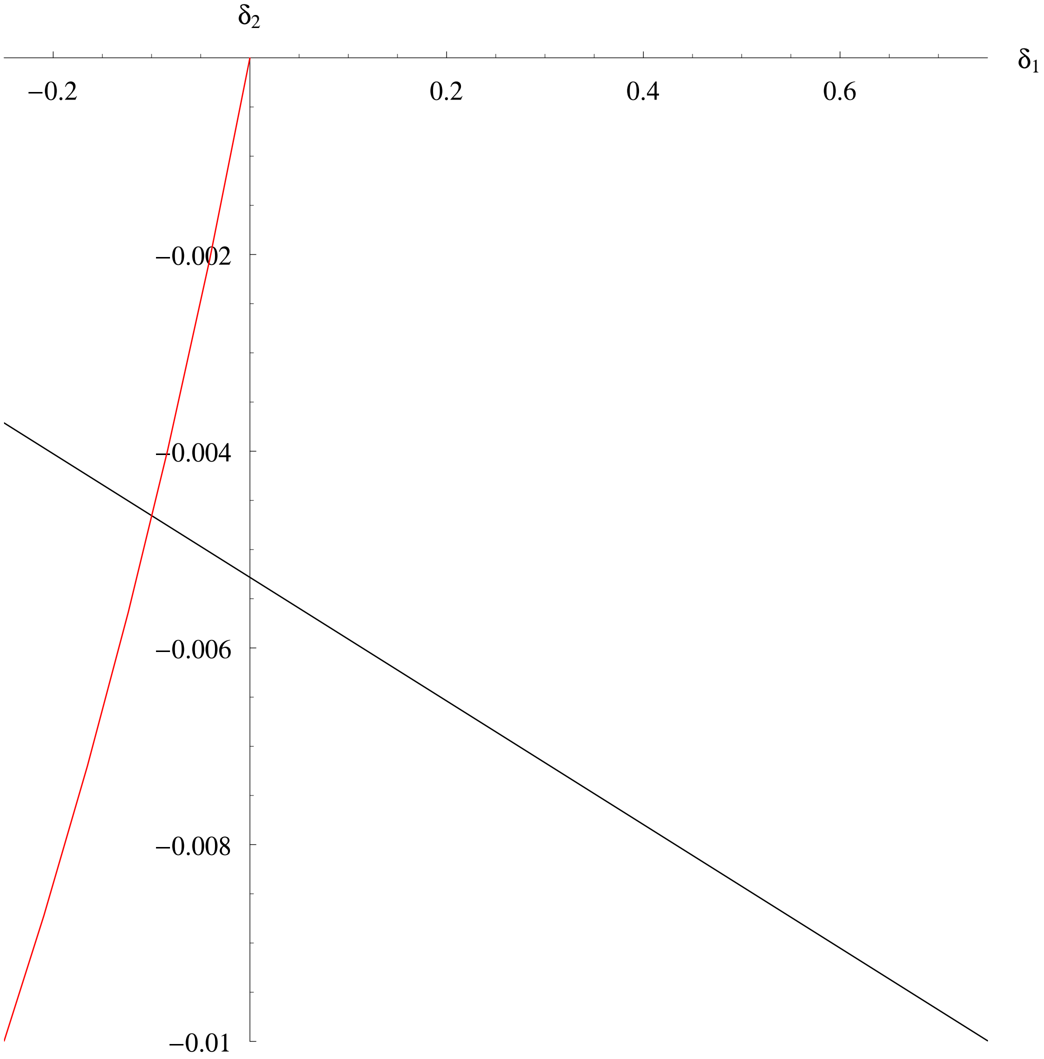
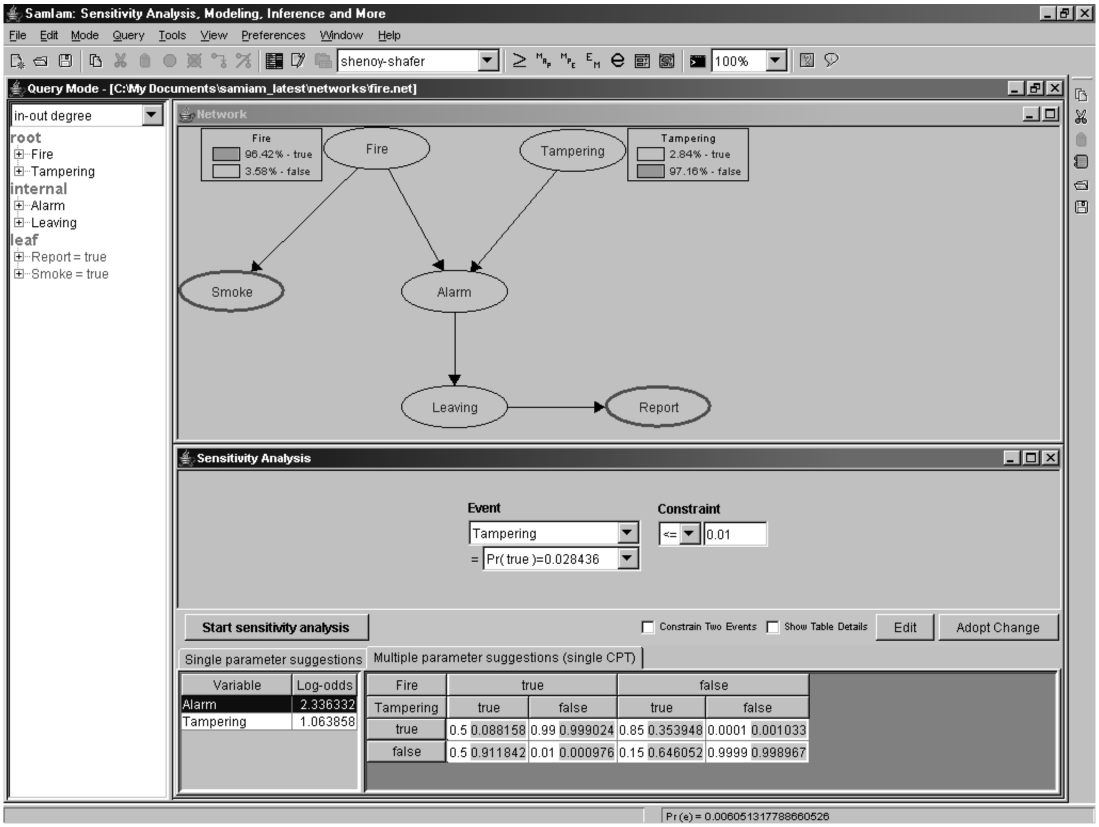


Figure 5.2: The plot of a near-optimal solution of single CPT changes given in Example 5.2. This solution is found by traversing on the curve originating from the origin where the log-odds changes in the two parameters are the same, until it meets the boundary of the solution space.

but not single parameter changes, that can ensure a certain query constraint. For example, consider again the Bayesian network Fire. Suppose we are given evidence $\mathbf{e} = \{Smoke = true, Report = true\}$, i.e., smoke is observed, and there is a report of people evacuating the building. The current Bayesian network gives us the posterior probability $Pr(Tampering = true \mid \mathbf{e}) = .0284$. We may now pose this question: what parameter changes can we apply to decrease this query value to at most .01? If we can only change a single parameter in the network, SamIam returns a simple answer: the only parameter you can change is the prior probability of tampering having occurred, from its initial parameter value of .02 to less than .00700. You cannot change any single parameter in the CPT of *Alarm* to ensure the constraint, and we may be inclined to believe that the parameters in this CPT are irrelevant to the query. However, if we are allowed to change multiple parameters in a single CPT, SamIam returns a new suggestion, telling us that we can indeed change the CPT of *Alarm* to ensure the constraint. The suggestion found by SamIam is shown in Figure 5.3, where the initial parameter values are in white background, and the suggested parameter values are in shaded background. Therefore, we can change the reliability of the alarm when it responds to various scenarios of fire and tampering to enforce the query constraint.

Third, even if changes of both types are available, single CPT changes are often preferred because they disturb the network less significantly, as they incur a smaller distance measure. For example, we can pose another query constraint, where we want to decrease the posterior probability $Pr(Tampering = true \mid \mathbf{e})$ from .0284 to at most .025. This time, for the CPT of *Alarm*, SamIam returns parameter change suggestions of both types. A possible single parameter change is to decrease the probability $Pr(Alarm = true \mid Fire = false, Tampering = true)$ from .85 to .684, incurring a distance measure of .960. On the other hand, if we


SamIam: Sensitivity Analysis, Modeling, Inference and More
File Edit Mode Query Tools View Preferences Window Help
Query Mode - [C:\My Documents\samiam_latest\networks\fire.net]
Network
Fire
Tampering
Smoke
Alarm
Leaving
Report
Sensitivity Analysis
Start sensitivity analysis
Edit
Adopt Change
Single parameter suggestions
Multiple parameter suggestions (single CPT)

shenoy-shafer

100%

in-out degree

root
- Fire
- Tampering

internal
- Alarm
- Leaving

leaf
- Report = true
- Smoke = true

Fire	
▨	96.42% - true
□	3.58% - false

Tampering	
□	2.84% - true
▨	97.16% - false

Event: Tampering = Pr(true)=0.028436

Constraint: <= 0.01

□ Constrain Two Events □ Show Table Details

Variable	Log-odds
Alarm	2.336332
Tampering	1.063858

Fire	true		false	
Tampering	true	false	true	false
true	0.5 0.088158	0.99 0.999024	0.85 0.353948	0.0001 0.001033
false	0.5 0.911842	0.01 0.000976	0.15 0.646052	0.9999 0.998967

Pr(e) = 0.006051317788660526

Figure 5.3: A screenshot of SamIam returning suggestions of single CPT changes for enforcing a query constraint.

change all parameters in the CPT simultaneously, the distance measure incurred by the solution suggested by SamIam is a much smaller value of .435. Therefore, the suggested single CPT change ensures a tighter bound on the change in any query value.

5.2 Tuning Parameters in Multiple CPTs

In this section, we remove the restriction that we are only allowed to change parameters in a single CPT, and assume that we are able to change parameters in multiple CPTs simultaneously. For example, we may want to change all parameters in the CPTs of variables X_1 and X_2, whose parents are $\mathbf{U}_1$ and $\mathbf{U}_2$ respectively. In this case, if we apply a change of $\Delta\tau_{x_1|\mathbf{u}_1}$ to each meta-parameter $\tau_{x_1|\mathbf{u}_1}$ in the CPT of X_1, and a change of $\Delta\tau_{x_2|\mathbf{u}_2}$ to each meta-parameter $\tau_{x_2|\mathbf{u}_2}$ in the CPT of X_2, the change in the joint probability $Pr(\mathbf{e})$ is given by:

$$\Delta Pr(\mathbf{e}) = \sum_{\mathbf{u}_1} \pi^{\mathbf{e}}_{x_1|\mathbf{u}_1} \Delta\tau_{x_1|\mathbf{u}_1} + \sum_{\mathbf{u}_2} \pi^{\mathbf{e}}_{x_2|\mathbf{u}_2} \Delta\theta_{x_2|\mathbf{u}_2} + \sum_{\mathbf{u}_1,\mathbf{u}_2} \pi^{\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2} \Delta\tau_{x_1|\mathbf{u}_1} \Delta\tau_{x_2|\mathbf{u}_2}, \tag{5.3}$$

where $\pi^{\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2}$ denotes the second derivative of $Pr(\mathbf{e})$ with respect to $\tau_{x_1|\mathbf{u}_1}$ and $\tau_{x_2|\mathbf{u}_2}$:

$$\pi^{\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2} \overset{\text{def}}{=} \frac{\partial Pr(\mathbf{e})}{\partial\tau_{x_1|\mathbf{u}_1} \partial\tau_{x_2|\mathbf{u}_2}}. \tag{5.4}$$

To find the solution of parameter changes that satisfies $Pr(y \mid \mathbf{e}) \geq \kappa$, from Equation 5.3, we have:

$$\begin{aligned} & pr(y, \mathbf{e}) - \kappa \cdot pr(\mathbf{e}) \\ \geq\ & \sum_{\mathbf{u}_1} \Delta\tau_{x_1|\mathbf{u}_1} \left(-\pi^{y,\mathbf{e}}_{x_1|\mathbf{u}_1} + \kappa \cdot \pi^{\mathbf{e}}_{x_1|\mathbf{u}_1} \right) + \sum_{\mathbf{u}_2} \Delta\tau_{x_2|\mathbf{u}_2} \left(-\pi^{y,\mathbf{e}}_{x_2|\mathbf{u}_2} + \kappa \cdot \pi^{\mathbf{e}}_{x_2|\mathbf{u}_2} \right) \\ & + \sum_{\mathbf{u}_1,\mathbf{u}_2} \Delta\tau_{x_1|\mathbf{u}_1} \Delta\tau_{x_2|\mathbf{u}_2} \left(-\pi^{y,\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2} + \kappa \cdot \pi^{\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2} \right), \end{aligned} \tag{5.5}$$

where $pr(\mathbf{e})$ and $pr(y, \mathbf{e})$ are the current probabilities of $\mathbf{e}$ and $y, \mathbf{e}$ respectively, the constants $\pi^{\mathbf{e}}_{x_i|\mathbf{u}_i}$ and $\pi^{y,\mathbf{e}}_{x_i|\mathbf{u}_i}$ are defined by Equation 2.5, and the constants $\pi^{\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2}$ and $\pi^{y,\mathbf{e}}_{x_1|\mathbf{u}_1,x_2|\mathbf{u}_2}$ are defined by Equation 5.4.

Therefore, we now need to additionally compute the second partial derivatives of $Pr(y, \mathbf{e})$ and $Pr(\mathbf{e})$ with respect to $\tau_{x_1|\mathbf{u}_1}$ and $\tau_{x_2|\mathbf{u}_2}$ for all pairs of parent instantiations of $\mathbf{u}_1$ and $\mathbf{u}_2$. A simple way to do this is to set evidence on every family instantiation $x_1, \mathbf{u}_1$, then find the derivatives with respect to $\tau_{x_2|\mathbf{u}_2}$ for all $\mathbf{u}_2$ [Dar03]. The complexity of this method is $O(n2^w|\Theta_{X_1|\mathbf{U}_1}|)$, where $|\Theta_{X_1|\mathbf{U}_1}|$ is the size of the CPT of X_1, i.e., the number of family instantiations of X_1. This approach is however limited to non-extreme values of $\tau_{x_1|\mathbf{u}_1}$, yet it allows us to use any general inference algorithm [Dar03]. For extreme parameters, we can use a specific inference approach [Dar03] to obtain these derivatives using the same complexity as given above.

The above results can be expanded to multiple parameter changes involving more than two CPTs. For example, if we change three CPTs simultaneously, we need to compute the third partial derivatives with respect to the corresponding parameters. The complexity of obtaining these higher order derivatives is $O(n2^w \prod_{X_i} |\Theta_{X_i|\mathbf{U}_i}|)$, where $\{X_i\}$ are the variables whose CPTs we are interested in, and $|\Theta_{X_i|\mathbf{U}_i}|$ is the size of the CPT of X_i [Dar03].

We would now like to compute the distance measure of multiple CPT parameter changes in order to find an optimal solution. Although this cannot be easily computed in some cases, for the special case where the families $\{X_1, \mathbf{U}_1\}$ and $\{X_2, \mathbf{U}_2\}$ are disjoint, i.e., X_1 and X_2 do not have a parent-child relationship and do not have a common parent, the distance measure between the initial and new distributions induced can be easily computed as:

$$D(Pr, Pr') = D(\Theta_{X_1|\mathbf{U}_1}, \Theta'_{X_1|\mathbf{U}_1}) + D(\Theta_{X_2|\mathbf{U}_2}, \Theta'_{X_2|\mathbf{U}_2}). \qquad (5.6)$$

Therefore, the total distance measure can be computed as the sum of the distances incurred individually by each of the CPT changes, as computed by Corollary 5.2.[6] Even though we have this restriction of disjointness for Equation 5.6, many CPTs satisfy this condition. For example, when the variables involved are sensors on different variables in a Bayesian network, their families are disjoint, and we can easily compute the distance measure using Equation 5.6.

[6] If the families $X_1, \mathbf{U}_1$ and $X_2, \mathbf{U}_2$ are not disjoint, the distance measure cannot be computed as the sum of the distance measure of the individual CPTs, since a pair of family instantiations of X_1 and X_2 may not be consistent. In this case, the sum is an upper bound of the distance measure, and we can still compute the exact distance measure using a procedure that multiplies two tables (thereby eliminating inconsistent pairs of instantiations), a harder but still manageable process.

Similar to single CPT changes, we are often more interested in finding a near-optimal solution than presenting the whole solution space. As in the previous case, we can find a near-optimal solution on the boundary where $Pr(y \mid \mathbf{e}) = \kappa$, and also on the curve where the log-odds changes in all parameters in each individual CPT are the same. With these two assumptions, we can find the combination of CPT changes that gives us the smallest distance measure.

Because of the computation involved, the key to an automated sensitivity analysis tool that attempts to find multiple CPT changes is to find relevant CPTs to check for solutions, instead of trying all combinations of CPTs, which would be computationally too costly. The first partial derivatives computed for finding single CPT changes can serve as a guide for identifying these relevant CPTs. For many CPTs, the first partial derivatives of the query with respect to the parameters are 0, thereby eliminating them from consideration. On the other hand, we should definitely consider CPTs where small parameters changes can induce large changes in the queries of interest. Therefore, the local search procedure in this case is not as straightforward as the one for single CPT changes.

CHAPTER 6

Sensitivity of Decisions Induced by Bayesian Networks

In previous chapters, we were interested in how the exact value of a probabilistic query responds to Bayesian network parameter changes. However, when using a Bayesian network, we may not be interested in the exact value of a query, but in the decisions made according to the value of the query.

For example, consider a Bayesian network called Pregnancy, whose structure is shown in Figure 6.1 and whose CPTs are shown in Table 6.1. The network represents a scenario where there are three different tests for detecting pregnancy of a cow. Given the results of the three tests, we can compute the probability of pregnancy given the evidence, and if this probability is no less than .9, we can say that we are confident that the cow is pregnant. Here, we are using the Bayesian network as a *classifier*, where we attempt to classify the input (test results) into a small number of (usually two) classes, depending on the query value with respect to a given probability threshold (whether the posterior probability of pregnancy is no less than .9). Therefore, the classifier acts as a logical function, returning yes or no as the answer instead of the exact probability value. The classifier induced under this scenario is shown in Table 6.2.

We can now formally define a classifier induced by a Bayesian network. Given a Bayesian network $\mathcal{B}$, which induces the probability distribution Pr, we select

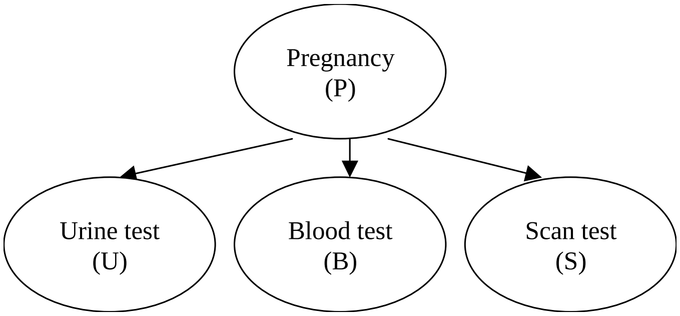


Figure 6.1: The structure of the naive Bayes network Pregnancy.

P	Θ_P
p	.87
$\bar{p}$	.13

P	S	$\Theta_{S\|P}$
p	s	.90
p	$\bar{s}$	.10
$\bar{p}$	s	.01
$\bar{p}$	$\bar{s}$	.99

P	B	$\Theta_{B\|P}$
p	b	.64
p	$\bar{b}$	.36
$\bar{p}$	b	.106
$\bar{p}$	$\bar{b}$	.894

P	U	$\Theta_{U\|P}$
p	u	.73
p	$\bar{u}$	.27
$\bar{p}$	u	.107
$\bar{p}$	$\bar{u}$	.893

Table 6.1: The CPTs of the naive Bayes network Pregnancy.

U	B	S	$Pr(p \mid u, b, s)$	$\geq .9$?
u	b	s	.999	yes
u	b	$\bar{s}$	.966	yes
u	$\bar{b}$	s	.999	yes
u	$\bar{b}$	$\bar{s}$	.65	no
$\bar{u}$	b	s	.999	yes
$\bar{u}$	b	$\bar{s}$	.552	no
$\bar{u}$	$\bar{b}$	s	.987	yes
$\bar{u}$	$\bar{b}$	$\bar{s}$	.076	no

Table 6.2: The classifier induced by the naive Bayes network Pregnancy with probability threshold .9.

a variable C, called the *class variable*, and a set of variables $\mathbf{E} = \{E_1, \ldots, E_n\}$ known as the *attributes*.[1] Each instantiation $\mathbf{e}$ of $\mathbf{E}$ is known as an *instance*. Moreover, given some *probability threshold* p, the Bayesian network can be viewed as inducing the function F, which maps each instance $\mathbf{e}$ into $\{0, 1\}$ as follows: $F(\mathbf{e}) = 1$ if $Pr(c \mid \mathbf{e}) \geq p$, and $F(\mathbf{e}) = 0$ otherwise. This function F is called a *Bayesian network classifier* [FGG97, GR01].

In this chapter, we will look at how decisions induced by Bayesian networks are affected by parameter changes, and provide some results using a principled approach for reasoning about Bayesian network classifiers [CD03b]. In particular, we are interested in answering the following types of questions:

- *Given two Bayesian networks $\mathcal{B}$ and $\mathcal{B}'$, do they induce the same classifier? If not, which, and how many, instances do they disagree on?* Instead of being interested in the changes in query values from $\mathcal{B}$ to $\mathcal{B}'$, we now focus on whether the output of any instance is different between the classifiers induced by networks $\mathcal{B}$ and $\mathcal{B}'$.

- *Given a Bayesian network $\mathcal{B}$, what are the allowable changes in some CPT in $\mathcal{B}$ that will not change the current classifier induced?* Instead of measuring robustness by bounding changes in query values due to the parameter changes, we now define it as such that the output of every instance remains the same after applying the parameter changes.

These questions can be answered by enumerating all instances $\mathbf{e}$ explicitly. However, this brute-force approach is often infeasible given the exponential number of instances. Instead, we propose to build a tractable logical representation

[1] The other variables in the network are called hidden or intermediate variables. They are not mentioned and are used for modeling purposes.

of the Bayesian network classifier, which allows us to answer the above questions in time polynomial in the size of the constructed representation.

The specific logical representation we propose is that of ordered decision diagrams (ODDs), which are known to be tractable. Although our long-term objective is to construct ODDs (or other logical representations) for general Bayesian network classifiers, we will focus on the simplest, yet very common, class of naive Bayes classifiers [DH73, LIT92], which are induced by naive Bayes networks.

6.1 Sensitivity Analysis of Naive Bayes Classifiers

A *naive Bayes classifier* is induced by a *naive Bayes network*, which contains the class variable C as the root, with the attributes $\mathbf{E} = \{E_1, \ldots, E_n\}$ as its children. No other nodes or edges exist in the network. For example, the Bayesian network $\mathsf{Pregnancy}$, whose structure is shown in Figure 6.1, is a naive Bayes network.

To classify an instance $\mathbf{e} = \{e_1, \ldots, e_n\}$, we need to compute the posterior probability $Pr(c \mid \mathbf{e})$. However, for ease of computation, we will compute this probability in log-odds space, where its log-odds is $\ln O(c \mid \mathbf{e}) = Pr(c \mid \mathbf{e})/(1 - Pr(c \mid \mathbf{e}))$. Given a naive Bayes network $\mathcal{B}$ where C is binary,[2] if χ is an instantiation of a subset of $\mathbf{E}$, and e_i is a value of an un-instantiated attribute E_i, we have:

$$\ln O(c \mid \chi, e_i) = \ln O(c \mid \chi) + W(e_i, c), \tag{6.1}$$

where $W(e_i, c)$ is the *weight of evidence* e_i in favor of c:

$$W(e_i, c) \overset{\text{def}}{=} \ln \frac{Pr(e_i \mid c)}{Pr(e_i \mid \bar{c})}.$$

[2] If C is non-binary, we can group all values $c^* \neq c$ into the pseudo-value $\bar{c}$.

We can now compute the value $\ln O(c \mid \mathbf{e})$ using Equation 6.1:

$$\ln O(c \mid \mathbf{e}) = \ln O(c) + \sum_{i=1}^{n} W(e_i, c). \tag{6.2}$$

We call the value $\ln O(c)$ the *prior log-odds* of $\mathcal{B}$. Therefore, a naive Bayes network is a tuple $\mathcal{B} = (C, \{E_1, \ldots, E_n\}, \ln O(c), \{W(e_i, c)\})$.

Given a probability threshold p, the *log-odds threshold* is $\sigma = \ln(p/(1-p))$, such that $Pr(c \mid \mathbf{e}) \geq p$ iff $\ln O(c \mid \mathbf{e}) \geq \sigma$. We now formally define the *naive Bayes classifier* induced by a naive Bayes network $\mathcal{B}$ with log-odds threshold σ.

Definition 6.1 *The naive Bayes classifier $F_{\mathcal{B}}^{\sigma}$ induced by a naive Bayes network $\mathcal{B}$ with log-odds threshold σ is defined as follows:*

$$F_{\mathcal{B}}^{\sigma}(\mathbf{e}) \overset{\text{def}}{=} \begin{cases} 1 & \textit{if } \ln O(c \mid \mathbf{e}) \geq \sigma; \\ 0 & \textit{otherwise.} \end{cases}$$

We now show an example naive Bayes classifier.

Example 6.1 *In the naive Bayes network* Pregnancy*, denoted by $\mathcal{B}$, P is the class variable, and the variables $\{U, B, S\}$ are the attributes. Given the log-odds threshold $\sigma = \ln(.9/.1) = 2.20$, the naive Bayes classifier $F_{\mathcal{B}}^{\sigma}$ induced determines whether the probability of pregnancy given an instance (the results of the three tests) is no less than .9, and is shown in Table 6.2.*

We now discuss the following key question: how much change can we apply to a CPT in the network $\mathcal{B}$ without changing the current classifier induced, $F_{\mathcal{B}}^{\sigma}$?

6.1.1 Changing the Prior Log-Odds

We first look at the case where we change only the CPT of the class variable C, and obtain a new naive Bayes network $\mathcal{B}'$. This is equivalent to changing only the prior log-odds $\ln O(c)$ to the new value $\ln O'(c)$. The question is, are $F^g_{\mathcal{B}}$ and $F^g_{\mathcal{B}'}$ the same classifier? This obviously depends on the amount of change in the prior log-odds. However, the following theorem states that the amount of allowable change in the prior log-odds can be determined precisely once we know the following two values, known as *margins*:

- The minimum value of $\ln O(c \mid \mathbf{e})$ attained by any positive instance $\mathbf{e}$:

$$\sigma^1 \overset{\text{def}}{=} \min_{\mathbf{e}:F^g_{\mathcal{B}}(\mathbf{e})=1} \ln O(c \mid \mathbf{e}); \tag{6.3}$$

- The maximum value of $\ln O(c \mid \mathbf{e})$ attained by any negative instance $\mathbf{e}$:

$$\sigma^0 \overset{\text{def}}{=} \max_{\mathbf{e}:F^g_{\mathcal{B}}(\mathbf{e})=0} \ln O(c \mid \mathbf{e}). \tag{6.4}$$

Theorem 6.1 *Let $\mathcal{B}'$ be a naive Bayes network obtained from $\mathcal{B}$ by changing the CPT of the class variable C, such that the prior log-odds changes from $\ln O(c)$ to $\ln O'(c)$. The classifiers $F^g_{\mathcal{B}}$ and $F^g_{\mathcal{B}'}$ are the same iff $\ln O'(c) \in [\ln O(c) + \sigma - \sigma^1, \ln O(c) + \sigma - \sigma^0)$, where σ^1 and σ^0 are given by Equations 6.3 and 6.4 respectively.*[3]

As an example, consider the naive Bayes classifier $F^g_{\mathcal{B}}$ given in Example 6.1. By enumerating all instances explicitly, we find that $\sigma^1 = 3.33$ and $\sigma^0 = .619$. Therefore, any change in the CPT of variable P will keep the classifier $F^g_{\mathcal{B}}$ unchanged as long as $\ln O'(p) \in [.772, 3.48)$, according to Theorem 6.1. Therefore, the classifier will remain unchanged as long as the new prior probability of pregnancy falls in the interval $[.684, .970)$. Note that the current value of this prior probability is .87, showing that we can apply a significant change to this prior probability without changing the classifier.

[3] *If there are no positive instances, $\sigma^1 = \infty$, and if there are no negative instances, $\sigma^0 = -\infty$.*

We will call the interval given in Theorem 6.1 the *equivalence interval* of the naive Bayes classifier $F^g_{\mathcal{B}}$, denoted by $I(F^g_{\mathcal{B}})$. Later, we will show how we can find the equivalence interval without enumerating all instances explicitly.

The maximum number of distinct naive Bayes classifiers (including the current classifier) that can be induced by changing the prior log-odds can also be counted, as shown by the following theorem.

Theorem 6.2 *The number of distinct naive Bayes classifiers (including the current classifier) that can be induced by changing the prior log-odds is at most* $\|\mathbf{E}\| + 1$*, where* $\|\mathbf{E}\|$ *is the number of instances.*[4]

If all attributes are binary, this number is $2^n + 1$. For the naive Bayes network Pregnancy, 9 different classifiers can be induced by changing the CPT of variable P, according to Theorem 6.2. However, note that the total number of distinct Boolean functions is $2^{2^n} = 256$ in this case.

To further illustrate Theorems 6.1 and 6.2, we now rephrase them using the mathematical notion of *equivalence class.* Given naive Bayes network $\mathcal{B} = (C, \{E_1, \ldots, E_n\}, \ln O(c), \{W(e_i, c)\})$, we define the set S as containing exactly all naive Bayes networks $\mathcal{B}'$ obtained from $\mathcal{B}$ by changing only the prior log-odds (including $\mathcal{B}$), i.e., $S = \{\mathcal{B}' : \mathcal{B}' = (C, \{E_1, \ldots, E_n\}, \ln O'(c), \{W(e_i, c)\})\}$. The equivalence class $[\mathcal{B}] \subset S$ is defined such that $\mathcal{B}' \in [\mathcal{B}]$ iff the classifiers $F^g_{\mathcal{B}}$ and $F^g_{\mathcal{B}'}$ are the same. Theorem 6.1 allows us to test for $\mathcal{B}' \in [\mathcal{B}]$ by verifying if $\ln O'(c) \in I(F^g_{\mathcal{B}})$,[5] while Theorem 6.2 gives us the number of equivalence classes that form the partition of S.[6]

[4] *In general, if* $\|X_i\|$ *is the cardinality of variable* X_i*, i.e., the number of possible values of* X_i*,* $\|X_1, \ldots, X_k\| = \prod_{i=1}^{k} \|X_i\|$ *is the number of instantiations of variables* $X_1, \ldots, X_k$*.*

[5] If $\mathcal{B}' \in [\mathcal{B}]$, we have $[\mathcal{B}'] = [\mathcal{B}]$ and $I(F^g_{\mathcal{B}'}) = I(F^g_{\mathcal{B}})$ by the definition of equivalence class.

[6] We note that both theorems hold not only for naive Bayes classifiers, but more generally for any Bayesian network classifier in which the attributes $\mathbf{E}$ are all descendants of the class variable C.

We close this section by emphasizing that Theorems 6.1 and 6.2 will be crucial to our algorithm that converts a naive Bayes classifier into an ordered decision diagram.

6.1.2 Changing the Weights of Evidence of an Attribute

We now look at the case where we change only the CPT of attribute E_i, and obtain the new naive Bayes network $\mathcal{B}'$. This is equivalent to changing only the weight of evidence e_i from $W(e_i, c)$ to the new value $W'(e_i, c)$ for every value e_i of E_i. The question is, are $F^g_{\mathcal{B}}$ and $F^g_{\mathcal{B}'}$ the same classifier? The following theorem states that this can be determined once we know the following two values for every e_i:

- The minimum value of $\ln O(c \mid \mathbf{e})$ attained by any positive instance $\mathbf{e}$ such that $e_i \in \mathbf{e}$:
$$\sigma^1_{e_i} \overset{\text{def}}{=} \min_{\mathbf{e}: e_i \in \mathbf{e}, F^g_{\mathcal{B}}(\mathbf{e})=1} \ln O(c \mid \mathbf{e}); \tag{6.5}$$
- The maximum value of $\ln O(c \mid \mathbf{e})$ attained by any negative instance $\mathbf{e}$ such that $e_i \in \mathbf{e}$:
$$\sigma^0_{e_i} \overset{\text{def}}{=} \max_{\mathbf{e}: e_i \in \mathbf{e}, F^g_{\mathcal{B}}(\mathbf{e})=0} \ln O(c \mid \mathbf{e}). \tag{6.6}$$

Theorem 6.3 *Let $\mathcal{B}'$ be a naive Bayes network obtained from $\mathcal{B}$ by changing the CPT of attribute E_i, such that the weight of evidence e_i changes from $W(e_i, c)$ to $W'(e_i, c)$ for every value e_i of E_i. The classifiers $F^g_{\mathcal{B}}$ and $F^g_{\mathcal{B}'}$ are the same iff for every e_i, $W'(e_i, c) \in [W(e_i, c) + \sigma - \sigma^1_{e_i}, W(e_i, c) + \sigma - \sigma^0_{e_i})$, where $\sigma^1_{e_i}$ and $\sigma^0_{e_i}$ are given by Equations 6.5 and 6.6 respectively.*

As an example, consider again the naive Bayes classifier $F^g_{\mathcal{B}}$ given in Example 6.1. If we would like to change the CPT of attribute U without changing the classifier, the allowable new weights of evidence are $W'(u,p) \in [.791, 3.50)$ and $W'(\bar{u},p) \in [-3.30, .791)$, according to Theorem 6.3. For example, even if we improve the reliability of the urine test by changing the false-negative rate from .27 to .1 and the false-positive rate from .107 to .05, the classifier will still remain unchanged.

The maximum number of distinct naive Bayes classifiers (including the current classifier) that can be induced by changing all weights of evidence of attribute E_i can also be counted, as shown by the following theorem.

Theorem 6.4 *The number of distinct naive Bayes classifiers (including the current classifier) that can be induced by changing all weights of evidence of attribute* E_i *is at most* $(a+1)^b - \lfloor a/2 \rfloor^b - \lceil a/2 \rceil^b$*, where* $a = \|\mathbf{E} - E_i\|$ *and* $b = \|E_i\|$*.*

If all attributes are binary, this number is $2^{2n-3} + 2^n + 1$. For the naive Bayes network Pregnancy, at most 17 different classifiers can be induced by changing the CPT of attribute U, according to Theorem 6.4.

6.2 Converting Naive Bayes Classifiers into Logical Representations

In this section, we will introduce an algorithm that converts a naive Bayes classifier into an *ordered decision diagram (ODD)*, defined as follows.

Definition 6.2 *An ordered decision diagram (ODD), with respect to variable order* $(E_1, \ldots, E_n)$*, is a rooted, directed, acyclic graph, with two sinks labeled with* 1 *and* 0*, called* 1-SINK *and* 0-SINK *respectively. Every node (except the sinks) in the ODD is labeled with a variable* E_i*, and for every value* e_i *of* E_i*, there is an*

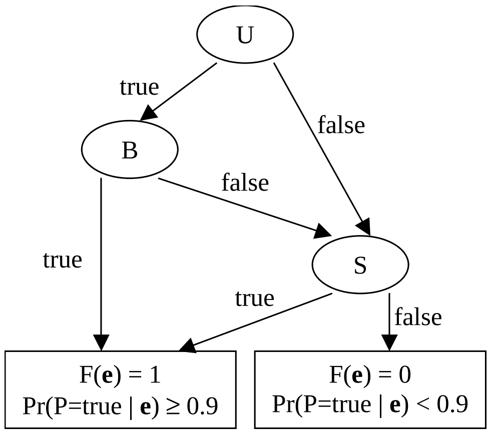


Figure 6.2: An ODD that represents the classifier induced by the naive Bayes network Pregnancy with probability threshold .9, with respect to variable order (U, B, S).

edge labeled with e_i exiting this node. Finally, a node is labeled with E_i and its child is labeled with E_j only if $j > i$.

An ODD represents a classifier F variables $\mathbf{E} = \{E_1, \ldots, E_n\}$ as follows. Given an instantiation $\mathbf{e} = \{e_1, \ldots, e_n\}$, we traverse the ODD starting at the root. At a node labeled with E_i, we go to the child pointed by the edge labeled with $e_i \in \mathbf{e}$. If we reach 1-SINK, we have $F(\mathbf{e}) = 1$, and if we reach 0-SINK, we have $F(\mathbf{e}) = 0$. Figure 6.2 shows the ODD that represents the naive Bayes classifier $F^{\sigma}_{\mathcal{B}}$ given in Example 6.1, with respect to variable order (U, B, S). If all the variables in the ODD are binary, as in this case, it is called an *ordered binary decision diagram (OBDD)* [Bry86], a well-researched representation of boolean functions. As we will discuss later in this section, the tractability of the ODD representation allows us to answer the questions we posed earlier in time polynomial in the size of the ODD.

6.2.1 Algorithm and Theoretical Results

Suppose now we are given a naive Bayes classifier $F^{\sigma}_{\mathcal{B}}$, which is induced by the naive Bayes network $\mathcal{B} = (C, \{E_1, \ldots, E_n\}, \ln O(c), \{W(e_i, c)\})$ with log-odds threshold σ. Our goal is to build an ODD $\mathcal{D}$ that represents $F^{\sigma}_{\mathcal{B}}$, with respect to attribute order $(E_1, \ldots, E_n)$. Before we state our algorithm and its complexity, we first explain two key observations underlying our algorithm.

The first key observation is as follows. Given an instantiation $\chi = e_1, \ldots, e_k$ of the first k attributes $E_1, \ldots, E_k$, we assume the node reached by the path χ from the root of ODD $\mathcal{D}$ is the root of a sub-ODD denoted by $\mathcal{D}_{\chi}$. We also note that a new naive Bayes network $\mathcal{B}_{\chi} = (C, \{E_{k+1}, \ldots, E_n\}, \ln O(c \mid \chi), \{W(e_i, c)\})$ can be obtained by removing attributes $E_1, \ldots, E_k$ from $\mathcal{B}$, and updating the prior log-odds to $\ln O(c \mid \chi)$. Note that the output of the naive Bayes classifier $F^{\sigma}_{\mathcal{B}}$ given instance $\mathbf{e} = \chi, e_{k+1}, \ldots, e_n$ can now be obtained from the new naive Bayes classifier $F^{\sigma}_{\mathcal{B}_{\chi}}$, since from Equation 6.2, we have:

$$\begin{aligned} \ln O(c \mid \mathbf{e}) &= \ln O(c) + \sum_{i=1}^{n} W(e_i, c) \\ &= \ln O(c \mid \chi) + \sum_{i=k+1}^{n} W(e_i, c). \end{aligned}$$

Therefore, $F^{\sigma}_{\mathcal{B}}(\mathbf{e}) = F^{\sigma}_{\mathcal{B}_{\chi}}(e_{k+1}, \ldots, e_n)$, and consequently, the sub-ODD $\mathcal{D}_{\chi}$ represents $F^{\sigma}_{\mathcal{B}_{\chi}}$.

The second key observation is based on Theorem 6.1. If χ^* is another instantiation of attributes $E_1, \ldots, E_k$, the path χ^* reaches the root of the sub-ODD $\mathcal{D}_{\chi^*}$, which represents the naive Bayes classifier $F^{\sigma}_{\mathcal{B}_{\chi^*}}$, where the naive Bayes network $\mathcal{B}_{\chi^*} = (C, \{E_{k+1}, \ldots, E_n\}, \ln O(c \mid \chi^*), \{W(e_i, c)\})$ can be obtained by removing attributes $E_1, \ldots, E_k$ from $\mathcal{B}$, and updating the prior log-odds to $\ln O(c \mid \chi^*)$. Because $\mathcal{B}_{\chi}$ and $\mathcal{B}_{\chi^*}$ differ only on their prior log-odds, from Theorem 6.1, the classifiers $F^{\sigma}_{\mathcal{B}_{\chi}}$ and $F^{\sigma}_{\mathcal{B}_{\chi^*}}$ are the same iff $\ln O(c \mid \chi^*) \in I(F^{\sigma}_{\mathcal{B}_{\chi}})$. If this is true, the two sub-ODDs $\mathcal{D}_\chi$ and $\mathcal{D}_{\chi*}$ are *isomorphic*, and we can build the ODD $\mathcal{D}$ such that the paths χ and χ^* reach the same node. This allows us to save space and time when building the ODD $\mathcal{D}$. The following theorem shows how we can compute the equivalence interval $I(F^\sigma_{\mathcal{B}_\chi})$ inductively, as it is key to our algorithm.

Theorem 6.5 *If χ is an instantiation of attributes $E_1, \ldots, E_k$, the equivalence interval $I(F^\sigma_{\mathcal{B}_\chi})$ can be computed if we know the equivalence interval $I(F^\sigma_{\mathcal{B}_{\chi,e_{k+1}}})$ for every value e_{k+1} of E_{k+1}:*

$$I(F^\sigma_{\mathcal{B}_\chi}) = \bigcap_{e_{k+1}} \Big\{x : x + W(e_{k+1}, c) \in I(F^\sigma_{\mathcal{B}_{\chi,e_{k+1}}})\Big\}.$$

In our algorithm, we associate the node *node* with the equivalence interval $I[\mathit{node}] = I(F^\sigma_{\mathcal{B}_\chi})$ if *node* is reached by path χ. Theorem 6.5 states that we can compute this equivalence interval if we are given the equivalence interval of every child of *node*. Therefore, we can compute the equivalence interval of every node in the ODD $\mathcal{D}$ inductively, with the end conditions $I[1\text{-}\textsc{sink}] = [\sigma, \infty)$ and $I[0\text{-}\textsc{sink}] = (-\infty, \sigma)$.

To identify isomorphic sub-ODDs, we employ $n + 1$ caches in our algorithm, one for each $k = 0, \ldots, n$, where the k-th cache stores nodes at depth k. In each cache, nodes are indexed by their equivalence intervals. Given some path χ of length k, we check if there already exists some *node* in the k-th cache where $\ln O(c \mid \chi) \in I[\mathit{node}]$. If this is true, the ODD $\mathcal{D}$ will be built such that the path χ also reaches *node*.

Algorithm 6.1 shows the procedure $\textsc{Build-ODD}(\mathcal{B}, \sigma)$, which returns the root of the ODD $\mathcal{D}$ that represents the naive Bayes classifier $F^\sigma_{\mathcal{B}}$, with respect to attribute order $(E_1, \ldots, E_n)$. After initialization of the sinks, the ODD $\mathcal{D}$ is built recursively by calling the procedure $\textsc{Build-sub-ODD}(k, v)$, shown in Algo-

Algorithm 6.1 Build-ODD($\mathcal{B}, \sigma$): returns the root of the ODD $\mathcal{D}$ that represents the naive Bayes classifier $F^{\sigma}_{\mathcal{B}}$, with respect to attribute order $(E_1, \ldots, E_n)$, where $\mathcal{B} = (C, \{E_1, \ldots, E_n\}, \ln O(c), \{W(e_i, c)\})$ is a naive Bayes network, and σ is the log-odds threshold.

```
1-sink ← Create-Node()
$I$[1-sink] ← $[\sigma, \infty)$
Store-in-Cache($n$, 1-sink)
0-sink ← Create-Node()
$I$[0-sink] ← $(-\infty, \sigma)$
Store-in-Cache($n$, 0-sink)
return Build-sub-ODD($0, \ln O(c)$)
```

rithm 6.2.[7] This procedure returns the root of the sub-ODD $\mathcal{D}_{\chi}$ that represents the naive Bayes classifier $F^{\sigma}_{\mathcal{B}_{\chi}}$, where χ is an instantiation of $E_1, \ldots, E_k$, and $v = \ln O(c \mid \chi)$ is the prior log-odds of $\mathcal{B}_{\chi}$.

The following theorem gives us a theoretical upper bound on the number of nodes in the ODD $\mathcal{D}$ and the time complexity of Algorithm 6.1, which can be proved using Theorem 6.2.

Theorem 6.6 *The number of nodes in the ODD $\mathcal{D}$ built by Algorithm 6.1 is at most:*

$$\sum_{k=0}^{n} \min\{\|E_1, \ldots, E_k\|, \|E_{k+1}, \ldots, E_n\| + 1\}.$$

[7] In Algorithm 6.2, we define the following procedures as: Create-Node() returns a newly-created node; Find-in-Cache(j, x) returns *node* in the j-th cache where $x \in I[node]$, or Nil if no such node exists; Store-in-Cache($j, node$) stores *node* in the j-th cache, indexed by $I[node]$; Add-Child($node, child, label$) adds *child* as a child of *node*, with *label* being the label of the edge from *node* to *child*; Offset(I, d) returns $\{x : x - d \in I\}$, i.e., the interval I moved by a displacement of d. We also assume that the weights of evidence $\{W(e_i, c)\}$ can be accessed globally.

Algorithm 6.2 Build-sub-ODD(k, v): returns the root of the sub-ODD $\mathcal{D}_\chi$ that represents the naive Bayes classifier $F^\sigma_{\mathcal{B}_\chi}$, where χ is an instantiation of $E_1, \ldots, E_k$, and $v = \ln O(c \mid \chi)$ is the prior log-odds of $\mathcal{B}_\chi$.

```
node ← Create-Node()
I[node] ← $(-\infty, \infty)$
for all values $e_{k+1}$ of $E_{k+1}$ do
    $v_{child} \leftarrow v + W(e_{k+1}, c)$
    child ← Find-in-Cache($k+1, v_{child}$)
    if child = Nil then
        child ← Build-sub-ODD($k+1, v_{child}$)
    end if
    Add-Child(node, child, $e_{k+1}$)
    I[node] ← I[node] $\cap$ Offset(I[child], $-W(e_{k+1}, c)$)
end for
Store-in-Cache($k$, node)
return node
```

If all attributes have at most b values, the space complexity is $O(b^{n/2})$. Moreover, the time complexity of Algorithm 6.1 is $O(nb^{n/2})$.

Theorem 6.6 means that we are able to convert a naive Bayes classifier with n attributes into an ODD in space and time that are no more than exponential in $n/2$. This is significant both theoretically and practically compared to the brute-force method whose space and time complexity are exponential in n. Therefore, classifiers with up to 50 attributes can be handled in practice. However, as we will show in our experimental results, the actual space and time required by our algorithm are usually much less than the theoretical upper bound, showing promise for classifiers with even more attributes. We also note that the actual number of nodes in the ODD depends on the attribute order, and later we will suggest some ordering heuristics that perform well in practice.

Finally, in the case where the class variable C is non-binary, we may be interested in mapping an instance $\mathbf{e}$ to the value of C that is the most likely given $\mathbf{e}$. To handle this problem, we can build a series of ODDs that classify which of any two values of C are more likely. For example, if C can take on three values $\{1, 2, 3\}$, we can build three ODDs, the first classifying whether $C = 1$ is more likely than $C = 2$ given $\mathbf{e}$, the second classifying whether $C = 1$ is more likely than $C = 3$ given $\mathbf{e}$, and the third classifying whether $C = 2$ is more likely than $C = 3$ given $\mathbf{e}$. The combination of these outputs of the ODDs gives us the most likely value of C given $\mathbf{e}$.

6.2.2 Experimental Results

We now show experimental results of building ODDs for both random and real-world naive Bayes classifiers using our algorithm.

In the first part of our experiment, we build ODDs that represent random naive Bayes classifiers with binary attributes $\mathbf{E} = \{E_1, \ldots, E_n\}$, for different values of n. The prior log-odds and the weights of evidence of the naive Bayes networks take on random values, which are translated to the log-odds space from the uniform probability space. The log-odds threshold is set at $\sigma = 0$, meaning $F^{g}_{\mathcal{B}}(\mathbf{e}) = 1$ iff $Pr(c \mid \mathbf{e}) \geq 0.5$. We generate 100 random classifiers for each n, and the results are displayed in Table 6.3. The second column shows the number of instances, i.e., $\|\mathbf{E}\| = 2^n$, while the third column shows the theoretical upper bound on the number of nodes in the ODDs given in Theorem 6.6. The fourth column shows the average number of nodes in the ODDs built using 100 random attribute orders. As we can see, the number of nodes is on average about two-

n	$\|\mathbf{E}\|$	Bound	Random	Desc.	Asc.
10	1024	99	64	56	51
15	32768	518	347	270	263
20	1×10^6	3080	2032	1541	1531
25	3×10^7	16395	11968	8753	8740
30	1×10^9	98317	66160	50116	50100

Table 6.3: Experimental results of building ODDs that represent random naive Bayes classifiers.

thirds of the bound. We also sort the attributes by the absolute differences of the weights of evidence, i.e., $|W(e_i, c) - W(\bar{e_i}, c)|$, where a larger absolute difference means the attribute E_i has more evidential impact on the posterior probabilities $Pr(c \mid \mathbf{e})$. The average sizes of the ODDs built using the attribute orders with descending and ascending orders of evidential impact are shown in the fifth and sixth columns respectively. In either case, the number of nodes is on average about half of the bound, an improvement over using random attribute orders.

In the second part of our experiment, we build ODDs that represent real-world naive Bayes classifiers. The naive Bayes networks are constructed by learning data obtained from the UCI Machine Learning Repository [HBM]. The log-odds threshold σ is also set at 0. The results are displayed in Table 6.4 for several networks. The second column shows n, the number of attributes in the classifier, while the third column shows $\|\mathbf{E}\|$, the number of instances. Note that many attributes in the networks are multi-valued. The fourth column shows the theoretical upper bound on the number of nodes in the ODDs given in Theorem 6.6.[8]

[8] Because the bound varies with the attribute order if the attributes do not have the same cardinality (number of values), the bound displayed here is computed using the attribute order where the ODD with the fewest number of nodes is obtained.

Network	n	$\|\mathbf{E}\|$	Bound	Best
Tic-tac-toe	9	19683	247	58
Votes	16	65536	774	396
Spect	22	4×10^6	6153	609
Breast-cancer-w	9	1×10^9	21117	4405
Hepatitis	19	2×10^{10}	46794	9644
Kr-vs-kp	36	1×10^{11}	917488	59905
Mushroom	22	1×10^{14}	1×10^8	43638

Table 6.4: Experimental results of building ODDs that represent real-world naive Bayes classifiers.

For each classifier, we build ODDs using 100 random attribute orders, plus the attribute orders with descending and ascending orders of evidential impact,[9] and the final column shows the fewest number of nodes among the ODDs built.

The results we produce are very satisfactory, as for many of these classifiers, there is an intractable number of instances, yet we are able to build ODDs with at most 60000 nodes in the best cases. The number of nodes actually created are also often much less than the theoretical upper bound, even with a random attribute order, since many of the CPTs in the classifiers are sparse, i.e., filled with parameters whose values are 0 or 1. An example is naive Bayes network Mushroom. The time to run our algorithm is also relatively short, as it takes less than five seconds to build an ODD with about 60000 nodes.

We also note that although the sizes of the ODDs vary with the attribute orders, experimentally we find that for each classifier, the size of the ODD in the

[9] For a multi-valued attribute E_i, we use the distance measure defined in Definition 4.1, which computes the difference between the maximum and minimum weights of evidence, i.e., $\max_{e_i} W(e_i, c) - \min_{e_i} W(e_i, c)$.

worst case is at most about twice the size of the ODD in the best case. Therefore, even with a random attribute order, we are able to build ODDs of reasonable size. In the future, we would like to explore other ordering heuristics. Currently, the method of sorting the attributes by ascending order of evidential impact gives us the best results in many, but not all cases.

Finally, our algorithm can also be augmented, without affecting its complexity, to generate *reduced ODDs* [Bry86], which eliminate nodes whose outgoing edges all point to the same child. However, we find that after including this reduction step, the sizes of the ODDs decrease by less than 1% in many of the cases, and less than 5% in most of the cases. Therefore, we do not include this in our algorithm for simplicity of exposition.

6.2.3 Applications

After presenting the theoretical and experimental results of our algorithm of converting a naive Bayes classifier into an ODD, we can discuss the variety of applications enabled by the construction of such an ODD.

We first point out that ODDs are tractable logical representations in the sense that they permit a number of operations on the functions they represent in time polynomial in their sizes, even though such operations are intractable in general [DM02]. In particular, given two ODDs $\mathcal{D}$ and $\mathcal{D}'$ with respect to the same variable order, with sizes s and s' respectively, we can perform the following operations:

- Testing whether $\mathcal{D}$ and $\mathcal{D}'$ are equivalent in $O(s + s')$ time;
- Counting the number of instances mapped to 1 or 0 by $\mathcal{D}$ (positive or negative instances) in $O(s)$ time;

- Testing whether all positive or negative instances of $\mathcal{D}$ satisfy some conjunction or disjunction of features (attribute-value pairs) in $O(s)$ time;
- Conjoining or disjoining $\mathcal{D}$ and $\mathcal{D}'$ in $O(s \cdot s')$ time.

All of the above operations on ODDs are supported by standard packages such as the CU Decision Diagram Package [Som]. These operations, plus many others, can be combined to answer queries. For example, if we want to know the number of positive instances in the intersection of two classifiers, we can first conjoin the two classifiers and then perform a count operation.

The equivalence operation is one of the most important operations because if two Bayesian network classifiers are shown to be equivalent, we can use either network to model the domain for the purpose of classifying instances. This is helpful if we want to test whether simplifying a Bayesian network, such as rounding off the parameters, changes the classification of any instance. We can also check whether adding another attribute will improve the classification ability of the network. For example, for the naive Bayes classifier given in Example 6.1, we may want to know if adding a particular new test will be beneficial in detecting pregnancy, i.e., given any set of results from the current tests, whether applying this new test may potentially support the presence or absence of pregnancy.

Moreover, we can use the equivalence operation to see whether the classification outputs given by networks produced from different learning algorithms are the same when run over the same data set, as the networks will differ on their parameters and possibly their structures. We can also determine whether adding some data samples will change the behavior of the classifier produced by any learning algorithm.

Another application of converting a naive Bayes classifier $F_{\mathcal{B}}^{g}$ into an ODD $\mathcal{D}$

is that we can efficiently find the intervals given in Theorems 6.1 and 6.3 as a side effect of our algorithm. This is due to the computation of the equivalence interval of every node in the ODD $\mathcal{D}$ by our algorithm. For example, we note that the equivalence interval $I(F^{\sigma}_{\mathcal{B}})$ given in Theorem 6.1, which contains the allowable prior log-odds that will keep the classifier unchanged from $F^{\sigma}_{\mathcal{B}}$, is equal to $I[root]$ if *root* is the root of the ODD $\mathcal{D}$.

We can also find the intervals given in Theorem 6.3, which contain the allowable weights of evidence of attribute E_i that will keep the classifier unchanged from $F^{\sigma}_{\mathcal{B}}$. However, in order to find these intervals, it is required that E_i must come first in the attribute order used to build the ODD $\mathcal{D}$. In this case, if the node $child_{e_i}$ is the child of the root of the ODD $\mathcal{D}$ reached by the edge labeled e_i, the equivalence interval $I[child_{e_i}] = I(F^{\sigma}_{\mathcal{B}_{e_i}})$ contains the allowable weight of evidence $W(e_i, c)$ that will keep the classifier unchanged from $F^{\sigma}_{\mathcal{B}}$.

Therefore, instead of enumerating all instances explicitly, which requires space and time complexity exponential in n, where n is the number of attributes, we can find the intervals given in Theorems 6.1 and 6.3 by converting the naive Bayes classifier into an ODD using our algorithm. The asymptotic space and time complexity are exponential only in $n/2$, but as shown by our experimental results, the actual time and space required are often much less.

CHAPTER 7

Sensitivity Analysis of Markov Networks

In earlier chapters, we focused our study of sensitivity analysis on the domain of Bayesian networks. We observed that the joint probability distribution induced by a Bayesian network corresponds to a multi-linear function, providing efficient methods for computing the derivative of a query with respect to a single Bayesian network parameter, with results summarized as follows:

- *What is the necessary parameter change we need to apply such that a given query constraint is satisfied?* We presented efficient procedures that find minimum changes in single parameters in Chapter 2, and approximate optimal changes in multiple parameters in Chapter 5.

- *What is the bound on the change in some query value if we apply an arbitrary parameter change?* In Chapter 3, we showed that the log-odds change in any conditional query in a Bayesian network is bounded by the log-odds change in any single parameter.

- *What is the bound on the difference between some query value induced by two networks that have the same structure but different parameter values?* We proposed a distance measure in Chapter 4 that allows us to bound the difference between the query values under two distributions. Based on this distance measure, and given two Bayesian networks that differ on only a single CPT, the global distance measure between the distributions induced by the networks can be computed from the local distance measure between the CPTs, thereby allowing us to provide a bound on the difference between the query values. Moreover, if we are given multiple CPTs where the variables are disjoint, the global distance measure can still be computed as the sum of the local distance measures between the individual CPTs.

In this chapter, we will address these key questions, but in the context of Markov networks [CD05c]. The main topic of interest is the extent to which these promising results hold for Markov networks as well. There is indeed a key difference between Bayesian networks and Markov networks that appears to suggest a lack of parallels in this case: whereas the joint probability distribution induced by a Bayesian network corresponds to a multi-linear function, the joint probability distribution induced by a Markov network corresponds to a ratio of multi-linear functions. As it turns out, however, a conditional probability has the same functional form of being a ratio of multi-linear functions, in both Bayesian and Markov networks. This similarity turns out to be the key factor here, allowing us to derive similarly efficient results for Markov networks. This is greatly beneficial because we can answer the previous three questions in the context of Markov networks as well, with the same computational complexity. For example, we can go through each parameter in a Markov network, compute the minimum single parameter changes necessary to enforce a query constraint, and find the one that disturbs the network the least. Alternatively, we can change all parameters in a single clique table, and find the change that minimizes network disturbance. Afterwards, we can compute a bound on any query change using the distance measure incurred by the parameter change.

Our results, however, point to a main semantical difference between Bayesian networks and Markov networks, relating to how we should quantify and measure parameter changes. That is, how should we quantify a parameter change from .3 to .4? In a Bayesian network, parameters are interpreted as conditional probabilities, and the measure that quantifies the change is the *relative odds change* in the parameter value. This means query values are much more sensitive to changes in extreme probability values, whether close to 0 or 1. On the other hand, in a Markov network, parameters are interpreted as compatibility ratios, and the measure that quantifies the change is the *relative change* in the parameter value. This difference stems from how the parameters in the two models are interpreted and will be explained in more depth later.

7.1 Tuning Markov Network Parameters

We now answer the following question in the context of Markov networks: what is the necessary change we can apply to certain parameter(s) such that a query constraint is satisfied, such as $Pr(y \mid \mathbf{e}) \geq \kappa$?

We first define the function $\phi(\mathbf{e})$ as the sum of the potentials of $\mathbf{x}$ that are consistent with $\mathbf{e}$:

$$\phi(\mathbf{e}) \overset{\text{def}}{=} \sum_{\mathbf{x} \sim \mathbf{e}} \psi(\mathbf{x}) = \sum_{\mathbf{x} \sim \mathbf{e}} \prod_{\mathbf{c} \sim \mathbf{x}} \theta_{\mathbf{c}},$$

Notice that $\phi(\mathbf{e})$ is linear in each parameter $\theta_{\mathbf{c}}$, and no two parameters in the same clique table are multiplied together. Therefore, if we apply a change of $\Delta\theta_{\mathbf{c}}$ to each parameter $\theta_{\mathbf{c}}$ in the clique table of $\mathbf{C}$, the change in $\phi(\mathbf{e})$ is:

$$\Delta\phi(\mathbf{e}) = \sum_{\mathbf{c}} \frac{\partial \phi(\mathbf{e})}{\partial \theta_{\mathbf{c}}} \Delta\theta_{\mathbf{c}}. \tag{7.1}$$

We also note that the conditional probability $Pr(y \mid \mathbf{e})$ can be expressed as:

$$Pr(y \mid \mathbf{e}) = \frac{Pr(y, \mathbf{e})}{Pr(\mathbf{e})} = \frac{\zeta\phi(y, \mathbf{e})}{\zeta\phi(\mathbf{e})} = \frac{\phi(y, \mathbf{e})}{\phi(\mathbf{e})}.$$

Since $\phi(\mathbf{e})$ and $\phi(y, \mathbf{e})$ are both multi-linear functions of the network parameters, a conditional probability can be expressed as a ratio of multi-linear functions of the network parameters. Because this property holds true for both Bayesian and Markov networks, this means the procedure of tuning Bayesian network parameters given in Section 2.1 can also be applied for tuning Markov network parameters.

Therefore, to ensure the query constraint $Pr(y \mid \mathbf{e}) \geq \kappa$, it suffices to ensure that $\phi(y, \mathbf{e}) \geq \kappa \cdot \phi(\mathbf{e})$, or equivalently:

$$\varphi(y, \mathbf{e}) + \Delta\phi(y, \mathbf{e}) \geq \kappa \left(\varphi(\mathbf{e}) + \Delta\phi(\mathbf{e})\right),$$

where $\varphi(\mathbf{e})$ and $\varphi(y, \mathbf{e})$ are the current ϕ values of $\mathbf{e}$ and $y, \mathbf{e}$ respectively. If we apply a change of $\Delta\theta_{\mathbf{c}}$ to $\theta_{\mathbf{c}}$, from Equation 7.1, we have:

$$\varphi(y, \mathbf{e}) + \sum_{\mathbf{c}} \frac{\partial \phi(y, \mathbf{e})}{\partial \theta_{\mathbf{c}}} \Delta\theta_{\mathbf{c}} \geq \kappa \left(\varphi(\mathbf{e}) + \sum_{\mathbf{c}} \frac{\partial \phi(\mathbf{e})}{\partial \theta_{\mathbf{c}}} \Delta\theta_{\mathbf{c}}\right).$$

Rearranging the terms, we get the following corollaries.

Corollary 7.1 *To ensure the probability distribution Pr induced by a Markov network satisfies the query constraint $Pr(y \mid \mathbf{e}) \geq \kappa$, we must change each parameter $\theta_{\mathbf{c}}$ in the clique table of $\mathbf{C}$ by $\Delta\theta_{\mathbf{c}}$ such that:*

$$\varphi(y, \mathbf{e}) - \kappa \cdot \varphi(\mathbf{e}) \geq \sum_{\mathbf{c}} \Delta\theta_{\mathbf{c}} \left(-\frac{\partial \phi(y, \mathbf{e})}{\partial \theta_{\mathbf{c}}} + \kappa \cdot \frac{\partial \phi(\mathbf{e})}{\partial \theta_{\mathbf{c}}}\right),$$

where $\varphi(y, \mathbf{e})$ and $\varphi(\mathbf{e})$ are the current ϕ values of $y, \mathbf{e}$ and $\mathbf{e}$ respectively.

Corollary 7.2 *If instead of changing all parameters in the clique table of $\mathbf{C}$, we are only allowed to change a single parameter $\theta_{\mathbf{c}}$ by $\Delta\theta_{\mathbf{c}}$, the solution of Corollary 7.1 becomes:*

$$\varphi(y, \mathbf{e}) - \kappa \cdot \varphi(\mathbf{e}) \geq \Delta\theta_{\mathbf{c}} \left(-\frac{\partial \phi(y, \mathbf{e})}{\partial \theta_{\mathbf{c}}} + \kappa \cdot \frac{\partial \phi(\mathbf{e})}{\partial \theta_{\mathbf{c}}}\right),$$

which returns a solution interval of $\Delta\theta_{\mathbf{c}}$.

Therefore, to solve for $\Delta\theta_{\mathbf{c}}$ in Corollary 7.2 for all network parameters, we need to compute the initial values $\varphi(y, \mathbf{e})$ and $\varphi(\mathbf{e})$, which should already be known when computing the initial probability of $y \mid \mathbf{e}$, and the partial derivatives $\partial\phi(y, \mathbf{e})/\partial\theta_{\mathbf{c}}$ and $\partial\phi(\mathbf{e})/\partial\theta_{\mathbf{c}}$ for all parameters $\theta_{\mathbf{c}}$. To do this, we can use a procedure whose complexity is $O(n2^w)$, where n is the number of variables in the Markov network, and w is the tree-width, which depends on the connectivity of the network structure, similar to the one proposed to compute partial derivatives for Bayesian networks [Dar03]. This can greatly help users debug a Markov network when they are faced with query results that do not match their expectations.

We now consider an example to illustrate this procedure.

Example 7.1 *We consider again the Markov network Disease given in Example 1.3. The distribution induced by the current Markov network gives us the conditional query value $Pr(\bar{a_2} \mid a_1) = .789$. Assume that we would like to change a single parameter in the clique table of $\{A_2, B_1\}$ to ensure the constraint $Pr(\bar{a_2} \mid a_1) \geq .9$. We need to use Corollary 7.2 to solve for the minimum $\Delta\theta_{\mathbf{c}}$ for each θ_c. The solutions are:*

$$\begin{aligned}
\Delta\theta_{a_2,b_1} &\leq -2.93; \\
\Delta\theta_{a_2,\bar{b_1}} &\leq -1.47; \\
\Delta\theta_{\bar{a_2},b_1} &\geq 12; \\
\Delta\theta_{\bar{a_2},\bar{b_1}} &\geq 8.8.
\end{aligned}$$

However, notice that because the parameter values have to be non-negative, the solution of $\Delta\theta_{a_2,b_1}$ is impossible to achieve. Therefore, no possible change in the parameter θ_{a_2,b_1} is possible to ensure the query constraint. On the other hand, we can decrease the parameter $\theta_{a_2,\bar{b_1}}$ from 2 to .533 to ensure the query constraint.

If we are able to change all parameters in the clique table of $\{A_2, B_1\}$ to satisfy the query constraint, we need to find a solution in Corollary 7.1. As a consequence, we are now faced with a solution space of multiple parameter changes, and we want to commit to a solution that disturbs the network the least. We will discuss this in the next section using the distance measure defined in Definition 4.1.

7.2 Bounding Belief Changes Between Markov Networks

We now answer the following question in the context of Markov networks: what is the bound on the difference between some query value induced by two networks that have the same structure but different parameter values? We will answer it by computing the distance measure between the distributions induced by the two Markov networks, which provides a bound on the amount of change in any query between two probability distributions.

As we have shown in Corollary 5.2, the distance measure can be computed using local information for Bayesian networks when changing a single CPT. We can get a similar result for Markov networks, where the distance measure between distributions induced by two Markov networks that differ on only a single clique table can be computed by the distance measure between the tables.

Theorem 7.1 *Given distributions Pr and Pr' induced by two Markov networks $\mathcal{M}$ and $\mathcal{M}'$ that differ on only the parameters in a single clique table, such that the clique tables are $\Theta_{\mathbf{C}}$ and $\Theta'_{\mathbf{C}}$ respectively, the distance measure between Pr*

and Pr' is given by:[1]

$$\begin{aligned} D(Pr, Pr') &= D(\Theta_{\mathbf{C}}, \Theta'_{\mathbf{C}}) \\ &= \ln \max_{\mathbf{c}} \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}} - \ln \min_{\mathbf{c}} \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}}, \end{aligned}$$

if $\partial\psi(\mathbf{x})/\partial\theta_{\mathbf{c}} \neq 0$ for all $\mathbf{c} \sim \mathbf{x}$.[2]

Therefore, the global distance measure between the distributions induced is equal to the local distance measure between the individual clique tables. This is useful for computing the bound on the change in a query after changing parameters in a clique table. For example, if we apply an arbitrary change on a single parameter $\theta_{\mathbf{c}}$, the distance measure is given by:

$$D(Pr, Pr') = \left| \ln \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}} \right|,$$

and the change in the value of query $y \mid \mathbf{e}$ is bounded by:

$$\frac{\theta_{\mathbf{c}}}{\theta'_{\mathbf{c}}} \leq \frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \leq \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}}. \tag{7.2}$$

This means for Markov networks, *the relative change in query odds is bounded by the relative change in the parameter itself, not the relative change in the parameter odds* as for Bayesian networks. This is an important distinction between Markov networks and Bayesian networks.

As an example, suppose we want to ensure the robustness of the query $Pr(y \mid \mathbf{e})$ after we apply a parameter change. Assume that we define robustness as the relative change in any query odds to be no more than 1.5, or a log-odds change

[1] *We have again loosen up the definition of the distance measure given in Definition 4.1 to accept clique tables instead of only probability distributions.*

[2] *This condition is satisfied when the network parameters are all strictly positive. However, this is a sufficient condition, not a necessary condition. The necessary condition is there exists some $\mathbf{x}$ such that $\partial\psi(\mathbf{x})/\partial\theta_{\mathbf{c}} \neq 0$ for all $\mathbf{c}$. This means that changing any parameter $\theta_{\mathbf{c}}$ will have some impact on the joint potential ψ.*

of no more than .405. For example, if currently we have $Pr(y \mid \mathbf{e}) = .75$, the new query value must stay in the interval $[.667, .818]$ after the parameter change. We may ask, how much change can we apply to a network parameter if we want to ensure robustness? *The answers for Bayesian networks and Markov networks are different due to our previous results*, as we will show next.

For a Bayesian network, the amount of permissible parameter change is determined by Theorem 3.3, and is plotted against the initial parameter value in Figure 7.1. We can see that the amount of permissible parameter change is small when the parameters have extreme values close to 0 or 1, since the relative odds change is large when even a very small absolute change is applied.

On the other hand, for a Markov network, the amount of permissible parameter change is determined by Inequality 7.2, and is plotted against the initial parameter value in Figure 7.2. We can see that the amount of permissible parameter change is proportional to the parameter values, since relative change is the factor here instead of relative odds change.

Therefore, for a Bayesian network, the sensitivity of the network with respect to a parameter is largest for extreme parameter values close to 0 or 1, and becomes smaller as the its value approaches .5, while for a Markov network, the sensitivity of the network with respect to a parameter is proportional to its value, and increases as it grows larger.

The distance measure is also useful in many other aspects of sensitivity analysis of Markov networks. For example, given the possible single parameter changes given in Example 7.1, we can choose the one that disturbs the network the least according to the distance measure. In this case, the most preferred single parameter change is to decrease the parameter $\theta_{a_2,\bar{b_1}}$ from 2 to .533, incurring a distance measure of 1.322.

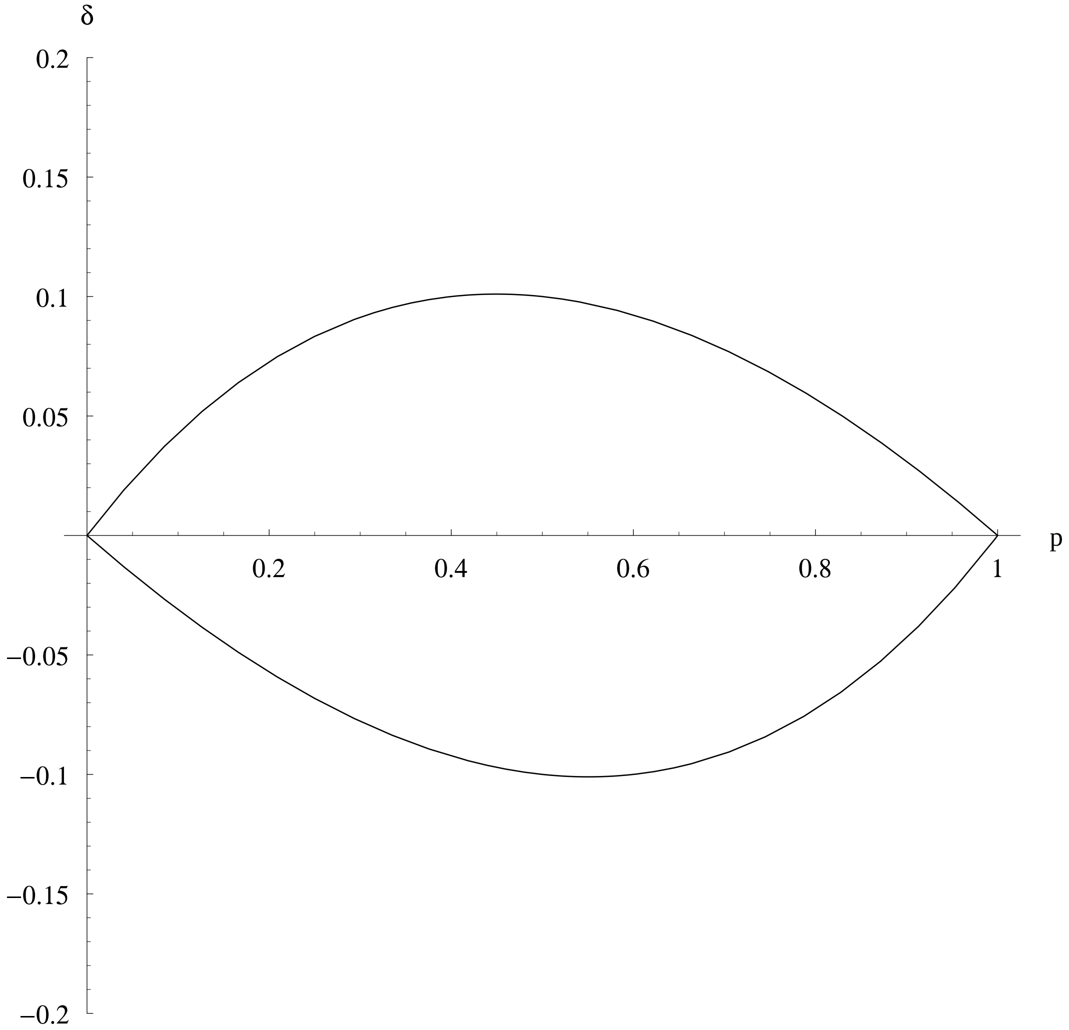


Figure 7.1: The plot of δ against p that would guarantee the query $Pr(y \mid \mathbf{e}) = .75$ to stay within the interval $[.667, .818]$, where p and $p' = p + \delta$ are the initial and new parameter values of a Bayesian network respectively.

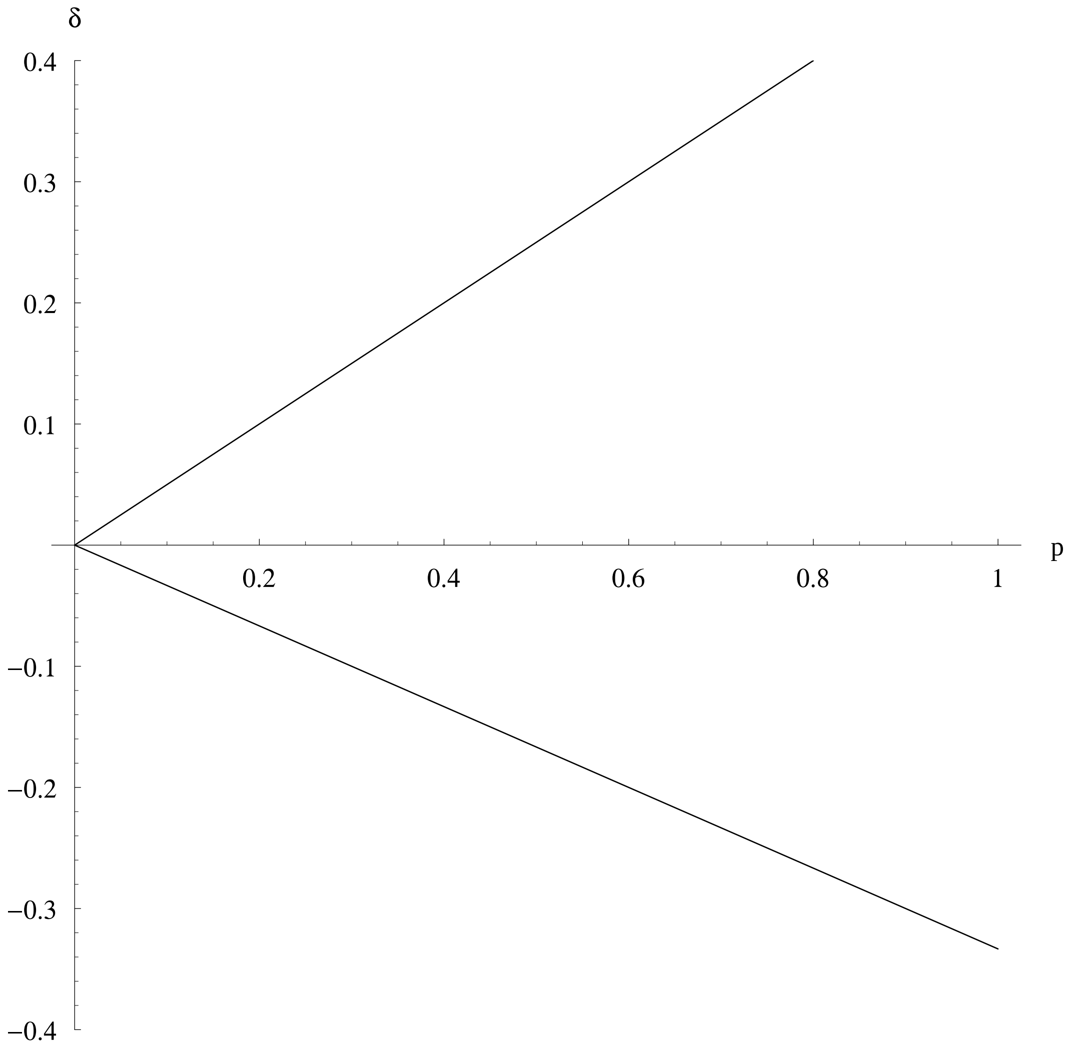


Figure 7.2: The plot of δ against p that would guarantee the query $Pr(y \mid \mathbf{e}) = .75$ to stay within the interval $[.667, .818]$, where p and $p' = p + \delta$ are the initial and new parameter values of a Markov network respectively.

Moreover, we can also use the distance measure to find an optimal solution of changing all parameters in a clique table, which is the solution given in Corollary 7.1 that minimizes the distance measure. Given an optimal solution, the relative changes in all parameters must be the same, since to obtain another solution that satisfies the constraint, we must increase the relative change in one parameter while decreasing the relative change in another, thereby incurring a larger distance measure. For example, to ensure the query constraint $Pr(\bar{a_2} \mid a_1) \geq .9$, we would like to decrease the parameters θ_{a_2,b_1} and $\theta_{a_2,\bar{b_1}}$ and increase the parameters $\theta_{\bar{a_2},b_1}$ and $\theta_{\bar{a_2},\bar{b_1}}$, such that the relative changes in all parameters are the same. However, because only the ratios between the parameters are important, we can keep the first two parameters constant and only increase the last two parameters. The optimal solution computed is:

$$\begin{aligned} \Delta\theta_{a_2,b_1} &= 0; \\ \Delta\theta_{a_2,\bar{b_1}} &= 0; \\ \Delta\theta_{\bar{a_2},b_1} &= 6.26; \\ \Delta\theta_{\bar{a_2},\bar{b_1}} &= 8.68. \end{aligned}$$

This optimal solution incurs a distance measure of 1.127. It involves all parameters in the clique table and incurs a smaller distance measure than any of the single parameter changes computed in Example 7.1.

Finally, if we change the parameters in different clique tables that do not share any variables, the distance measure can be computed as the sum of the local distance measures between the clique tables, similar to Bayesian networks.

CHAPTER 8

Belief Revision

In Section 1.3, we mentioned two types of local belief changes: soft evidence and Bayesian network parameter changes. In previous chapters, we focused on Bayesian network parameter changes, where we can obtain a unique new probability distribution. However, for the case of soft evidence, it is not obvious what the new probability distribution should be. In this chapter, we will look at belief revision, which addresses this problem. We will compare two existing methods of belief revision, and use the distance measure defined in Chapter 4 to bound global belief changes due to soft evidence [CD03a, CD05b].

In contrast to hard evidence, which confirms for certain that a random variable has taken on some value, we are often given *soft evidence* (or uncertain evidence) that makes us believe that the probability distribution of a random variable has changed from our initial beliefs. Due to different interpretations of the evidential reports, there are two main methods of revising probabilistic beliefs given soft evidence [Pea01]. The first method is known as *Jeffrey's rule* and is based on the principle of *probability kinematics*, which can be viewed as a principle for minimizing belief changes [Jef65]. The second method is called the *virtual evidence method* and was proposed by Pearl in the context of Bayesian networks, even though it can be easily generalized to arbitrary probability distributions, and is based on recasting soft evidence as hard evidence on some virtual event [Pea88]. We will analyze both methods with respect to the following questions:

1. How should one specify soft evidence?
2. How should one revise a probability distribution?
3. How should one interpret informal evidential statements?
4. Should, and do, iterated belief revisions commute?
5. What guarantees can be offered on the amount of belief change induced by a particular revision?

To answer the first question, we note that soft evidence must be specified as a formal constraint on posterior beliefs. This can be an absolute constraint on posterior beliefs, or a relative constraint on how posterior beliefs should relate to prior beliefs. Yet, the constraint specified by a given evidence usually does not define posterior beliefs uniquely, and hence, we need to adopt a principle that commits to a unique set of posterior beliefs that satisfy the given constraint. This principle then defines the belief revision method. However, even if we choose a method of specifying evidence formally, and a method of revising beliefs, there is still the problem of interpreting informal evidential statements, which are usually specified in natural language. These statements may not map directly to our formal specification language for evidence, and it is this process of interpretation that appears to underlie most of the controversies on revision methods.

Our main findings can be summarized as follows. First of all, Jeffrey's rule and Pearl's method both revise beliefs using the principle of probability kinematics. Whereas Jeffrey's rule explicitly commits to this principle, Pearl's method is based on a different revision principle, yet it implicitly implies the principle of probability kinematics, leading to the same revision method as that of Jeffrey's. The difference between Jeffrey's rule and Pearl's method is in the way soft evidence is specified. Jeffrey requires soft evidence to be specified in terms of the *effect* it has on beliefs once accepted, which is a function of both evidence strength and beliefs held before the evidence is observed. Pearl, on the other hand, requires soft evidence to be specified in terms of its *strength* only. Despite this difference, we will show that we can easily translate between the two methods of specifying evidence and provide the equations for carrying out this translation.

The multiplicity of methods for specifying evidence also raises an important question: how should informal statements about evidence be captured formally using available methods? For example, what should the statement, "seeing these clouds, I believe there is an 80% chance that it will rain," translate to? We will discuss the implicit difference between the two revision methods in interpreting informal evidential statements, where we emphasize its subtlety and show how it appears to be the reason that the two methods reach different conclusions.

As for the question of iterated belief revisions, it is well known that Jeffrey's rule does not commute, and hence, the order in which different pieces of soft evidence are accepted matters [DZ82]. This has long been perceived as a problem, until clarified recently by the work of Wagner who observed that the method of specifying evidence used by Jeffrey's rule is dependent of what is believed before the evidence is observed, and hence, should not be commutative to start with [Wag02]. Wagner proposed a method of specifying evidence, based on the notion of *Bayes factor*, and argued that this method specifies only the strength of evidence, and is independent of the beliefs held before attaining the evidence. Wagner argued that when evidence is specified in this particular way, iterated revisions should commute. He even showed that combining this method of specifying evidence with the principle of probability kinematics leads to a revision rule that commutes. We will actually show that Pearl's method of virtual evidence specifies evidence according to Bayes factors, exactly as proposed by Wagner, and hence, corresponds exactly to the proposal he calls for. Therefore, the results we will discuss in this chapter unify the two main methods of probabilistic belief revision proposed by Jeffrey and Pearl, and show that differences between them amount only to a difference in the protocol for specifying soft evidence.

Our last set of results relate to the problem of providing guarantees on the amount of belief change induced by a revision. In Chapter 4, we proposed a distance measure for bounding belief changes, by providing guarantees on the amount of query change. We will demonstrate how this distance measure can be computed when one distribution is obtained from another using the principle of probability kinematics. The guarantees provided by this distance measure can be realized when applying either Jeffrey's rule or Pearl's method, since they both perform revisions based on the principle of probability kinematics.

8.1 Probability Kinematics and Jeffrey's Rule

Consider the problem of revising a probability distribution Pr given soft evidence relating to a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$. One of the methods of specifying soft evidence is through the effect that it would have on beliefs once accepted. Specifically, according to this method, we have to specify evidence by providing the following set of *posterior probabilities*:

$$Pr'(\gamma_i) = \rho_i \text{ for } i = 1, \ldots, n,$$

where Pr' denotes the new probability distribution that results from accepting the given evidence. Therefore, to revise the distribution Pr, we must choose a unique posterior distribution Pr' that satisfies the above constraint. The principle of probability kinematics [Jef65], which we now define, assumes that the conditional belief in any event α given any γ_i remains unchanged.

Definition 8.1 *Given two probability distributions Pr and Pr' that disagree on the probabilities they assign to a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, the distribution Pr' is said to be obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$, iff for every event α in the probability space:*

$$Pr(\alpha \mid \gamma_i) = Pr'(\alpha \mid \gamma_i) \textit{ for } i = 1, \ldots, n. \tag{8.1}$$

This concept was proposed by Jeffrey [Jef65] to capture the notion that even though Pr and Pr' disagree on the probabilities of events $\gamma_1, \ldots, \gamma_n$, they agree on their relevance to every event α.

We now define the belief revision method of Jeffrey's rule [Jef65], which can be viewed as consisting of two components: a suggestion to specify soft evidence as a constraint on the posterior probabilities of events $\gamma_1, \ldots, \gamma_n$; and a proposal to choose the posterior distribution using the principle of probability kinematics.

Definition 8.2 *Given an initial distribution Pr and some soft evidence bearing on a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, and assuming that such evidence is specified by the set of posterior probabilities:*

$$Pr'(\gamma_i) = \rho_i \textit{ for } i = 1, \ldots, n, \tag{8.2}$$

the new posterior distribution Pr' proposed by Jeffrey's rule is as follows:

$$Pr'(\alpha) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \rho_i \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)}. \tag{8.3}$$

Theorem 8.1 *The posterior distribution Pr' given in Equation 8.3 is the one and only distribution that satisfies the constraint given by Equation 8.2 and is obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$.*

We emphasizxe that we are drawing a distinction between the principle of probability kinematics and the revision method of Jeffrey's rule, which are often considered synonymous. As we mentioned, Jeffrey's rule arises from a combination of two proposals: the principle of probability kinematics, and the specification of soft evidence using a posterior distribution. It is possible to combine the principle of probability kinematics with other methods of specifying evidence, as we will discuss later.

We now show an example of using Jeffrey's rule.

Example 8.1 *[Jef65] Suppose we are given a piece of cloth, where its color can be one of: green (c_g), blue (c_b), or violet (c_v). We want to know whether, on the next day, the cloth will be sold (s), or not sold ($\bar{s}$). Our initial state of belief is given by the distribution Pr:*

$$\begin{aligned}&Pr(s, c_g) = .12, \quad Pr(s, c_b) = .12, \quad Pr(s, c_v) = .32,\\ &Pr(\bar{s}, c_g) = .18, \quad Pr(\bar{s}, c_b) = .18, \quad Pr(\bar{s}, c_v) = .08.\end{aligned}$$

Our initial belief on the color of the cloth is $(Pr(c_g), Pr(c_b), Pr(c_v)) = (.3, .3, .4)$. Suppose we now inspect the cloth by candlelight, and conclude that our new belief on the color of the cloth should be $(Pr'(c_g), Pr'(c_b), Pr'(c_v)) = (.7, .25, .05)$. If we revise our beliefs by applying Jeffrey's rule (Equation 8.3), we get the new distribution Pr':

$$\begin{aligned}&Pr'(s, c_g) = .28, \quad Pr'(s, c_b) = .10, \quad Pr'(s, c_v) = .04,\\ &Pr'(\bar{s}, c_g) = .42, \quad Pr'(\bar{s}, c_b) = .15, \quad Pr'(\bar{s}, c_v) = .01.\end{aligned}$$

8.2 Virtual Evidence and Pearl's Method

The problem of revising a probability distribution using soft evidence can be approached from a different perspective than that of the principle of probability kinematics. For example, when we have soft evidence about some mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, we can recast this evidence as *hard evidence* on some *virtual event* η, where the relevance of $\gamma_1, \ldots, \gamma_n$ to η is uncertain. According to this approach, the uncertainty regarding evidence on $\gamma_1, \ldots, \gamma_n$ is now interpreted as uncertainty in the relevance of $\gamma_1, \ldots, \gamma_n$ to the virtual event η, and this uncertainty is specified by the likelihood of γ_i given this virtual evidence η, $Pr(\eta \mid \gamma_i)$ for $i = 1, \ldots, n$. This belief revision method, called the virtual evidence method, is defined explicitly as follows.

Definition 8.3 *Given an initial distribution Pr and some soft evidence η bearing on a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, and assuming that such evidence is specified by $\lambda_1, \ldots, \lambda_n$ such that:*

$$\lambda_1 : \ldots : \lambda_n = Pr(\eta \mid \gamma_1) : \ldots : Pr(\eta \mid \gamma_n), \tag{8.4}$$

the revised distribution proposed by the virtual evidence method is $Pr(\cdot \mid \eta)$. Moreover, this method assumes that for every event α in the probability space, we have:

$$Pr(\eta \mid \gamma_i, \alpha) = Pr(\eta \mid \gamma_i) \textit{ for } i = 1, \ldots, n. \tag{8.5}$$

That is, the virtual event η depends only on the events $\gamma_1, \ldots, \gamma_n$ and is independent of every event α given γ_i for $i = 1, \ldots, n$.

Note that the likelihoods $Pr(\eta \mid \gamma_1), \ldots, Pr(\eta \mid \gamma_n)$ are not essential for the virtual evidence method, but the *likelihood ratios* $\lambda_1 : \ldots : \lambda_n = Pr(\eta \mid \gamma_1) : \ldots : Pr(\eta \mid \gamma_n)$ are. The following theorem proves that we can uniquely define the posterior distribution $Pr(\cdot \mid \eta)$ using the assumption given by Equation 8.5.

Theorem 8.2 *Given the constraint given by Equation 8.4, and the assumption given by Equation 8.5, we have:*

$$Pr(\alpha \mid \eta) = \frac{\sum_{i=1}^{n} \lambda_i Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)}. \tag{8.6}$$

Therefore, under the assumption given by Equation 8.5, the virtual evidence method is able to reduce the incorporation of soft evidence into that of incorporating certain evidence using Bayes conditioning.

The virtual evidence method is a generalization of Pearl's method of virtual evidence, which Pearl proposed in the context of Bayesian networks [Pea88]. The closed form of this method as given by Equation 8.6 for arbitrary probability distributions is original as far as we know.

We now show an example of using the virtual evidence method.

Example 8.2 *[Pea88] Suppose we are concerned with whether on any given day, the alarm of Mr. Holmes' house is triggered (values a and $\bar{a}$), and whether there is burglary at his house (values b and $\bar{b}$). Our initial state of belief is given by the distribution Pr:*

$$Pr(a, b) = .000095, \quad Pr(a, \bar{b}) = .009999,$$
$$Pr(\bar{a}, b) = .000005, \quad Pr(\bar{a}, \bar{b}) = .989901.$$

This means that on any given day, there is a burglary at Mr. Holmes' house with probability $Pr(b) = 1 \times 10^{-4}$. One day, Mr. Holmes receives a call from his neighbor, Mrs. Gibbons, saying she may have heard the alarm of his house being triggered. Since Mrs. Gibbons suffers from a hearing problem, Mr. Holmes concludes that there is an 80% chance that Mrs. Gibbons did hear the alarm triggered. This can be interpreted as follows: the probability that Mrs. Gibbons will make the call given that the alarm has been triggered is four times the probability that Mrs. Gibbons will make the call given that the alarm has not been *triggered. This soft evidence can be recast as hard evidence on the virtual event* η *(the event of Mr. Holmes receiving a call from Mrs. Gibbons), with likelihood ratios* $\lambda_a : \lambda_{\bar{a}} = Pr(\eta \mid a) : Pr(\eta \mid \bar{a}) = 4 : 1$*. We can apply Equation 8.6 and obtain the new distribution* $Pr(\cdot \mid \eta)$*:*

$$\begin{aligned} Pr(a, b \mid \eta) &\approx .000369, \quad Pr(a, \bar{b} \mid \eta) \approx .038820, \\ Pr(\bar{a}, b \mid \eta) &\approx .000005, \quad Pr(\bar{a}, \bar{b} \mid \eta) \approx .960806. \end{aligned}$$

Therefore, the new probability that there is a burglary at Mr. Holmes' house after revising our beliefs given this piece of virtual evidence is $Pr(b \mid \eta) \approx 3.74 \times 10^{-4}$.

8.3 Comparing the Revision Methods

From the illustrations of the two belief revision methods, Jeffrey's rule and Pearl's method of virtual evidence, we can see that a belief revision method can be broken into two parts: a formal constraint that is used to specify the soft evidence, and a principle of belief revision that commits to a unique distribution among many that satisfy the evidential constraint. For the formal constraint, Jeffrey's rule specifies evidence using posterior probabilities, while Pearl's method specifies evidence using likelihood ratios. For the principle of belief revision, Jeffrey's rule obeys the principle of probability kinematics explicitly, while in Pearl's method, beliefs are revised by conditioning on a virtual event η. In this section, we will compare the two revision methods with respect to these two parts, and consequently show how we can translate between the two methods.

8.3.1 Pearl's Method and Probability Kinematics

Our first result shows that the principle of belief revision underlying Pearl's method does indeed satisfy the principle of probability kinematics. Therefore, it actually uses the same belief revision principle as Jeffrey's rule, and what they differ on is how soft evidence is specified.

Theorem 8.3 *Given an initial distribution Pr and some soft evidence η bearing on a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, the new posterior distribution $Pr(\cdot \mid \eta)$ proposed by the virtual evidence method is the one obtained from the initial distribution Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$.*

Theorem 8.3 clarifies a misconception that Jeffrey's rule and Pearl's method are two different belief revision methods. In fact, what they differ on is not how beliefs are revised as they both obey the principle of probability kinematics, but in the constraint that is used to specify evidence. We now show how we can easily translate between the two different evidence specification methods.

8.3.2 Translating from Pearl's Method to Jeffrey's Rule

We first show how we can translate from the evidential constraint used by Pearl's method into one used by Jeffrey's rule.

Theorem 8.4 *Suppose we have an initial distribution Pr and some soft evidence η bearing on a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, and this piece of soft evidence is specified by likelihood ratios $\lambda_1 : \ldots : \lambda_n = Pr(\eta \mid \gamma_1) : \ldots : Pr(\eta \mid \gamma_n)$. The new posterior distribution $Pr(\cdot \mid \eta)$ proposed by the virtual evidence method can be obtained using Jeffrey's rule given that the soft evidence is specified by the following set of posterior probabilities:*

$$Pr'(\gamma_i) = \rho_i = Pr(\gamma_i \mid \eta) \text{ for } i = 1, \ldots, n. \tag{8.7}$$

We now illustrate the translation from Pearl's method to Jeffrey's rule in Theorem 8.4 by revisiting Example 8.2.

Example 8.3 *In Example 8.2, the new distribution $Pr(\cdot \mid \eta)$ is obtained from the initial distribution Pr by applying revision using Pearl's method. From Theorem 8.4, the equivalent distribution $Pr' = Pr(\cdot \mid \eta)$ can be obtained by applying Jeffrey's rule, given soft evidence specified by the following set of posterior probabilities, using Equation 8.6:*

$$\begin{aligned}
Pr'(a) &= Pr(a \mid \eta) \\
&= \frac{\lambda_a Pr(a)}{\lambda_a Pr(a) + \lambda_{\bar{a}} Pr(\bar{a})} \\
&= \frac{4 \times .010094}{4 \times .010094 + 1 \times .989906} \\
&\approx .039189; \\
Pr'(\bar{a}) &= Pr(\bar{a} \mid \eta) \\
&= \frac{\lambda_{\bar{a}} Pr(\bar{a})}{\lambda_a Pr(a) + \lambda_{\bar{a}} Pr(\bar{a})} \\
&= \frac{1 \times .989906}{4 \times .010094 + 1 \times .989906} \\
&\approx .960811.
\end{aligned}$$

8.3.3 Translating from Jeffrey's Rule to Pearl's Method

We now show how we can translate from the evidential constraint used by Jeffrey's rule into one used by Pearl's method.

Theorem 8.5 *Suppose we have an initial distribution Pr and some soft evidence bearing on a set of mutually exclusive and exhaustive events $\gamma_1, \dots, \gamma_n$, and this piece of soft evidence is specified by a set of posterior probabilities $Pr'(\gamma_i) = \rho_i$ for $i = 1, \dots, n$. The new posterior distribution Pr' proposed by Jeffrey's rule can be obtained using the virtual evidence method given that the soft evidence is specified by the following likelihood ratios:*

$$\lambda_1 : \dots : \lambda_n = \frac{\rho_1}{Pr(\gamma_1)} : \dots : \frac{\rho_n}{Pr(\gamma_n)}. \tag{8.8}$$

We now illustrate the translation from Jeffrey's rule to Pearl's method in Theorem 8.5 by revisiting Example 8.1.

Example 8.4 *In Example 8.1, the new distribution Pr' is obtained from the initial distribution Pr by applying revision using Jeffrey's rule. From Theorem 8.5, the equivalent distribution $Pr(\cdot \mid \eta) = Pr'$ can be obtained by applying Pearl's method, given virtual evidence η specified by the following likelihood ratios:*

$$\begin{aligned}\lambda_{c_g} : \lambda_{c_b} : \lambda_{c_v} &= \frac{Pr'(c_g)}{Pr(c_g)} : \frac{Pr'(c_b)}{Pr(c_b)} : \frac{Pr'(c_v)}{Pr(c_v)} \\ &= \frac{.7}{.3} : \frac{.25}{.3} : \frac{.05}{.4} \\ &= 7 : 2.5 : .375.\end{aligned}$$

8.4 Belief Revision in Bayesian Networks

In this section, we will describe the procedure of applying belief revision in the context of Bayesian networks, using either Pearl's method or Jeffrey's rule. The method of revision by virtual evidence was first introduced by Pearl in the context of Bayesian networks [Pea88]. Suppose we have some virtual evidence η bearing on variable Y in a Bayesian network, which has values $y_1, \ldots, y_n$. This virtual evidence is represented in the Bayesian network by adding an auxiliary variable Z, and a directed edge $Y \rightarrow Z$, where one value of Z, say z, corresponds to the virtual event η. This ensures the key assumption given by Equation 8.5, that the virtual event z is independent of every event α given y_i, i.e., $Pr(z \mid y_i, \alpha) = Pr(z \mid y_i)$ for $i = 1, \ldots, n$, which follows from the independence semantics of Bayesian networks [Pea88]. The uncertainty of evidence is quantified by the likelihood ratios $\lambda_1, \ldots, \lambda_n$, and the parameter values in the CPT of Z are assigned such that $Pr(z \mid y_1) : \ldots : Pr(z \mid y_n) = \lambda_1 : \ldots : \lambda_n$.[1] Finally, we accommodate the

[1] Note that there are multiple CPTs that satisfy this condition.

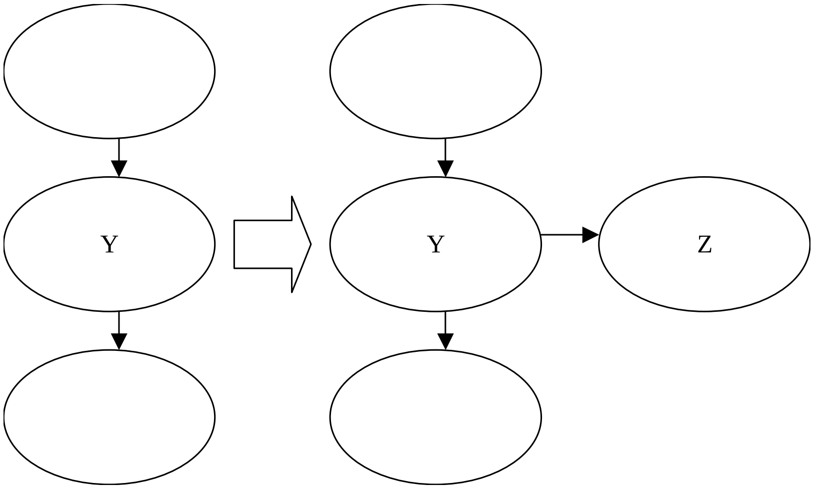


Figure 8.1: An illustration of the process of applying the virtual evidence method in a Bayesian network, by adding an auxiliary variable Z as a child of variable Y, where the virtual evidence η bears on.

presence of the virtual event z by asserting the observation $Z = z$ in the Bayesian network. This process is shown in Figure 8.1. We now show a simple example.

Example 8.5 *The probability distribution Pr in Example 8.2 can be induced by the following Bayesian network with two variables: A, which represents whether the alarm of Mr. Holmes' house is triggered (values a and $\bar{a}$); and B, which represents whether there is a burglary at his house (values b and $\bar{b}$). To represent the influence between the two variables, there is a directed edge from B to A. We also assign values to the following network parameters: $Pr(a \mid b) = .95$, meaning the alarm is triggered if there is a burglary with probability $.95$; $Pr(a \mid \bar{b}) = .01$, meaning the alarm is triggered if there is no burglary with probability $.01$; and $Pr(b) = 1 \times 10^{-4}$, meaning on any given day, there is a burglary at Mr. Holmes' house with probability 1×10^{-4}.*

Suppose now Mr. Holmes receives a call from his neighbor, Mrs. Gibbons, *saying she may have heard the alarm of his house being triggered, and he concludes that there is an* 80% *chance that Mrs. Gibbons did hear the alarm triggered. This soft evidence can be recast as hard evidence on the virtual event* η*, with likelihood ratios* $\lambda_a : \lambda_{\bar{a}} = 4 : 1$*. To incorporate this virtual evidence into the Bayesian network, we add the auxiliary variable* Z*, and the directed edge* $A \rightarrow Z$*, where the value* z *of* Z *corresponds to the virtual event* η*, and the parameters in the CPT of* Z *are assigned such that* $Pr(z \mid a) : Pr(z \mid \bar{a}) = 4 : 1$*. For example, we can assign* $Pr(z \mid a) = .4$ *and* $Pr(z \mid \bar{a}) = .1$*. After asserting the observation* $Z = z$ *in the Bayesian network, we can easily compute values of probabilistic queries by performing inference. For example, the probability that there is a burglary at Mr. Holmes' house is now* $Pr(b \mid z) \approx 3.74 \times 10^{-4}$.

On the other hand, there was no known proposal for applying Jeffrey's rule in the context of Bayesian networks. However, due to our results on the translation between Jeffrey's rule and Pearl's method in Section 8.3, we immediately have a proposal, by first translating the evidential constraint used by Jeffrey's rule into one used by Pearl's method using Theorem 8.5, and then performing belief revision using the procedure shown above. We now show a simple example.

Example 8.6 *The probability distribution* Pr *in Example 8.1 can be induced by the following Bayesian network with two variables:* C*, which represents the color of the cloth (values* c_g*,* c_b *and* c_v*); and* S*, which represents whether the cloth is sold on the next day (values* s *and* $\bar{s}$*). To represent the influence between the two variables, there is a directed edge from* C *to* S*. We also assign values to the following network parameters:* $(Pr(c_g), Pr(c_b), Pr(c_v)) = (.3, .3, .4)$*; and* $Pr(s \mid c_g) = .4$*,* $Pr(s \mid c_b) = .4$*, and* $Pr(s \mid c_v) = .8$*.*

Suppose we now inspect the cloth by candlelight, and conclude that our new belief on the color of the cloth should be $(Pr'(c_g), Pr'(c_b), Pr'(c_v)) = (.7, .25, .05)$.

To incorporate this soft evidence into the Bayesian network, we first have to interpret the inspection of the cloth by candlelight as virtual evidence. In Example 8.4, we show how we can translate this evidential constraint into one used by Pearl's method. The soft evidence can now be recast as virtual event η*, with likelihood ratios* $\lambda_{c_g} : \lambda_{c_b} : \lambda_{c_v} = 7 : 2.5 : .375$*. We can now follow the procedure of incorporating virtual evidence: we first add an auxiliary variable* Z *as a child of variable* C*, then specify the CPT of* Z *such that the parameter values are consistent with the likelihood ratios, and finally assert the observation of the virtual event.*

8.5 Interpreting Evidential Statements

We now turn our attention to the investigation of the evidence specification protocols adopted by Jeffrey's rule and Pearl's method in relation to the problem of formally interpreting evidential statements.[2] Consider the following statement as an example:

> "Looking at this evidence, I am willing to bet 2 : 1 that David is not the killer."

This statement can be formally interpreted using either protocol. For example, if α denotes the event that David is not the killer, this statement can be interpreted in two ways:

1. After accepting the evidence, the probability that David is not the killer becomes twice the probability that David is the killer: $Pr'(\alpha) = 2/3$ and $Pr'(\bar{\alpha}) = 1/3$;

[2] This section is a summary of Pearl's discussions on this issue [Pea01], in the context of the approach we take in this chapter by dividing the belief revision process into an evidence specification method and a revision principle.

2. The probability that I will see this evidence η given that David is not the killer is twice the probability that I will see it given that David is the killer: $Pr(\eta \mid \alpha) : Pr(\eta \mid \bar{\alpha}) = 2 : 1$.

The first interpretation translates directly into a formal piece of evidence, Jeffrey's style, and can be characterized as an "all things considered" interpretation because it is a statement about the agent's final beliefs, which are functions of both his prior beliefs and the evidence [Pea01]. On the other hand, the second interpretation translates directly into a formal piece of evidence, Pearl's style, and can be characterized as a "nothing else considered" interpretation because it is a statement about the evidence only [Pea01].

The two interpretations can lead to contradictory conclusions about the evidence. If we use the "nothing else considered" approach to interpret our previous statement, we will conclude that the evidence is against David being the killer. However, if we use the "all things considered" interpretation, it is not clear whether the evidence is for or against David being the killer, unless we know its prior probability. For example, if David is one of four suspects who are equally likely to be the killer, i.e., its prior probability is $Pr(\alpha) = 3/4$, this evidence has actually increased the probability that David is the killer! Because of this, Pearl argued for the "nothing else considered" interpretation, as it provides a summary of the evidence alone, and discussed how people tend to use betting odds to quantify their beliefs even when they are based on the evidence only [Pea01].

Example 8.2 provides another opportunity to illustrate the subtlety involved in interpreting evidential statements. The evidential statement in this case is "Mr. Holmes concludes that there is an 80% chance that Mrs. Gibbons did hear the alarm triggered." Interpreting this statement using the "all things considered" approach gives us the conclusion that $Pr'(a) : Pr'(\bar{a}) = 4 : 1$, where a

denotes the event that the alarm has been triggered. This interpretation assumes that the 4 : 1 ratio applies to the posterior beliefs in a and $\bar{a}$, after Mr. Holmes has accommodated the evidence provided by Mrs. Gibson. However, in Example 8.2, this statement was given a "nothing else considered" interpretation, as by Pearl [Pea88, pp. 44–47], where the 4 : 1 ratio is taken as a quantification of the evidence strength, i.e., the statement is interpreted as $Pr(\eta \mid a) : Pr(\eta \mid \bar{a}) = 4 : 1$, where η is the evidence. In fact, the two interpretations lead to two different probability distributions, and hence, give us different results for probabilistic queries. For example, if we use the "all things considered" approach in interpreting this evidential statement, the probability of having a burglary is $Pr'(b) = 7.53 \times 10^{-3}$, which is much larger than the probability we get using the "nothing else considered" approach in Example 8.2, which is $Pr(b \mid \eta) = 3.74 \times 10^{-4}$.

From the discussions above, the formal interpretation of evidential statements appears to be a non-trivial task, which can be sensitive to context and communication protocols. Regardless of how this is accomplished though, we need to stress that the process of mapping an informal evidential statement into a revised probability distribution involves three distinct elements:

1. Adopting a formal method of specifying evidence;

2. Interpreting the informal evidential statement as a formal piece of evidence, according to the evidence specification method;

3. Applying a revision, by mapping the initial probability distribution and the formal piece of evidence into a new probability distribution, according to a belief revision principle.

As we have shown previously, Jeffrey's rule and Pearl's method both employ the same belief revision principle, i.e., the principle of probability kinematics.

Moreover, although they adopt different formal methods of specifying evidence, we can translate between the two methods, as shown in Section 8.3.

8.6 Virtual Evidence and Bayes Factors

In this section, we will aim to clarify the virtual evidence method by relating it to some classical concepts in probability theory.

First of all, when quantifying the strength of some evidence η on a hypotheses γ, we often compute the ratio of the odds of γ before and after accepting the evidence, $O(\gamma \mid \eta)/O(\gamma)$. This ratio is called the *odds factor in favor of* γ *by* η [Goo83], and its logarithm is called the *weight of evidence* η *in favor of* γ [Goo83]. The odds factor is given by [Goo50, Goo83]:

$$\frac{O(\gamma \mid \eta)}{O(\gamma)} = \frac{Pr(\eta \mid \gamma)}{Pr(\eta \mid \bar{\gamma})}.$$

Therefore, if we are given virtual evidence η on events γ and $\bar{\gamma}$, the likelihood ratio $Pr(\eta \mid \gamma)/Pr(\eta \mid \bar{\gamma})$ serves as a measure of the strength of the virtual evidence η on the hypotheses γ. The virtual evidence η argues for γ if the likelihood ratio is greater than 1, and argues against γ if the likelihood ratio is less than 1.

If we are given the general case where we have soft evidence on a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$, where $n > 2$, we need to expand the notion of odds factor, by defining the odds of γ_i against γ_j given η as the following probability ratio:

$$O(\gamma_i : \gamma_j \mid \eta) \overset{\text{def}}{=} \frac{Pr(\gamma_i \mid \eta)}{Pr(\gamma_j \mid \eta)}.$$

We can now in turn define the *odds factor in favor of* γ_i *against* γ_j *by* η as the ratio $O(\gamma_i : \gamma_j \mid \eta)/O(\gamma_i : \gamma_j)$. This ratio of new-to-old odds is actually equivalent to the Bayes factor defined in Definition 4.3:

$$B_{Pr(\cdot|\eta),Pr}(\gamma_i : \gamma_j) = \frac{O(\gamma_i : \gamma_j \mid \eta)}{O(\gamma_i : \gamma_j)}.$$

Therefore, the Bayes factor can be used as a measure of how much evidence η is in favor of γ_i against γ_j, and we can revise beliefs using the notion of Bayes factors. We first specify soft evidence on a set of mutually exclusive and exhaustive events $\gamma_1, \ldots, \gamma_n$ by providing the Bayes factor for every pair of events γ_i and γ_j, then commit to the principle of probability kinematics for belief revision. An interesting property of this method of specification is that Bayes factors do not constrain the initial probability distribution Pr, i.e., any soft evidence specified by Bayes factors is compatible with every distribution Pr.[3] Therefore, they are suitable for a "nothing else considered" interpretation of evidential statements.

In fact, we can show that this revision method using Bayes factors corresponds to the virtual evidence method. This has a number of implications. First, it provides an alternative and more classical semantics for the virtual evidence method. Second, it again confirms that the virtual evidence method obeys the principle of probability kinematics. Third, it shows that revisions by the virtual evidence method are commutative, as we will illustrate later. The following theorem shows how we can easily find the Bayes factors when we specify soft evidence using virtual evidence.

Theorem 8.6 *If $Pr(\cdot \mid \eta)$ is obtained from Pr by applying Pearl's method, given virtual evidence η specified by likelihood ratios $\lambda_1, \ldots, \lambda_n$, we have:*

$$B_{Pr(\cdot|\eta),Pr}(\gamma_i : \gamma_j) = \frac{\lambda_i}{\lambda_j} \text{ for } i, j = 1, \ldots, n.$$

[3] This is not true if we use ratios of probabilities instead of ratios of odds. For example, if $Pr'(\gamma) = 2Pr(\gamma)$, we must have $Pr(\gamma) \leq .5$ because $Pr'(\gamma) \leq 1$ [Wag02].

Therefore, we can obtain the same distribution as $Pr(\cdot \mid \eta)$ if we specify the soft evidence by the Bayes factor λ_i/λ_j for every pair of events γ_i and γ_j, and then revise our beliefs according to the principle of probability kinematics. The advantage of using the virtual evidence method for specifying soft evidence is that we only need to specify the n likelihood ratios $\lambda_1, \ldots, \lambda_n$ in order to define the n^2 Bayes factors that are necessary for belief revision.

8.6.1 Reasoning About Evidence

As we have said before, the virtual evidence method can be interpreted as a "nothing else considered" revision method, and does not depend on one's prior beliefs. In fact, this specification of evidence can be reasoned about and interpreted even when we do not have any prior beliefs. We illustrate this by the following example from Halpern and Pucella [HP03].

Suppose Alice has two coins, a fair one and a double-headed one. If she non-probabilistically chooses one of them and tosses it repeatedly, what is the probability of the coin landing heads in a single toss? Without knowing which coin she chooses, and how she chooses it, the only conclusion that can be drawn is that the probability is either 1/2 (if the fair coin is chosen) or 1 (if the double-headed coin is chosen).

Suppose now we know the results of the first 100 tosses, and all of them landed heads. What is the probability that the next toss lands heads? We can again conclude that it is still either 1/2 or 1 depending on which coin is used, as either coin cannot be ruled out from our observation. This is hardly useful because no matter how many of these consecutive tosses that landed heads we witness, the conclusion we can make on the probability that the next toss will land heads remains unchanged, when in fact the probability of the coin being double-headed should increase from before the tosses were made.

However, this piece of evidential information can be easily expressed and used if we interpret it as virtual evidence. If γ denotes the event of the coin being double-headed, $\bar{\gamma}$ denotes the event of the coin being fair, and η denotes the event of the coin landing heads in a single toss, we can quantify the evidence strength of the virtual event η on the hypotheses γ by the likelihood ratio $Pr(\eta \mid \gamma)/Pr(\eta \mid \bar{\gamma}) = 1/(1/2) = 2$. Because the likelihood ratio is greater than 1, this piece of evidence is in favor of the coin being double-headed, no matter what our prior beliefs are. Moreover, if we witness the coin all landing heads in 100 tosses, the likelihood ratio of this observation is 2^{100}, which means it is very strongly in favor of the coin being double-headed. Obviously, it is still not possible to determine the posterior probability of the coin being double-headed without knowing its prior probability. For example, we will still believe the coin is unlikely to be double-headed if its prior probability is 10^{-100}.

The advantage of specifying evidential information using virtual evidence is that the evidence can be shared among different agents with different prior beliefs (even those without prior beliefs), and it will be interpreted the same way by the different agents because the specification depends only on the evidence but not the prior beliefs. The likelihood ratios specified in the virtual evidence method capture completely whether the soft evidence is in favor or against a hypotheses, and also its strength. Recently, Halpern and Pucella proposed a logic for reasoning about evidence [HP03], which essentially views evidence as a confirmation function from the prior beliefs before making the observation, to the posterior beliefs after making the observation. The measure of evidence they use is the likelihood ratio, since it is the only function that does not assume that we have any prior beliefs on the hypotheses.

8.6.2 Commutativity of Iterated Revisions

We now proceed to discuss the problem of the commutativity of iterated revisions, i.e., whether the order in which we accept soft evidence matters.[4]

It is well known that iterated revisions by Jeffrey's rule are not commutative [DZ82]. As a simple example, suppose we are given a piece of soft evidence that suggests that the probability of event α is .7, followed by another that suggests that the probability of α is .8. After accepting both pieces of evidence in this particular order using Jeffrey's rule, the new probability of α becomes .8. However, if the reversed order is employed, the new probability of α becomes .7. In general, even when we are given pieces of soft evidence on different events, iterated revisions by Jeffrey's rule are not commutative, i.e., the final states of belief are different when different orders of incorporating evidence are used.

This was viewed as a problematic aspect of Jeffrey's rule for a long time, until clarified recently by Wagner [Wag02]. Wagner observed and stressed that the evidence specification method adopted by Jeffrey is suitable for the "all things considered" interpretation of evidential statements. Moreover, he argued convincingly that when evidential statements carry this interpretation, they must not be commutative to start with. So the lack of commutativity is not a problem of the revision method, but a property of the evidence specification method.

On the other hand, revisions by the virtual evidence method are commutative, and this is supported by Wagner, who suggested specifying evidence by Bayes factors leads to commutativity [Wag02]. Interestingly enough, he showed that when evidence is specified by Bayes factors and the revision method obeys the

[4] There is a key distinction between iterated revisions using hard evidence versus soft evidence. In the former case, pieces of evidence may be logically inconsistent, which adds another dimension of complexity to the problem [DP97], leading to different properties and treatments.

principle of probability kinematics, belief revision becomes commutative.[5] These two properties are satisfied by the virtual evidence method, as shown earlier.

8.7 Bounding Belief Changes Due To Belief Revision

An important question relating to belief revision is that of measuring the extent to which a revision disturbs existing beliefs. In Definition 4.1, we proposed a distance measure defined between two probability distributions that can be used to bound the amount of belief change induced by a revision. According to Theorem 4.2, if we are given a belief revision method, and are able to compute the distance measure between the initial and revised distributions, we can get a tight bound on the new probability of any conditional event after the belief revision, given its initial probability. Moreover, we showed in Section 4.2 that compared with other popular measures such as KL-divergence and Euclidean distance, this distance measure is the only one that can bound belief changes in the precise sense of providing a tight bound on the new probability of any conditional event.

We now compute the distance measure for belief revision methods based on the principle of probability kinematics.

Theorem 8.7 *If Pr' is obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$, the distance measure between Pr and Pr' is given by:*

$$D(Pr, Pr') = \ln \max_{i=1}^{n} \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} - \ln \min_{i=1}^{n} \frac{Pr'(\gamma_i)}{Pr(\gamma_i)}.$$

Using Theorem 8.7, we can easily compute the distance measure for revisions based on Jeffrey's rule and Pearl's method.

[5] Wagner showed not only that the representation of soft evidence using Bayes factors is sufficient for commutativity, but in a large number of cases, necessary.

Corollary 8.1 *If Pr' is obtained from Pr by applying Jeffrey's rule, given soft evidence specified by the set of posterior probabilities $Pr'(\gamma_i) = \rho_i$ for $i = 1, \ldots, n$, the distance measure between Pr and Pr' is given by:*

$$D(Pr, Pr') = \ln \max_{i=1}^{n} \frac{\rho_i}{Pr(\gamma_i)} - \ln \min_{i=1}^{n} \frac{\rho_i}{Pr(\gamma_i)}.$$

Corollary 8.2 *If $Pr(\cdot \mid \eta)$ is obtained from Pr by applying Pearl's method, given virtual evidence η specified by likelihood ratios $\lambda_1, \ldots, \lambda_n$, the distance measure between Pr and $Pr(\cdot \mid \eta)$ is given by:*

$$D(Pr, Pr(\cdot \mid \eta)) = \ln \max_{i=1}^{n} \lambda_i - \ln \min_{i=1}^{n} \lambda_i.$$

The significance of Corollaries 8.1 and 8.2 is that we can compute the distance measure easily in both cases. For Jeffrey's rule, we can compute the distance measure by knowing only the prior and posterior probabilities of events $\gamma_1, \ldots, \gamma_n$. For Pearl's method, we can compute the distance measure by knowing only the likelihood ratios $\lambda_1, \ldots, \lambda_n$. For both revision methods, the distance measure can be computed in constant time from the soft evidence, and we can guarantee a bound on amount of the belief change due to the fact that they both obey the principle of probability kinematics, without explicitly knowing the initial and revised distributions.

We close this section by showing that the principle of probability kinematics is optimal in a very precise sense: it commits to a probability distribution that minimizes the distance measure.

Theorem 8.8 *The distribution Pr' obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$ is optimal in the following sense. Among all possible distributions that agree with Pr' on the probabilities of events $\gamma_1, \ldots, \gamma_n$, Pr' is the closest to Pr according to the distance measure defined in Definition 4.1.*

CHAPTER 9

Conclusion

We have studied many different aspects of sensitivity analysis of probabilistic graphical models throughout this work. As a conclusion, we provide a summary of the contributions of this work:

- We introduced procedures and complexity results for changing network parameters to ensure certain query constraints, such as $Pr(y \mid \mathbf{e}) \geq \kappa$, for cases of single Bayesian network parameters (Chapter 2), multiple Bayesian network parameters (Chapter 5), and Markov network parameters (Chapter 7). We found that computing solutions for single parameter changes, or for parameter changes in a single table (CPTs for Bayesian networks, clique tables for Markov networks), requires complexity that is asymptotically the same as performing basic inference. For the case of multiple parameter changes, we also developed a numerical procedure to find a near-optimal parameter change in the solution space which minimizes model disturbance. These procedures are integral to building and debugging networks by allowing experts to focus on relevant parameters when enforcing sanity checks on certain query results.

- We provided network-independent bounds on changes in queries due to arbitrary changes in network parameters, for both Bayesian networks (Chapter 3) and Markov networks (Chapter 7). We obtained the bounds by investigating the sensitivity of a query with respect to a parameter, and the bounds allowed us to provide guarantees on the amount of change in a query given the change in a parameter even if we do not know any information about the network. These results can help characterize conditions where parameter changes do or do not matter, and provide bounds on permissible parameter changes that can assure query robustness, such as limiting the absolute change in a certain query to a certain amount.

- We proposed a new distance measure for quantifying probabilistic belief changes (Chapter 4), which can be used to bound the amount of query change between two probability distributions. We did an extensive study of this distance measure by comparing it with existing measures such as KL-divergence, and found that they cannot provide a similar guarantee. We showed that this distance measure can also be easily computed for distributions induced by Bayesian networks (Chapter 4) and Markov networks (Chapter 7), where the distributions differ on only the parameters of a single table (CPTs for Bayesian networks, clique tables for Markov networks). We also applied this distance measure to belief revision.

- We provided algorithms and complexity results on the sensitivity of decisions made by naive Bayes classifiers (Chapter 6). We developed an algorithm that converts a naive Bayes classifier into an ordered decision diagram, which is a logical representation allowing us to perform reasoning for the purposes of sensitivity analysis in time polynomial in its size. Our theoretical results showed that the upper bound on the size of the ordered decision diagram is much less than the number of instances that we need to enumerate if we use a brute-force approach of reasoning, while our experimental results showed that for many networks, the actual sizes of the ordered decision diagrams are much less than the theoretical upper bound.

- We discussed the topic of belief revision, where we analyzed the sensitivity of global beliefs to local belief changes in terms of soft evidence (Chapter 8). We looked at how a state of belief should be revised when given soft evidence, by dividing the process into different aspects, and compared two popular methods, Jeffrey's rule and Pearl's method of virtual evidence, in terms of these aspects, and identified their similarities and differences. We also used the distance measure we proposed to bound the global belief change due to soft evidence once we commit to the belief revision principle of probability kinematics, which are obeyed by both Jeffrey's rule and Pearl's method of virtual evidence, where the measure can be computed from information about the soft evidence only.

Finally, we also point out some possible future directions for the research of sensitivity analysis of probabilistic graphical models:

- The general bounds on query changes due to parameter changes, given in Chapters 3 and 7, are useful when no information about the Bayesian or Markov network is given. However, when some partial information about the network is given, such as certain properties of the current state of belief, or the class of queries we are interested in, we may be able to find tighter bounds on the amount of query change as a result.

- The tuning of network parameters to enforce a query constraint was discussed in Chapter 2, where we introduced a procedure that finds single parameter changes, and in Chapter 5, where we introduced a local search procedure that finds optimal single CPT changes, and also the underlying theory of finding such changes involving multiple CPTs. However, due to the many combinations of multiple CPTs, we need an automated procedure that identifies relevant CPTs to check for solutions. Moreover, we need a local search procedure that finds an optimal solution in the solution space of multiple CPT changes.

- We introduced a distance measure in Chapter 4 that can be considered as a worst-case bound, compared with KL-divergence that can be considered as an average-case bound. This distance measure can potentially be applied in fields where KL-divergence is commonly used, such as approximate reasoning and machine learning, where it can provide worse-case bounds on the query results.

- We focused our study on the sensitivity of decisions to the domain of naive Bayes classifiers in Chapter 6, by converting them into ordered decision diagrams. In the future, we would like to expand beyond naive Bayes classifiers. In particular, classifiers induced from Tree Augmented Naive Bayes networks (TANs) and Augmented Naive Bayes networks (ANBs), which are both derivatives of naive Bayes networks where directed edges are added between attributes, are good candidates because they are commonly used. Ultimately, we would like to provide an algorithm to convert a general Bayesian network classifier into a logical representation.

- Sensitivity analysis of networks with continuous variables, and sensitivity analysis of other probabilistic graphical models such as influence diagrams, are natural next steps for this research.

APPENDIX A

Proofs of Theorems

A.1 Proofs of Theorems in Chapter 2

Proof of Theorem 2.1 The partial derivative of $Pr(\mathbf{e})$ with respect to the network parameter $\theta_{x|\mathbf{u}}$ is given by [RBK95]:[1]

$$\frac{\partial Pr(\mathbf{e})}{\partial \theta_{x|\mathbf{u}}} = \frac{Pr(\mathbf{e}, x, \mathbf{u})}{\theta_{x|\mathbf{u}}},$$

if $\theta_{x|\mathbf{u}} \neq 0$, and:

$$\frac{\partial Pr(\mathbf{e})}{\partial \theta_{\bar{x}|\mathbf{u}}} = \frac{Pr(\mathbf{e}, \bar{x}, \mathbf{u})}{\theta_{\bar{x}|\mathbf{u}}},$$

if $\theta_{\bar{x}|\mathbf{u}} \neq 0$. Because $\theta_{x|\mathbf{u}} = \tau_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}} = 1 - \tau_{x|\mathbf{u}}$, we have:

$$\begin{aligned}\frac{\partial Pr(\mathbf{e})}{\partial \tau_{x|\mathbf{u}}} &= \frac{\partial Pr(\mathbf{e})}{\partial \theta_{x|\mathbf{u}}} - \frac{\partial Pr(\mathbf{e})}{\partial \theta_{\bar{x}|\mathbf{u}}} \\ &= \frac{Pr(\mathbf{e}, x, \mathbf{u})}{\theta_{x|\mathbf{u}}} - \frac{Pr(\mathbf{e}, \bar{x}, \mathbf{u})}{\theta_{\bar{x}|\mathbf{u}}},\end{aligned}$$

if $\theta_{x|\mathbf{u}} \neq 0$ and $\theta_{\bar{x}|\mathbf{u}} \neq 0$.□

A.2 Proofs of Theorems in Chapter 3

Proof of Theorem 3.1 We assume that X is a binary variable. The case where X is multi-valued can be easily dealt with by grouping all values $x^* \neq x$

[1] We allow the notations $\partial Pr(\mathbf{e})/\partial \theta_{x|\mathbf{u}}$ and $\partial Pr(\mathbf{e})/\partial \theta_{\bar{x}|\mathbf{u}}$ by assuming $Pr(\mathbf{e})$ as a function of $\theta_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}}$, even though it is not allowed in Bayesian networks to change only $\theta_{x|\mathbf{u}}$ or $\theta_{\bar{x}|\mathbf{u}}$.

into the pseudo-value $\bar{x}$. The partial derivative of $Pr(y \mid \mathbf{e})$ with respect to the parameter $\theta_{x|\mathbf{u}}$ is equal to [Dar03]:[2]

$$\frac{\partial Pr(y \mid \mathbf{e})}{\partial \theta_{x|\mathbf{u}}} = \frac{Pr(y, x, \mathbf{u} \mid \mathbf{e}) - Pr(x, \mathbf{u} \mid \mathbf{e}) Pr(y \mid \mathbf{e})}{Pr(x \mid \mathbf{u})}.$$

Since $Pr(y, x, \mathbf{u} \mid \mathbf{e}) \leq Pr(y \mid \mathbf{e})$ and $Pr(y, x, \mathbf{u} \mid \mathbf{e}) \leq Pr(x, \mathbf{u} \mid \mathbf{e})$, we have:

$$\begin{aligned}
\frac{\partial Pr(y \mid \mathbf{e})}{\partial \theta_{x|\mathbf{u}}} &\leq \frac{Pr(y, x, \mathbf{u} \mid \mathbf{e}) - Pr(y, x, \mathbf{u} \mid \mathbf{e}) Pr(y \mid \mathbf{e})}{Pr(x \mid \mathbf{u})} \\
&= \frac{Pr(y, x, \mathbf{u} \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})} \\
&\leq \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})}.
\end{aligned}$$

Therefore, the partial derivative of $Pr(y \mid \mathbf{e})$ with respect to the meta-parameter $\tau_{x|\mathbf{u}}$ is upper-bounded by:

$$\begin{aligned}
\frac{\partial Pr(y \mid \mathbf{e})}{\partial \tau_{x|\mathbf{u}}} &= \frac{\partial Pr(y \mid \mathbf{e})}{\partial \theta_{x|\mathbf{u}}} - \frac{\partial Pr(y \mid \mathbf{e})}{\partial \theta_{\bar{x}|\mathbf{u}}} \\
&\leq \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})} + \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{1 - Pr(x \mid \mathbf{u})} \\
&= Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e})) \left(\frac{1}{Pr(x \mid \mathbf{u})} + \frac{1}{1 - Pr(x \mid \mathbf{u})} \right) \\
&= \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})\,(1 - Pr(x \mid \mathbf{u}))}.
\end{aligned}$$

To find the lower bound on the derivative, we note that $Pr(y \mid \mathbf{e}) = 1 - Pr(\bar{y} \mid \mathbf{e})$, and thus:

$$\begin{aligned}
\frac{\partial Pr(y \mid \mathbf{e})}{\partial \tau_{x|\mathbf{u}}} &= -\frac{\partial Pr(\bar{y} \mid \mathbf{e})}{\partial \tau_{x|\mathbf{u}}} \\
&\geq -\frac{Pr(\bar{y} \mid \mathbf{e})\,(1 - Pr(\bar{y} \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})\,(1 - Pr(x \mid \mathbf{u}))} \\
&= -\frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})\,(1 - Pr(x \mid \mathbf{u}))}.
\end{aligned}$$

[2] We again allow the notations $\partial Pr(y \mid \mathbf{e})/\partial \theta_{x|\mathbf{u}}$ and $\partial Pr(y \mid \mathbf{e})/\partial \theta_{\bar{x}|\mathbf{u}}$ by assuming $Pr(y \mid \mathbf{e})$ as a function of $\theta_{x|\mathbf{u}}$ and $\theta_{\bar{x}|\mathbf{u}}$, even though it is not allowed in Bayesian networks to change only $\theta_{x|\mathbf{u}}$ or $\theta_{\bar{x}|\mathbf{u}}$.

Combining the upper bound and the lower bound, we have:

$$\left|\frac{\partial Pr(y \mid \mathbf{e})}{\partial \tau_{x|\mathbf{u}}}\right| \leq \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})\,(1 - Pr(x \mid \mathbf{u}))}.\Box$$

Proof of Theorem 3.2 Because $\Delta\tau_{x|\mathbf{u}}$ is infinitesimal, from Theorem 3.1, we have:

$$\begin{aligned}\left|\frac{\Delta Pr(y \mid \mathbf{e})}{\Delta \tau_{x|\mathbf{u}}}\right| &\simeq \left|\frac{\partial Pr(y \mid \mathbf{e})}{\partial \tau_{x|\mathbf{u}}}\right| \\ &\leq \frac{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))}{Pr(x \mid \mathbf{u})\,(1 - Pr(x \mid \mathbf{u}))}.\end{aligned}$$

Arranging the terms, we have:

$$\begin{aligned}\left|\frac{\Delta Pr(y \mid \mathbf{e})}{Pr(y \mid \mathbf{e})}\right| &\leq \frac{1 - Pr(y \mid \mathbf{e})}{1 - Pr(x \mid \mathbf{u})}\left|\frac{\Delta \tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}}\right| \\ &\leq \frac{1}{.5}\left|\frac{\Delta \tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}}\right| \\ &= 2\left|\frac{\Delta \tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}}\right|,\end{aligned}$$

since $Pr(x \mid \mathbf{u}) = \tau_{x|\mathbf{u}} \leq .5.\Box$

Proof of Theorem 3.3 We obtain this result by integrating the bound on $\partial Pr(y \mid \mathbf{e})/\partial\tau_{x|\mathbf{u}}$ given in Theorem 3.1. In particular, if we change $\tau_{x|\mathbf{u}}$ to $\tau'_{x|\mathbf{u}} > \tau_{x|\mathbf{u}}$, and consequently $Pr(y \mid \mathbf{e})$ changes to $Pr'(y \mid \mathbf{e})$, we can separate the variables in the upper bound on the derivative in Theorem 3.1, integrate over the intervals, and yield:

$$\int_{Pr(y|\mathbf{e})}^{Pr'(y|\mathbf{e})} \frac{dPr(y \mid \mathbf{e})}{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))} \leq \int_{\tau_{x|\mathbf{u}}}^{\tau'_{x|\mathbf{u}}} \frac{d\tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}\left(1 - \tau_{x|\mathbf{u}}\right)}.$$

This gives us the solution:

$$\ln\frac{Pr'(y \mid \mathbf{e})}{1 - Pr'(y \mid \mathbf{e})} - \ln\frac{Pr(y \mid \mathbf{e})}{1 - Pr(y \mid \mathbf{e})} \leq \ln\frac{\tau'_{x|\mathbf{u}}}{1 - \tau'_{x|\mathbf{u}}} - \ln\frac{\tau_{x|\mathbf{u}}}{1 - \tau_{x|\mathbf{u}}},$$

which is equivalent to:

$$\ln O'(y \mid \mathbf{e}) - \ln O(y \mid \mathbf{e}) \leq \ln O'(x \mid \mathbf{u}) - \ln O(x \mid \mathbf{u}),$$

and after taking exponentials, we have:

$$\frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \leq \frac{O'(x \mid \mathbf{u})}{O(x \mid \mathbf{u})}.$$

Similarly, we can separate the variables in the lower bound on the derivative in Theorem 3.1, integrate over the intervals, and yield:

$$\int_{Pr(y|\mathbf{e})}^{Pr'(y|\mathbf{e})} \frac{dPr(y \mid \mathbf{e})}{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))} \geq -\int_{\tau_{x|\mathbf{u}}}^{\tau'_{x|\mathbf{u}}} \frac{d\tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}\left(1 - \tau_{x|\mathbf{u}}\right)}.$$

This gives us the solution:

$$\ln \frac{Pr'(y \mid \mathbf{e})}{1 - Pr'(y \mid \mathbf{e})} - \ln \frac{Pr(y \mid \mathbf{e})}{1 - Pr(y \mid \mathbf{e})} \geq -\ln \frac{\tau'_{x|\mathbf{u}}}{1 - \tau'_{x|\mathbf{u}}} + \ln \frac{\tau_{x|\mathbf{u}}}{1 - \tau_{x|\mathbf{u}}},$$

which is equivalent to:

$$\ln O'(y \mid \mathbf{e}) - \ln O(y \mid \mathbf{e}) \leq -\ln O'(x \mid \mathbf{u}) + \ln O(x \mid \mathbf{u}),$$

and after taking exponentials, we have:

$$\frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \geq \frac{O(x \mid \mathbf{u})}{O'(x \mid \mathbf{u})}.$$

Therefore, we have the following inequality if $\tau'_{x|\mathbf{u}} > \tau_{x|\mathbf{u}}$:

$$\frac{O(x \mid \mathbf{u})}{O'(x \mid \mathbf{u})} \leq \frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \leq \frac{O'(x \mid \mathbf{u})}{O(x \mid \mathbf{u})}.$$

On the other hand, if we change $\tau_{x|\mathbf{u}}$ to $\tau'_{x|\mathbf{u}} < \tau_{x|\mathbf{u}}$, we can instead integrate from $\tau'_{x|\mathbf{u}}$ to $\tau_{x|\mathbf{u}}$. The integrals satisfy these two inequalities:

$$\begin{aligned}
\int_{Pr'(y|\mathbf{e})}^{Pr(y|\mathbf{e})} \frac{dPr(y \mid \mathbf{e})}{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))} &\leq \int_{\tau'_{x|\mathbf{u}}}^{\tau_{x|\mathbf{u}}} \frac{d\tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}\left(1 - \tau_{x|\mathbf{u}}\right)};\\
\int_{Pr'(y|\mathbf{e})}^{Pr(y|\mathbf{e})} \frac{dPr(y \mid \mathbf{e})}{Pr(y \mid \mathbf{e})\,(1 - Pr(y \mid \mathbf{e}))} &\geq -\int_{\tau'_{x|\mathbf{u}}}^{\tau_{x|\mathbf{u}}} \frac{d\tau_{x|\mathbf{u}}}{\tau_{x|\mathbf{u}}\left(1 - \tau_{x|\mathbf{u}}\right)}.
\end{aligned}$$

We can solve for them similarly and get the following result:

$$\frac{O'(x \mid \mathbf{u})}{O(x \mid \mathbf{u})} \leq \frac{O'(y \mid \mathbf{e})}{O(y \mid \mathbf{e})} \leq \frac{O(x \mid \mathbf{u})}{O'(x \mid \mathbf{u})}.$$

Combining the results for both $\tau'_{x|\mathbf{u}} > \tau_{x|\mathbf{u}}$ and $\tau'_{x|\mathbf{u}} < \tau_{x|\mathbf{u}}$, we have:

$$|\ln O'(y \mid \mathbf{e}) - \ln O(y \mid \mathbf{e})| \leq |\ln O'(x \mid \mathbf{u}) - \ln O(x \mid \mathbf{u})| .\square$$

A.3 Proofs of Theorems in Chapter 4

Proof of Theorem 4.1 The distance measure defined in Definition 4.1 satisfies these three properties:

Positiveness Obviously, we have $D(Pr, Pr') \geq 0$ and $D(Pr, Pr) = 0$. Since $\sum_\omega Pr(\omega) = \sum_\omega Pr'(\omega) = 1$, we have $\max_\omega (Pr'(\omega)/Pr(\omega)) \geq 1$ and $\min_\omega (Pr'(\omega)/Pr(\omega)) \leq 1$. Moreover, when $D(Pr, Pr') = 0$, we must have $\max_\omega (Pr'(\omega)/Pr(\omega)) = \min_\omega (Pr'(\omega)/Pr(\omega)) = 1$. Therefore, $Pr(\omega) = Pr'(\omega)$ for all ω, and thus $Pr = Pr'$.

Symmetry Since we have $\max_\omega (Pr'(\omega)/Pr(\omega)) = \min_\omega (Pr(\omega)/Pr'(\omega))$ and $\min_\omega (Pr'(\omega)/Pr(\omega)) = \max_\omega (Pr(\omega)/Pr'(\omega))$, we can easily prove that $D(Pr, Pr') = D(Pr', Pr)$.

Triangle inequality Given two worlds ω_i and ω_j such that $Pr''(\omega_i)/Pr(\omega_i) = \max_\omega (Pr''(\omega)/Pr(\omega))$ and $Pr''(\omega_j)/Pr(\omega_j) = \min_\omega (Pr''(\omega)/Pr(\omega))$, and because $\ln(Pr''(\omega)/Pr(\omega)) = \ln(Pr''(\omega)/Pr'(\omega)) - \ln(Pr'(\omega)/Pr(\omega))$, we have:

$$\begin{aligned} & D(Pr, Pr'') \\ = \;& \ln \max_\omega \frac{Pr''(\omega)}{Pr(\omega)} - \ln \min_\omega \frac{Pr''(\omega)}{Pr(\omega)} \end{aligned}$$

$$\begin{aligned}
&= \ln \frac{Pr''(\omega_i)}{Pr(\omega_i)} - \ln \frac{Pr''(\omega_j)}{Pr(\omega_j)} \\
&= \left(\ln \frac{Pr''(\omega_i)}{Pr'(\omega_i)} - \ln \frac{Pr'(\omega_i)}{Pr(\omega_i)}\right) - \left(\ln \frac{Pr''(\omega_j)}{Pr'(\omega_j)} - \ln \frac{Pr'(\omega_j)}{Pr(\omega_j)}\right) \\
&= \ln \frac{Pr'(\omega_j)}{Pr(\omega_j)} - \ln \frac{Pr'(\omega_i)}{Pr(\omega_i)} + \ln \frac{Pr''(\omega_i)}{Pr'(\omega_i)} - \ln \frac{Pr''(\omega_j)}{Pr'(\omega_j)} \\
&\leq \ln \max_{\omega} \frac{Pr'(\omega)}{Pr(\omega)} - \ln \min_{\omega} \frac{Pr'(\omega)}{Pr(\omega)} + \ln \max_{\omega} \frac{Pr''(\omega)}{Pr'(\omega)} - \ln \min_{\omega} \frac{Pr''(\omega)}{Pr'(\omega)} \\
&= D(Pr, Pr') + D(Pr', Pr''). \Box
\end{aligned}$$

Proof of Theorem 4.2 If distributions Pr and Pr' do not have the same support, we have $D(Pr, Pr') = \infty$, and thus $-\infty = e^{-D(Pr,Pr')} \leq O'(\alpha \mid \beta)/O(\alpha \mid \beta) \leq e^{D(Pr,Pr')} = \infty$. Otherwise, the odds ratio $O'(\alpha \mid \beta)/O(\alpha \mid \beta)$ can be expressed as:

$$\begin{aligned}
\frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} &= \frac{\frac{Pr'(\alpha|\beta)}{Pr'(\bar{\alpha}|\beta)}}{\frac{Pr(\alpha|\beta)}{Pr(\bar{\alpha}|\beta)}} \\
&= \frac{\frac{Pr'(\alpha,\beta)}{Pr'(\bar{\alpha},\beta)}}{\frac{Pr(\alpha,\beta)}{Pr(\bar{\alpha},\beta)}} \\
&= \frac{\frac{\sum_{\omega \models \alpha,\beta} Pr'(\omega)}{\sum_{\omega \models \bar{\alpha},\beta} Pr'(\omega)}}{\frac{\sum_{\omega \models \alpha,\beta} Pr(\omega)}{\sum_{\omega \models \bar{\alpha},\beta} Pr(\omega)}} \\
&= \frac{\frac{\sum_{\omega \models \alpha,\beta} Pr'(\omega)}{\sum_{\omega \models \alpha,\beta} Pr(\omega)}}{\frac{\sum_{\omega \models \bar{\alpha},\beta} Pr'(\omega)}{\sum_{\omega \models \bar{\alpha},\beta} Pr(\omega)}}.
\end{aligned}$$

We can now obtain the upper bound on the odds ratio:

$$\begin{aligned}
\frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} &\leq \frac{\max_{\omega \models \alpha,\beta} \frac{Pr'(\omega)}{Pr(\omega)}}{\min_{\omega \models \bar{\alpha},\beta} \frac{Pr'(\omega)}{Pr(\omega)}} \\
&\leq \frac{\max_{\omega} \frac{Pr'(\omega)}{Pr(\omega)}}{\min_{\omega} \frac{Pr'(\omega)}{Pr(\omega)}} \\
&= e^{D(Pr,Pr')}.
\end{aligned}$$

Similarly, we can also obtain the lower bound on the odds ratio:

$$\begin{aligned}\frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} &\geq \frac{\min_{\omega \models \alpha,\beta} \frac{Pr'(\omega)}{Pr(\omega)}}{\max_{\omega \models \bar{\alpha},\beta} \frac{Pr'(\omega)}{Pr(\omega)}} \\ &\geq \frac{\min_{\omega} \frac{Pr'(\omega)}{Pr(\omega)}}{\max_{\omega} \frac{Pr'(\omega)}{Pr(\omega)}} \\ &= e^{-D(Pr,Pr')}.\end{aligned}$$

Therefore, we have $e^{-D(Pr,Pr')} \leq O'(\alpha \mid \beta)/O(\alpha \mid \beta) \leq e^{D(Pr,Pr')}$. If both $O'(\alpha \mid \beta)$ and $O(\alpha \mid \beta)$ take on either 0 or ∞, the theorem still holds because $0/0 \overset{\text{def}}{=} 1$ and $\infty/\infty \overset{\text{def}}{=} 1$. Finally, to prove the tightness of the bound, we note that for every pair of distributions Pr and Pr', there are events $\alpha = \omega_i$ and $\beta = \omega_i \vee \omega_j$, where $\omega_i = \arg\max_{\omega} (Pr'(\omega)/Pr(\omega))$ and $\omega_j = \arg\min_{\omega} (Pr'(\omega)/Pr(\omega))$, such that:

$$\begin{aligned}\frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} &= \frac{\frac{\sum_{\omega \models \alpha,\beta} Pr'(\omega)}{\sum_{\omega \models \alpha,\beta} Pr(\omega)}}{\frac{\sum_{\omega \models \bar{\alpha},\beta} Pr'(\omega)}{\sum_{\omega \models \bar{\alpha},\beta} Pr(\omega)}} \\ &= \frac{\frac{Pr'(\omega_i)}{Pr(\omega_i)}}{\frac{Pr'(\omega_j)}{Pr(\omega_j)}} \\ &= \frac{\max_{\omega} \frac{Pr'(\omega)}{Pr(\omega)}}{\min_{\omega} \frac{Pr'(\omega)}{Pr(\omega)}} \\ &= e^{D(Pr,Pr')}.\end{aligned}$$

Similarly, we can get $O'(\bar{\alpha} \mid \beta)/O(\bar{\alpha} \mid \beta) = e^{-D(Pr,Pr')}$.□

Proof of Theorem 4.3 To obtain the largest probability change for the conditional event $\alpha \mid \beta$ given some KL-divergence, the new probability distribution Pr' needs to satisfy the following three properties:

- For all $\omega \models \alpha, \beta$, $Pr'(\omega) = Pr(\omega)(Pr'(\alpha \mid \beta)/Pr(\alpha \mid \beta))$;

- For all $\omega \models \bar{\alpha}, \beta$, $Pr'(\omega) = Pr(\omega)(Pr'(\bar{\alpha} \mid \beta)/Pr(\bar{\alpha} \mid \beta))$;

- For all $\omega \models \bar{\beta}$, $Pr'(\omega) = Pr(\omega)$, i.e., $Pr'(\bar{\beta}) = Pr(\bar{\beta})$.

Under these three conditions, we have the following equality:

$$\begin{aligned} KL(Pr, Pr') &= -Pr(\alpha, \beta) \ln \frac{Pr'(\alpha, \beta)}{Pr(\alpha, \beta)} - Pr(\bar{\alpha}, \beta) \ln \frac{Pr'(\bar{\alpha}, \beta)}{Pr(\bar{\alpha}, \beta)} \\ &= -Pr(\beta) \left(Pr(\alpha \mid \beta) \ln \frac{Pr'(\alpha \mid \beta)}{Pr(\alpha \mid \beta)} + Pr(\bar{\alpha} \mid \beta) \ln \frac{Pr'(\bar{\alpha} \mid \beta)}{Pr(\bar{\alpha} \mid \beta)} \right). \end{aligned}$$

Since this is the case where we get the largest probability change, the following inequality must be satisfied:

$$\begin{aligned} & KL(Pr, Pr') \\ \geq \;& -Pr(\beta) \left(Pr(\alpha \mid \beta) \ln \frac{Pr'(\alpha \mid \beta)}{Pr(\alpha \mid \beta)} + (1 - Pr(\alpha \mid \beta)) \ln \frac{1 - Pr'(\alpha \mid \beta)}{1 - Pr(\alpha \mid \beta)} \right). \end{aligned}$$

We can now substitute in $Pr'(\alpha \mid \beta) = O'(\alpha \mid \beta)/(O'(\alpha \mid \beta) + 1)$ and $Pr(\alpha \mid \beta) = O(\alpha \mid \beta)/(O(\alpha \mid \beta) + 1)$, and after simplifying, we get:

$$KL(Pr, Pr') \geq Pr(\beta) \left(\ln \frac{O'(\alpha \mid \beta) + 1}{O(\alpha \mid \beta) + 1} - \frac{O(\alpha \mid \beta)}{O(\alpha \mid \beta) + 1} \ln \frac{O'(\alpha \mid \beta)}{O(\alpha \mid \beta)} \right) . \square$$

Proof of Theorem 4.4 To prove this theorem, we first establish the following two lemmas.

Lemma A.1 *Given two distributions* Pr *and* Pr'*, and a set of worlds* $S^{>}$*, where* $\forall_{\omega \in S^{>}} Pr'(\omega) > Pr(\omega)$*, let* $r^{>} = \max_{\omega \in S^{>}} (Pr'(\omega)/Pr(\omega))$*. We have the following inequality:*

$$\sum_{\omega \in S^{>}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \leq -\frac{Pr'(S^{>}) - Pr(S^{>})}{r^{>} - 1} \ln r^{>}. \tag{A.1}$$

Lemma A.2 *Given two distributions Pr and Pr', and a set of worlds $S^{<}$, where $\forall_{\omega \in S^{<}} Pr'(\omega) < Pr(\omega)$, let $r^{<} = \min_{\omega \in S^{<}} (Pr'(\omega)/Pr(\omega))$. We have the following inequality:*

$$\sum_{\omega \in S^{<}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \leq -\frac{Pr'(S^{<}) - Pr(S^{<})}{r^{<} - 1} \ln r^{<}. \qquad \text{(A.2)}$$

Proofs of Lemmas A.1 and A.2 We use induction to prove Lemma A.1. Let $S^{>} = \{\omega\}$, where $Pr'(\omega) > Pr(\omega)$, and $r^{>} = Pr'(\omega)/Pr(\omega)$. We have:

$$\begin{aligned} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} &= -\frac{(r^{>} - 1)\, Pr(\omega)}{r^{>} - 1} \ln r^{>} \\ &= -\frac{Pr'(\omega) - Pr(\omega)}{r^{>} - 1} \ln r^{>}. \end{aligned}$$

Therefore, Inequality A.1 is true for $S^{>} = \{\omega\}$. Assume that we are now given mutually exclusive sets $S_1^{>}, \ldots, S_n^{>}$, where for all i, $\forall_{\omega \in S_i^{>}} Pr'(\omega) > Pr(\omega)$, and Inequality A.1 is true, i.e., if $r_i^{>} = \max_{\omega \in S_i^{>}} (Pr'(\omega)/Pr(\omega))$, we have:

$$\sum_{\omega \in S_i^{>}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \leq -\frac{Pr'(S_i^{>}) - Pr(S_i^{>})}{r_i^{>} - 1} \ln r_i^{>}.$$

We want to prove that Inequality A.1 is also true for $S^{>} = \bigcup_{i=1}^{n} S_i^{>}$. Let $r^{>} = \max_{\omega \in S^{>}} (Pr'(\omega)/Pr(\omega))$. For all i, $Pr'(S_i^{>}) - Pr(S_i^{>}) > 0$, and also $r^{>} \geq r_i^{>} > 1$, which gives us:

$$-\frac{\ln r^{>}}{r^{>} - 1} \geq -\frac{\ln r_i^{>}}{r_i^{>} - 1}.$$

Therefore, we have:

$$\begin{aligned} \sum_{\omega \in S^{>}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} &= \sum_{i=1}^{n} \sum_{\omega \in S_i^{>}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \\ &\leq \sum_{i=1}^{n} -\frac{Pr'(S_i^{>}) - Pr(S_i^{>})}{r_i^{>} - 1} \ln r_i^{>} \\ &\leq \sum_{i=1}^{n} -\frac{Pr'(S_i^{>}) - Pr(S_i^{>})}{r^{>} - 1} \ln r^{>} \\ &= -\frac{Pr'(S^{>}) - Pr(S^{>})}{r^{>} - 1} \ln r^{>}, \end{aligned}$$

which proves that Inequality A.1 is also true for $S^{>} = \bigcup_{i=1}^{n} S_i^{>}$. Therefore, Lemma A.1 is true by induction. Lemma A.2 can be proved similarly.$\square$

We now proceed with the proof of Theorem 4.4. Given distributions Pr and Pr', we partition all worlds into three subsets:

- $S^{>}$, where $\forall_{\omega \in S^{>}} Pr'(\omega) > Pr(\omega)$;
- $S^{<}$, where $\forall_{\omega \in S^{<}} Pr'(\omega) < Pr(\omega)$;
- $S^{=}$, where $\forall_{\omega \in S^{=}} Pr'(\omega) = Pr(\omega)$.

It is obvious that we must have $Pr'(S^{<}) - Pr(S^{<}) = -(Pr'(S^{>}) - Pr(S^{>}))$. If we are given $r^{>} = \max_{\omega} (Pr'(\omega)/Pr(\omega))$ and $r^{<} = \min_{\omega} (Pr'(\omega)/Pr(\omega))$, from Lemmas A.1 and A.2, the KL-divergence between Pr and Pr' is bounded by:

$$\begin{aligned}
&KL(Pr, Pr') \\
&= \sum_{\omega} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \\
&= \sum_{\omega \in S^{>}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} + \sum_{\omega \in S^{<}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} + \sum_{\omega \in S^{=}} -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \\
&\leq -\frac{Pr'(S^{>}) - Pr(S^{>})}{r^{>} - 1} \ln r^{>} - \frac{Pr'(S^{<}) - Pr(S^{<})}{r^{<} - 1} \ln r^{<} \\
&= (Pr'(S^{>}) - Pr(S^{>})) \left(-\frac{\ln r^{>}}{r^{>} - 1} + \frac{\ln r^{<}}{r^{<} - 1} \right).
\end{aligned}$$

We first maximize $Pr'(S^{>}) - Pr(S^{>})$ with respect to $r^{>}$ and $r^{<}$. The constraints we have are: $Pr'(S^{>})/Pr(S^{>}) \leq r^{>}$, $Pr'(S^{<})/Pr(S^{<}) \geq r^{<}$, and $Pr(S^{>}) + Pr(S^{<}) = Pr'(S^{>}) + Pr'(S^{<}) \leq 1$. The solution we get is:

$$Pr'(S^{>}) - Pr(S^{>}) \leq -\frac{(r^{>} - 1)(r^{<} - 1)}{r^{>} - r^{<}}.$$

Therefore, we have:

$$\begin{aligned}
KL(Pr, Pr') &\leq \left(-\frac{(r^{>} - 1)(r^{<} - 1)}{r^{>} - r^{<}} \right) \left(-\frac{\ln r^{>}}{r^{>} - 1} + \frac{\ln r^{<}}{r^{<} - 1} \right) \\
&= \frac{(r^{<} - 1) \ln r^{>} - (r^{>} - 1) \ln r^{<}}{r^{>} - r^{<}}.
\end{aligned}$$

We now maximize this term with respect to $d = D(Pr, Pr') = \ln r^{>} - \ln r^{<}$. The maximum value is attained at $r^{>} = e^d d/(e^d - 1)$ and $r^{<} = d/(e^d - 1)$. Therefore, the KL-divergence is bounded by:

$$
\begin{aligned}
KL(Pr, Pr') &\leq \frac{\left(\frac{d}{e^d-1} - 1\right) \ln \frac{e^d d}{e^d-1} - \left(\frac{e^d d}{e^d-1} - 1\right) \ln \frac{d}{e^d-1}}{\frac{e^d d}{e^d-1} - \frac{d}{e^d-1}} \\
&= \frac{\left(d - \left(e^d - 1\right)\right)\left(d + \ln \frac{d}{e^d-1}\right) - \left(e^d d - \left(e^d - 1\right)\right) \ln \frac{d}{e^d-1}}{\left(e^d - 1\right) d} \\
&= \frac{\left(d - \left(e^d - 1\right)\right) d - \left(e^d - 1\right) d \ln \frac{d}{e^d-1}}{\left(e^d - 1\right) d} \\
&= \frac{d}{e^d - 1} - 1 - \ln \frac{d}{e^d - 1}. \square
\end{aligned}
$$

Proof of Theorem 4.5 If $\gamma_1, \ldots, \gamma_n$ is a set of mutually exclusive and exhaustive events, we have:

$$
\begin{aligned}
\sum_i Pr(\gamma_i) \ln B_{Pr', Pr}(\alpha : \gamma_i) &= \sum_i Pr(\gamma_i) \ln \frac{\frac{Pr'(\alpha)}{Pr(\alpha)}}{\frac{Pr'(\gamma_i)}{Pr(\gamma_i)}} \\
&= \sum_i Pr(\gamma_i) \ln \frac{Pr'(\alpha)}{Pr(\alpha)} - \sum_i Pr(\gamma_i) \ln \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} \\
&= \ln \frac{Pr'(\alpha)}{Pr(\alpha)} \sum_i Pr(\gamma_i) - \sum_i Pr(\gamma_i) \ln \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} \\
&= \ln \frac{Pr'(\alpha)}{Pr(\alpha)} - \sum_i Pr(\gamma_i) \ln \frac{Pr'(\gamma_i)}{Pr(\gamma_i)}.
\end{aligned}
$$

We also note the following inequality about KL-divergence:

$$
\begin{aligned}
KL(Pr, Pr') &= \sum_\omega -Pr(\omega) \ln \frac{Pr'(\omega)}{Pr(\omega)} \\
&\geq \sum_i -Pr(\gamma_i) \ln \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} \\
&\geq 0.
\end{aligned}
$$

Therefore, we have:

$$
\ln \frac{Pr'(\alpha)}{Pr(\alpha)} \leq \sum_i Pr(\gamma_i) \ln B_{Pr', Pr}(\alpha : \gamma_i) \leq \ln \frac{Pr'(\alpha)}{Pr(\alpha)} + KL(Pr, Pr'),
$$

or:

$$0 \leq \sum_i Pr(\gamma_i) \ln B_{Pr',Pr}(\alpha : \gamma_i) - \ln \frac{Pr'(\alpha)}{Pr(\alpha)} \leq KL(Pr, Pr').$$

If instead of an arbitrary partition $\gamma_1, \ldots, \gamma_n$, we are given the set of worlds ω, we get the following equality relation:

$$\sum_\omega Pr(\omega) \ln B_{Pr',Pr}(\alpha : \omega) - \ln \frac{Pr'(\alpha)}{Pr(\alpha)} = KL(Pr, Pr'). \Box$$

Proof of Theorem 4.6 To prove this theorem, we first establish the following lemma.

Lemma A.3 *Assume that we change parameter $\theta_{x|\mathbf{u}}$ to $\theta'_{x|\mathbf{u}}$ for every value x, and $Pr(\mathbf{u}) > 0$. For every x where $\theta'_{x|\mathbf{u}} > 0$ or $\theta_{x|\mathbf{u}} > 0$, there must exist some $\omega \models x, \mathbf{u}$ such that it satisfies the condition $Pr'(\omega)/Pr(\omega) = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$. For all other worlds ω that do not satisfy this condition, we must have $Pr'(\omega) = Pr(\omega)$, and thus $Pr'(\omega)/Pr(\omega) = 1$.*[3]

Proof of Lemma A.3 We first note that $Pr'(\mathbf{u}) = Pr(\mathbf{u}) > 0$. For any world ω, either $\omega \models \bar{\mathbf{u}}$, or $\omega \models x, \mathbf{u}$ for some x. We now consider the different cases of ω.

- If $\omega \models \bar{\mathbf{u}}$, we must have $Pr'(\omega) = Pr(\omega)$, since we are only changing parameters $\theta_{x|\mathbf{u}}$.
- If $\omega \models x, \mathbf{u}$, we consider four cases of x:
  - If $\theta'_{x|\mathbf{u}} = \theta_{x|\mathbf{u}} = 0$, we must have $Pr'(x, \mathbf{u}) = Pr(x, \mathbf{u}) = 0$. Therefore, for all worlds $\omega \models x, \mathbf{u}$, $Pr'(\omega) = Pr(\omega) = 0$.

[3] *Either $Pr'(\omega) = Pr(\omega) > 0$, and thus $Pr'(\omega)/Pr(\omega) = 1$, or $Pr'(\omega) = Pr(\omega) = 0$, and thus $Pr'(\omega)/Pr(\omega) \stackrel{\text{def}}{=} 1$.*

- If $\theta'_{x|\mathbf{u}} = 0$ and $\theta_{x|\mathbf{u}} > 0$, we must have $Pr'(x, \mathbf{u}) = 0$ and $Pr(x, \mathbf{u}) > 0$. Therefore, for all worlds $\omega \models x, \mathbf{u}$, either $Pr'(\omega) = Pr(\omega) = 0$; or $Pr'(\omega) = 0$ and $Pr(\omega) > 0$, giving us $Pr'(\omega)/Pr(\omega) = 0 = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$. Moreover, because $Pr(x, \mathbf{u}) > 0$, there must exist some $\omega \models x, \mathbf{u}$ such that $Pr(\omega) > 0$, and thus satisfying the condition $Pr'(\omega)/Pr(\omega) = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$.
- If $\theta'_{x|\mathbf{u}} > 0$ and $\theta_{x|\mathbf{u}} = 0$, we must have $Pr'(x, \mathbf{u}) > 0$ and $Pr(x, \mathbf{u}) = 0$. Therefore, for all worlds $\omega \models x, \mathbf{u}$, either $Pr'(\omega) = Pr(\omega) = 0$; or $Pr'(\omega) > 0$ and $Pr(\omega) = 0$, giving us $Pr'(\omega)/Pr(\omega) = \infty = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$. Moreover, because $Pr'(x, \mathbf{u}) > 0$, there must exist some $\omega \models x, \mathbf{u}$ such that $Pr'(\omega) > 0$, and thus satisfying the condition $Pr'(\omega)/Pr(\omega) = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$.
- If $\theta'_{x|\mathbf{u}} > 0$ and $\theta_{x|\mathbf{u}} > 0$, we must have $Pr'(x, \mathbf{u}) > 0$ and $Pr(x, \mathbf{u}) > 0$. Therefore, for all worlds $\omega \models x, \mathbf{u}$, either $Pr'(\omega) = Pr(\omega) = 0$; or $Pr'(\omega) > 0$ and $Pr(\omega) > 0$, giving us $Pr'(\omega)/Pr(\omega) = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$. Moreover, because $Pr'(x, \mathbf{u}) > 0$ and $Pr(x, \mathbf{u}) > 0$, there must exist some $\omega \models x, \mathbf{u}$ such that $Pr'(\omega) > 0$ and $Pr(\omega) > 0$, and thus satisfying the condition $Pr'(\omega)/Pr(\omega) = \theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}}$.

Therefore Lemma A.3 is correct.□

From Lemma A.3, we can prove that $\max_\omega (Pr'(\omega)/Pr(\omega)) = \max_x (\theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}})$ and $\min_\omega (Pr'(\omega)/Pr(\omega)) = \min_x (\theta'_{x|\mathbf{u}}/\theta_{x|\mathbf{u}})$. Therefore, we have $D(Pr, Pr') = D(\Theta_{X,\mathbf{u}}, \Theta'_{X,\mathbf{u}})$.□

Proof of Theorem 4.7 Let $\Theta'_{X|\mathbf{u}}$ be the distribution generated by the proportional scheme defined in Definition 2.1, which sets $\theta'_{x^*|\mathbf{u}} = (1 - \theta'_{x|\mathbf{u}})(\theta_{x^*|\mathbf{u}}/(1 - \theta_{x|\mathbf{u}}))$ for all $x^* \neq x$, and $\Theta''_{X|\mathbf{u}}$ be an arbitrary distribution with $\theta''_{x|\mathbf{u}} = \theta'_{x|\mathbf{u}}$. We want to prove that $D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}}) \geq D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}})$, by considering the following cases:

- If $\theta_{x|\mathbf{u}} = 1$ and $\theta''_{x|\mathbf{u}} = \theta'_{x|\mathbf{u}} < 1$, we must have $D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}}) = \infty \geq D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}})$.

- If $\theta_{x|\mathbf{u}} = 0$ and $\theta''_{x|\mathbf{u}} = \theta'_{x|\mathbf{u}} > 0$, we must have $D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}}) = \infty \geq D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}})$.

- If $\theta_{x|\mathbf{u}} \neq 0$ and $\theta_{x|\mathbf{u}} \neq 1$, we first consider the case where $\theta''_{x|\mathbf{u}} = \theta'_{x|\mathbf{u}} > \theta_{x|\mathbf{u}}$. If distributions $\Theta_{X|\mathbf{u}}$ and $\Theta''_{X|\mathbf{u}}$ do not have the same support, we have $D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}}) = \infty \geq D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}})$. Otherwise, we have the following inequality:

$$\begin{aligned}
D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}}) &\geq \ln \frac{\theta''_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \min_{x^* \neq x} \frac{\theta''_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} \\
&\geq \ln \frac{\theta''_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \frac{\sum_{x^* \neq x} \theta''_{x^*|\mathbf{u}}}{\sum_{x^* \neq x} \theta_{x^*|\mathbf{u}}} \\
&= \ln \frac{\theta''_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \frac{1 - \theta''_{x|\mathbf{u}}}{1 - \theta_{x|\mathbf{u}}} \\
&= \ln \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \frac{1 - \theta'_{x|\mathbf{u}}}{1 - \theta_{x|\mathbf{u}}} \\
&= D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}}).
\end{aligned}$$

  We can prove $D(\Theta_{X|\mathbf{u}}, \Theta''_{X|\mathbf{u}}) \geq D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}})$ for the case of $\theta''_{x|\mathbf{u}} = \theta'_{x|\mathbf{u}} < \theta_{x|\mathbf{u}}$ similarly.

Therefore, the proportional scheme defined in Definition 2.1 gives us the smallest distance measure, which is equal to:

$$\begin{aligned}
D(\Theta_{X|\mathbf{u}}, \Theta'_{X|\mathbf{u}}) &= \left| \ln \frac{\theta'_{x|\mathbf{u}}}{\theta_{x|\mathbf{u}}} - \ln \frac{1 - \theta'_{x|\mathbf{u}}}{1 - \theta_{x|\mathbf{u}}} \right| \\
&= \left| \ln \frac{\theta'_{x|\mathbf{u}}}{1 - \theta'_{x|\mathbf{u}}} - \ln \frac{\theta_{x|\mathbf{u}}}{1 - \theta_{x|\mathbf{u}}} \right| . \square
\end{aligned}$$

A.4 Proofs of Theorems in Chapter 6

Proof of Theorem 6.1 From Equation 6.2, we have $\ln O'(c \mid \mathbf{e}) = \ln O(c \mid \mathbf{e}) + \Delta \ln O(c)$, where $\Delta \ln O(c) = \ln O'(c) - \ln O(c)$. We investigate both cases of $\Delta \ln O(c)$ being negative and positive:

- If $\Delta \ln O(c) < 0$, for every $\mathbf{e}$ such that $F^{\sigma}_{\mathcal{B}}(\mathbf{e}) = 0$, we must have $F^{\sigma}_{\mathcal{B}'}(\mathbf{e}) = 0$. On the other hand, for every $\mathbf{e}$ such that $F^{\sigma}_{\mathcal{B}}(\mathbf{e}) = 1$, we still have $F^{\sigma}_{\mathcal{B}'}(\mathbf{e}) = 1$ iff $\ln O(c \mid \mathbf{e}) \geq \sigma - \Delta \ln O(c)$. Therefore, the classifiers $F^{\sigma}_{\mathcal{B}}$ and $F^{\sigma}_{\mathcal{B}'}$ are the same iff $\sigma^1 = \min_{\mathbf{e}:F^{\sigma}_{\mathcal{B}}(\mathbf{e})=1} \ln O(c \mid \mathbf{e}) \geq \sigma - \Delta \ln O(c)$.
- If $\Delta \ln O(c) > 0$, for every $\mathbf{e}$ such that $F^{\sigma}_{\mathcal{B}}(\mathbf{e}) = 1$, we must have $F^{\sigma}_{\mathcal{B}'}(\mathbf{e}) = 1$. On the other hand, for every $\mathbf{e}$ such that $F^{\sigma}_{\mathcal{B}}(\mathbf{e}) = 0$, we still have $F^{\sigma}_{\mathcal{B}'}(\mathbf{e}) = 0$ iff $\ln O(c \mid \mathbf{e}) < \sigma - \Delta \ln O(c)$. Therefore, the classifiers $F^{\sigma}_{\mathcal{B}}$ and $F^{\sigma}_{\mathcal{B}'}$ are the same iff $\sigma^0 = \max_{\mathbf{e}:F^{\sigma}_{\mathcal{B}}(\mathbf{e})=0} \ln O(c \mid \mathbf{e}) < \sigma - \Delta \ln O(c)$.

Therefore, the classifiers $F^{\sigma}_{\mathcal{B}}$ and $F^{\sigma}_{\mathcal{B}'}$ are the same iff $\Delta \ln O(c) \in [\sigma - \sigma^1, \sigma - \sigma^0)$, which is equivalent to $\ln O'(c) \in I(F^{\sigma}_{\mathcal{B}}) = [\ln O(c) + \sigma - \sigma^1, \ln O(c) + \sigma - \sigma^0)$.□

Proof of Theorem 6.2 When we change the prior log-odds of a naive Bayes network, we induce a different classifier only when $\ln O(c \mid \mathbf{e})$ for some instance $\mathbf{e}$ passes σ, thereby changing the classification of $\mathbf{e}$ from 0 to 1 or from 1 to 0. Therefore, the number of distinct classifiers (including the current classifier) that can be induced by changing the prior log-odds is at most $\|\mathbf{E}\| + 1$, and is exactly $\|\mathbf{E}\| + 1$ if there does not exist two different instances $\mathbf{e}$ and $\mathbf{e}^*$ such that $\ln O(c \mid \mathbf{e}) = \ln O(c \mid \mathbf{e}^*)$.□

Proof of Theorem 6.3 The proof is similar to that of Theorem 6.1. From Equation 6.2, we have $\ln O'(c \mid \mathbf{e}) = \ln O(c \mid \mathbf{e}) + \Delta W(e_i, c)$ if $e_i \in \mathbf{e}$, where

$\Delta W(e_i, c) = W(e_i, c) - W(e_i, c)$. If $\Delta W(e_i, c) < 0$, the classifiers $F^{\sigma}_{\mathcal{B}}$ and $F^{\sigma}_{\mathcal{B}'}$ are the same iff $\sigma^1_{e_i} = \min_{\mathbf{e}:e_i \in \mathbf{e}, F^{\sigma}_{\mathcal{B}}(\mathbf{e})=1} \ln O(c \mid \mathbf{e}) \geq \sigma - \Delta W(e_i, c)$. If $\Delta W(e_i, c) > 0$, the classifiers $F^{\sigma}_{\mathcal{B}}$ and $F^{\sigma}_{\mathcal{B}'}$ are the same iff $\sigma^0_{e_i} = \max_{\mathbf{e}:e_i \in \mathbf{e}, F^{\sigma}_{\mathcal{B}}(\mathbf{e})=0} \ln O(c \mid \mathbf{e}) < \sigma - \Delta W(e_i, c)$. Therefore, the classifiers $F^{\sigma}_{\mathcal{B}}$ and $F^{\sigma}_{\mathcal{B}'}$ are the same iff for every e_i, $\Delta W(e_i, c) \in [\sigma - \sigma^1_{e_i}, \sigma - \sigma^0_{e_i})$, which is equivalent to $W(e_i, c) \in [W(e_i, c) + \sigma - \sigma^1_{e_i}, W(e_i, c) + \sigma - \sigma^0_{e_i})$.□

Proof of Theorem 6.4 The number of distinct classifiers (including the current classifier) that can be induced by changing all weights of evidence of attribute E_i appears to be $(\|\mathbf{E} - E_i\| + 1)^{\|E_i\|}$ at first glance, since from Theorem 6.2, we know that $\|\mathbf{E} - E_i\| + 1$ distinct classifiers can be induced by changing the prior log-odds of the new network $\mathcal{B}_{e_i}$, which is obtained by removing attribute E_i from $\mathcal{B}$, and for every value e_i of E_i, the classifier $F^{\sigma}_{\mathcal{B}_{e_i}}$ can be equivalent to any of these distinct classifiers if its prior log-odds $\ln O(c \mid e_i)$ can take on any value. However, this is true only if we can also change $\ln O(c)$. This is not true if $\ln O(c)$ cannot be changed because of the restriction that among all weights of evidence $W(e_i, c)$, at least one must be positive and at least one must be negative (unless all are zero), due to the fact that when going from one probability distribution to another, at least one probability must increase and at least one must decrease (unless all probabilities are the same). To find the actual maximum number of distinct classifiers, we have to solve the following analogous problem with $a = \|\mathbf{E} - E_i\|$ and $b = \|E_i\|$: given $S = \{0, 1, \ldots, a\}$, and $k \in S$, what is the number of permutations of $(k_1, \ldots, k_b) \in S^b$, if $(\bigvee_{i=1}^{b} k_i \geq k) \wedge (\bigvee_{i=1}^{b} k_i \leq k)$? The answer is $(a+1)^b - k^b - (a-k)^b$, and its maximum value is $(a+1)^b - \lfloor a/2 \rfloor^b - \lceil a/2 \rceil^b$, attained when $k = \lfloor a/2 \rfloor$.□

Proof of Theorem 6.5 Given instantiations χ and χ^* of attributes $E_1, \ldots, E_k$, the following statements are equivalent:

1. Classifiers $F^{\sigma}_{\mathcal{B}_\chi}$ and $F^{\sigma}_{\mathcal{B}_{\chi^*}}$ are the same;
2. $\ln O(c \mid \chi^*) \in I(F^{\sigma}_{\mathcal{B}_\chi})$;
3. For every value e_{k+1} of E_{k+1}, the classifiers $F^{\sigma}_{\mathcal{B}_{\chi, e_{k+1}}}$ and $F^{\sigma}_{\mathcal{B}_{\chi^*, e_{k+1}}}$ are the same;
4. For every value e_{k+1} of E_{k+1}, $\ln O(c \mid \chi^*, e_{k+1}) \in I(F^{\sigma}_{\mathcal{B}_{\chi, e_{k+1}}})$.

Moreover, from Equation 6.1, we have $\ln O(c \mid \chi^*, e_{k+1}) = \ln O(c \mid \chi^*) + W(e_{k+1}, c)$. Therefore, the equivalence interval $I(F^{\sigma}_{\mathcal{B}_\chi})$ can be computed if we know the equivalence interval $I(F^{\sigma}_{\mathcal{B}_{\chi, e_{k+1}}})$ for every value e_{k+1} of E_{k+1}, by finding the values of $\ln O(c \mid \chi^*)$ such that $\ln O(c \mid \chi^*, e_{k+1}) \in I(F^{\sigma}_{\mathcal{B}_{\chi, e_{k+1}}})$ for every e_{k+1}, and we have:

$$I(F^{\sigma}_{\mathcal{B}_\chi}) = \bigcap_{e_{k+1}} \left\{ x : x + W(e_{k+1}, c) \in I(F^{\sigma}_{\mathcal{B}_{\chi, e_{k+1}}}) \right\} . \square$$

Proof of Theorem 6.6 Because a node in the k-th cache is reached by some path $e_1, \ldots, e_k$, the number of nodes in the k-th cache can be no more than $\|E_1, \ldots, E_k\|$. We also know that a node in the k-th cache is the root of a sub-ODD that represents a naive Bayes classifier with attributes $E_{k+1}, \ldots, E_n$. Theorem 6.2 shows that at most $\|E_{k+1}, \ldots, E_n\| + 1$ distinct classifiers can be induced by changing the prior log-odds, and this number also bounds the number of nodes in the k-th cache, since Algorithm 6.2 does not create duplicate nodes corresponding to isomorphic sub-ODDs. Therefore, the number of nodes in the k-th cache is at most $\min\{\|E_1, \ldots, E_k\|, \|E_{k+1}, \ldots, E_n\| + 1\}$. This proves that the number of nodes in the ODD is at most $\sum_{k=0}^{n} \min\{\|E_1, \ldots, E_k\|, \|E_{k+1}, \ldots, E_n\| + 1\}$, since there are $n+1$ caches, with $k = 0, \ldots, n$. We can also easily see that if all attributes have at most b values, the space complexity is $O(b^{n/2})$. Moreover, because the nodes in each cache are indexed by their equivalence intervals, we can find and store the nodes in each cache using binary search. Therefore, the time complexity of Algorithm 6.1 is $O(nb^{n/2})$.$\square$

A.5 Proofs of Theorems in Chapter 7

Proof of Theorem 7.1 Given instantiation $\mathbf{c}$ such that $\mathbf{c} \sim \mathbf{x}$, the potential of $\mathbf{x}$ is linear in the parameter $\theta_{\mathbf{c}}$, and the ratio of $\psi'(\mathbf{x})$ and $\psi(\mathbf{x})$ induced by $\mathcal{M}$ and $\mathcal{M}'$ respectively is:

$$\frac{\psi'(\mathbf{x})}{\psi(\mathbf{x})} = \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}},$$

if $\partial\psi(\mathbf{x})/\partial\theta_{\mathbf{c}} \neq 0$. We have:

$$\frac{Pr'(\mathbf{x})}{Pr(\mathbf{x})} = \frac{\zeta'\psi'(\mathbf{x})}{\zeta\psi(\mathbf{x})} = \frac{\zeta'\theta'_{\mathbf{c}}}{\zeta\theta_{\mathbf{c}}}.$$

Note that because the clique parameters have changed, the normalizing constants ζ and ζ' for networks $\mathcal{M}$ and $\mathcal{M}'$ respectively are different. Therefore, the distance measure between Pr and Pr' is given by:

$$\begin{aligned}
D(Pr, Pr') &= \ln \max_{\mathbf{x}} \frac{Pr'(\mathbf{x})}{Pr(\mathbf{x})} - \ln \min_{\mathbf{x}} \frac{Pr'(\mathbf{x})}{Pr(\mathbf{x})} \\
&= \ln \max_{\mathbf{c}} \frac{\zeta'\theta'_{\mathbf{c}}}{\zeta\theta_{\mathbf{c}}} - \ln \min_{\mathbf{c}} \frac{\zeta'\theta'_{\mathbf{c}}}{\zeta\theta_{\mathbf{c}}} \\
&= \ln \max_{\mathbf{c}} \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}} - \ln \min_{\mathbf{c}} \frac{\theta'_{\mathbf{c}}}{\theta_{\mathbf{c}}} \\
&= D(\Theta_{\mathbf{C}}, \Theta'_{\mathbf{C}}),
\end{aligned}$$

if $\partial\psi(\mathbf{x})/\partial\theta_{\mathbf{c}} \neq 0$ for all $\mathbf{c} \sim \mathbf{x}$.$\square$

A.6 Proofs of Theorems in Chapter 8

Proof of Theorem 8.1 Given the distribution Pr' in Equation 8.3, because events $\gamma_1, \ldots, \gamma_n$ are mutually exclusive, we have:

$$\begin{aligned} Pr'(\gamma_i) &= \rho_i \frac{Pr(\gamma_i)}{Pr(\gamma_i)} \\ &= \rho_i \text{ for } i = 1, \ldots, n. \end{aligned}$$

satisfying Equation 8.2, and:

$$Pr'(\alpha, \gamma_i) = \rho_i \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)} \text{ for } i = 1, \ldots, n.$$

Therefore, Pr' is obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$, i.e., it satisfies Equation 8.1, since:

$$\begin{aligned} Pr'(\alpha \mid \gamma_i) &= \frac{Pr'(\alpha, \gamma_i)}{Pr'(\gamma_i)} \\ &= \frac{\rho_i \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)}}{\rho_i} \\ &= \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)} \\ &= Pr(\alpha \mid \gamma_i) \text{ for } i = 1, \ldots, n. \end{aligned}$$

On the other hand, if there is a distribution Pr' that satisfies both Equations 8.1 and 8.2, the probability of event α under Pr' must be:

$$\begin{aligned} Pr'(\alpha) &= \sum_{i=1}^{n} Pr'(\alpha \mid \gamma_i) Pr'(\gamma_i) \\ &= \sum_{i=1}^{n} Pr(\alpha \mid \gamma_i) \rho_i \\ &= \sum_{i=1}^{n} \rho_i \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)} . \square \end{aligned}$$

Proof of Theorem 8.2 We want to prove that the distribution given by Equation 8.6 is the unique distribution $Pr(\cdot \mid \eta)$ identified by the virtual evidence method in Definition 8.3. First of all, if there is a distribution $Pr(\cdot \mid \eta)$ that satisfies both Equations 8.4 and 8.5, we have $Pr(\eta \mid \gamma_i) = k\lambda_i$ for $i = 1, \ldots, n$, where k is a constant, and the probability of event α under distribution $Pr(\cdot \mid \eta)$ must be:

$$\begin{aligned}
Pr(\alpha \mid \eta) &= \frac{Pr(\alpha, \eta)}{Pr(\eta)} \\
&= \frac{\sum_{i=1}^{n} Pr(\alpha, \eta, \gamma_i)}{Pr(\eta)} \\
&= \frac{\sum_{i=1}^{n} Pr(\eta \mid \gamma_i, \alpha) Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} Pr(\eta \mid \gamma_j) Pr(\gamma_j)} \\
&= \frac{\sum_{i=1}^{n} Pr(\eta \mid \gamma_i) Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} Pr(\eta \mid \gamma_j) Pr(\gamma_j)} \\
&= \frac{\sum_{i=1}^{n} k\lambda_i Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} k\lambda_j Pr(\gamma_j)} \\
&= \frac{\sum_{i=1}^{n} \lambda_i Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)}.
\end{aligned}$$

On the other hand, given a distribution that satisfies Equation 8.6, because events $\gamma_1, \ldots, \gamma_n$ are mutually exclusive, we have:

$$\begin{aligned}
Pr(\eta \mid \gamma_1) : \ldots : Pr(\eta \mid \gamma_n) &= \frac{Pr(\eta, \gamma_1)}{Pr(\gamma_1)} : \ldots : \frac{Pr(\eta, \gamma_n)}{Pr(\gamma_n)} \\
&= \frac{Pr(\gamma_1 \mid \eta) Pr(\eta)}{Pr(\gamma_1)} : \ldots : \frac{Pr(\gamma_n \mid \eta) Pr(\eta)}{Pr(\gamma_n)} \\
&= \frac{\frac{\lambda_1 Pr(\gamma_1)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)} Pr(\eta)}{Pr(\gamma_1)} : \ldots : \frac{\frac{\lambda_n Pr(\gamma_n)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)} Pr(\eta)}{Pr(\gamma_n)} \\
&= \lambda_1 : \ldots : \lambda_n,
\end{aligned}$$

satisfying Equation 8.4. Therefore, $\lambda_i = kPr(\eta \mid \gamma_i)$ for $i = 1, \ldots, n$, where k is a constant, and Equation 8.5 is also satisfied because:

$$\begin{aligned}
Pr(\eta \mid \gamma_i, \alpha) &= \frac{Pr(\eta, \gamma_i, \alpha)}{Pr(\gamma_i, \alpha)} \\
&= \frac{Pr(\gamma_i, \alpha \mid \eta) Pr(\eta)}{Pr(\gamma_i, \alpha)}
\end{aligned}$$

$$\begin{aligned}
&= \frac{\frac{\lambda_i Pr(\gamma_i,\alpha)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)} Pr(\eta)}{Pr(\gamma_i,\alpha)} \\
&= \frac{\lambda_i Pr(\eta)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)} \\
&= \frac{k Pr(\eta \mid \gamma_i) Pr(\eta)}{\sum_{j=1}^n k Pr(\eta \mid \gamma_j) Pr(\gamma_j)} \\
&= \frac{Pr(\eta \mid \gamma_i) Pr(\eta)}{\sum_{j=1}^n Pr(\eta \mid \gamma_j) Pr(\gamma_j)} \\
&= \frac{Pr(\eta \mid \gamma_i) Pr(\eta)}{Pr(\eta)} \\
&= Pr(\eta \mid \gamma_i) \text{ for } i = 1, \ldots, n. \square
\end{aligned}$$

Proof of Theorem 8.3 Given the distribution $Pr(\cdot \mid \eta)$ in Equation 8.6, because events $\gamma_1, \ldots, \gamma_n$ are mutually exclusive, we have:

$$Pr(\gamma_i \mid \eta) = \frac{\lambda_i Pr(\gamma_i)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)} \text{ for } i = 1, \ldots, n.$$

and:

$$Pr(\alpha, \gamma_i \mid \eta) = \frac{\lambda_i Pr(\alpha, \gamma_i)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)} \text{ for } i = 1, \ldots, n.$$

Therefore, $Pr(\cdot \mid \eta)$ is obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$, i.e., it satisfies Equation 8.1, since:

$$\begin{aligned}
Pr(\alpha \mid \gamma_i, \eta) &= \frac{Pr(\alpha, \gamma_i \mid \eta)}{Pr(\gamma_i \mid \eta)} \\
&= \frac{\frac{\lambda_i Pr(\alpha,\gamma_i)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)}}{\frac{\lambda_i Pr(\gamma_i)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)}} \\
&= \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)} \\
&= Pr(\alpha \mid \gamma_i) \text{ for } i = 1, \ldots, n. \square
\end{aligned}$$

Proof of Theorem 8.4 From Equations 8.6 and 8.7, we have:

$$Pr'(\gamma_i) = \rho_i = \frac{\lambda_i Pr(\gamma_i)}{\sum_{j=1}^n \lambda_j Pr(\gamma_j)} \text{ for } i = 1, \ldots, n.$$

We can substitute the set of posterior probabilities into the distribution obtained by Jeffrey's rule, given by Equation 8.3, and get:

$$\begin{aligned} Pr'(\alpha) &= \sum_{i=1}^{n} \rho_i \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)} \\ &= \sum_{i=1}^{n} \frac{\lambda_i Pr(\gamma_i)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)} \cdot \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)} \\ &= \frac{\sum_{i=1}^{n} \lambda_i Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)}. \end{aligned}$$

This is exactly the distribution obtained by the virtual evidence method, given by Equation 8.6, with likelihood ratios $\lambda_1, \ldots, \lambda_n$.$\square$

Proof of Theorem 8.5 From Equation 8.8, we have:

$$\lambda_i = \frac{k\rho_i}{Pr(\gamma_i)} \text{ for } i = 1, \ldots, n,$$

where k is a constant. We can substitute the likelihood ratios $\lambda_1, \ldots, \lambda_n$ into the distribution obtained by Pearl's method, given by Equation 8.6, and get:

$$\begin{aligned} Pr(\alpha \mid \eta) &= \frac{\sum_{i=1}^{n} \lambda_i Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} \lambda_j Pr(\gamma_j)}. \\ &= \frac{\sum_{i=1}^{n} \frac{k\rho_i}{Pr(\gamma_i)} Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} \frac{k\rho_j}{Pr(\gamma_j)} Pr(\gamma_j)}. \\ &= \frac{\sum_{i=1}^{n} \frac{\rho_i}{Pr(\gamma_i)} Pr(\alpha, \gamma_i)}{\sum_{j=1}^{n} \rho_j} \\ &= \sum_{i=1}^{n} \rho_i \frac{Pr(\alpha, \gamma_i)}{Pr(\gamma_i)}, \end{aligned}$$

since $\sum_{j=1}^{n} \rho_j = \sum_{j=1}^{n} Pr'(\gamma_j) = 1$. This is exactly the distribution obtained by Jeffrey's rule, given by Equation 8.3, with the set of posterior probabilities $Pr'(\gamma_i) = \rho_i$ for $i = 1, \ldots, n$.$\square$

Proof of Theorem 8.6 From Equation 8.6, we have:

$$\frac{Pr(\gamma_i \mid \eta)}{Pr(\gamma_i)} = \frac{\frac{\lambda_i Pr(\gamma_i)}{\sum_{k=1}^{n} \lambda_k Pr(\gamma_k)}}{Pr(\gamma_i)}$$

$$= \frac{\lambda_i}{\sum_{k=1}^n \lambda_k Pr(\gamma_k)} \text{ for } i = 1, \ldots, n.$$

Therefore, we have:

$$\begin{aligned} B_{Pr(\cdot|\eta),Pr}(\gamma_i : \gamma_j) &= \frac{\frac{Pr(\gamma_i|\eta)}{Pr(\gamma_j|\eta)}}{\frac{Pr(\gamma_i)}{Pr(\gamma_j)}} \\ &= \frac{\frac{Pr(\gamma_i|\eta)}{Pr(\gamma_i)}}{\frac{Pr(\gamma_j|\eta)}{Pr(\gamma_j)}} \\ &= \frac{\frac{\lambda_i}{\sum_{k=1}^n \lambda_k Pr(\gamma_k)}}{\frac{\lambda_j}{\sum_{k=1}^n \lambda_k Pr(\gamma_k)}} \\ &= \frac{\lambda_i}{\lambda_j} \text{ for } i, j = 1, \ldots, n. \square \end{aligned}$$

Proof of Theorem 8.7 If the two sets of probabilities $Pr(\gamma_1), \ldots, Pr(\gamma_n)$ and $Pr'(\gamma_1), \ldots, Pr'(\gamma_n)$ do not have the same support, there must exist some world ω where $Pr(\omega) = 0$ and $Pr'(\omega) \neq 0$ or $Pr(\omega) \neq 0$ and $Pr'(\omega) = 0$, and thus the distributions Pr and Pr' also do not have the same support, giving us $D(Pr, Pr') = \infty = \ln \max_{i=1}^n (Pr'(\gamma_i)/Pr(\gamma_i)) - \ln \min_{i=1}^n (Pr'(\gamma_i)/Pr(\gamma_i))$. Otherwise, given a world ω where $\omega \models \gamma_i$, from Equation 8.1, we have:

$$\begin{aligned} \frac{Pr'(\omega)}{Pr(\omega)} &= \frac{Pr'(\omega \mid \gamma_i)Pr'(\gamma_i)}{Pr(\omega \mid \gamma_i)Pr(\gamma_i)} \\ &= \frac{Pr'(\gamma_i)}{Pr(\gamma_i)}. \end{aligned}$$

Therefore, the distance measure between Pr and Pr' can be computed by:

$$\begin{aligned} D(Pr, Pr') &= \ln \max_\omega \frac{Pr'(\omega)}{Pr(\omega)} - \ln \min_\omega \frac{Pr'(\omega)}{Pr(\omega)} \\ &= \ln \max_{i=1}^n \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} - \ln \min_{i=1}^n \frac{Pr'(\gamma_i)}{Pr(\gamma_i)}. \square \end{aligned}$$

Proof of Theorem 8.8 Let Pr'' be any distribution such that $Pr''(\gamma_i) = Pr'(\gamma_i)$ for $i = 1, \ldots, n$. We would like to prove that $D(Pr, Pr'') \geq D(Pr, Pr')$, where Pr' is obtained from Pr by probability kinematics on $\gamma_1, \ldots, \gamma_n$. If distributions Pr and Pr'' do not have the same support, we have $D(Pr, Pr'') = \infty \geq D(Pr, Pr')$. Otherwise, we can write the following inequalities:

$$\begin{aligned}
Pr(\gamma_i) \max_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} &= \sum_{\omega \models \gamma_i} Pr(\omega) \max_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} \\
&\geq \sum_{\omega \models \gamma_i} Pr(\omega) \frac{Pr''(\omega)}{Pr(\omega)} \text{ for } i = 1, \ldots, n; \\
Pr(\gamma_i) \min_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} &= \sum_{\omega \models \gamma_i} Pr(\omega) \min_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} \\
&\leq \sum_{\omega \models \gamma_i} Pr(\omega) \frac{Pr''(\omega)}{Pr(\omega)} \text{ for } i = 1, \ldots, n.
\end{aligned}$$

Since $\sum_{\omega \models \gamma_i} Pr(\omega)(Pr''(\omega)/Pr(\omega)) = \sum_{\omega \models \gamma_i} Pr''(\omega) = Pr''(\gamma_i) = Pr'(\gamma_i)$ for $i = 1, \ldots, n$, this gives us:

$$\min_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} \leq \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} \leq \max_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} \text{ for } i = 1, \ldots, n.$$

Therefore, the distance measure between Pr and Pr'' is bounded by:

$$\begin{aligned}
D(Pr, Pr'') &= \ln \max_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} - \ln \min_{\omega} \frac{Pr''(\omega)}{Pr(\omega)} \\
&\geq \ln \max_{i=1}^{n} \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} - \ln \min_{i=1}^{n} \frac{Pr'(\gamma_i)}{Pr(\gamma_i)} \\
&= D(Pr, Pr').
\end{aligned}$$

Therefore, the distribution Pr' gives us the smallest distance among all possible distributions that agree with Pr' on the probabilities of events $\gamma_1, \ldots, \gamma_n$.□

References

[ACC] David Allen, Keith Cascio, Hei Chan, Mark Chavira, James D. Park, and Adnan Darwiche. "SamIam: Sensitivity Analysis, Modeling, Inference and More." URL: `http://reasoning.cs.ucla.edu/samiam/`.

[Bry86] Randal E. Bryant. "Graph-Based Algorithms for Boolean Function Manipulation." *IEEE Transactions on Computers*, **35**:677–691, 1986.

[CD01] Hei Chan and Adnan Darwiche. "When Do Numbers Really Matter?" In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 65–74. Morgan Kaufmann Publishers, 2001.

[CD02a] Hei Chan and Adnan Darwiche. "A Distance Measure for Bounding Probabilistic Belief Change." In *Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI)*, pp. 539–545. AAAI Press, 2002.

[CD02b] Hei Chan and Adnan Darwiche. "When Do Numbers Really Matter?" *Journal of Artificial Intelligence Research*, **17**:265–287, 2002.

[CD03a] Hei Chan and Adnan Darwiche. "On the Revision of Probabilistic Beliefs Using Uncertain Evidence." In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 99–105. Morgan Kaufmann Publishers, 2003.

[CD03b] Hei Chan and Adnan Darwiche. "Reasoning About Bayesian Network Classifiers." In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 107–115. Morgan Kaufmann Publishers, 2003.

[CD04] Hei Chan and Adnan Darwiche. "Sensitivity Analysis in Bayesian Networks: From Single to Multiple Parameters." In *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 67–75. AUAI Press, 2004.

[CD05a] Hei Chan and Adnan Darwiche. "A Distance Measure for Bounding Probabilistic Belief Change." *International Journal of Approximate Reasoning*, **38**:149–174, 2005.

[CD05b] Hei Chan and Adnan Darwiche. "On the Revision of Probabilistic Beliefs Using Uncertain Evidence." *Artificial Intelligence*, **163**:67–90, 2005.

[CD05c] Hei Chan and Adnan Darwiche. "Sensitivity Analysis in Markov Networks." In *Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1300–1305. Professional Book Center, 2005.

[CGH97] Enrique Castillo, José Manuel Gutiérrez, and Ali S. Hadi. "Sensitivity Analysis in Discrete Bayesian Networks." *IEEE Transactions on Systems, Man, and Cybernetics, Part A (Systems and Humans)*, **27**:412–423, 1997.

[CJK00] Veerle M. H. Coupé, Finn V. Jensen, Uffe Kjærulff, and Linda C. van der Gaag. "A Computational Architecture for N-way Sensitivity Analysis of Bayesian Networks." Technical report, 2000.

[CPO99] Veerle M. H. Coupé, Niels Peek, Jaap Ottenkamp, and J. Dik F. Habbema. "Using Sensitivity Analysis for Efficient Quantification of a Belief Network." *Artificial Intelligence in Medicine*, **17**:223–247, 1999.

[Dar03] Adnan Darwiche. "A Differential Approach to Inference in Bayesian Networks." *Journal of the ACM*, **50**:280–305, 2003.

[DH73] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* John Wiley & Sons, New York, 1973.

[DM02] Adnan Darwiche and Pierre Marquis. "A Knowledge Compilation Map." *Journal of Artificial Intelligence Research*, **17**:229–264, 2002.

[DP97] Adnan Darwiche and Judea Pearl. "On the Logic of Iterated Belief Revision." *Artificial Intelligence*, **87**:1–29, 1997.

[DZ82] Persi Diaconis and Sandy L. Zabell. "Updating Subjective Probability." *Journal of the American Statistical Association*, **77**:822–830, 1982.

[FGG97] Nir Friedman, Dan Geiger, and Moises Goldszmidt. "Bayesian Network Classifiers." *Machine Learning*, **29**:131–163, 1997.

[G88] Peter Gärdenfors. *Knowledge in Flux: Modeling the Dynamics of Epistemic States.* MIT Press, Cambridge, 1988.

[GG84] Stuart Geman and Donald Geman. "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images." *IEEE Transactions in Pattern Analysis and Machine Intelligence*, **6**:721–741, 1984.

[GGS97] Russell Greiner, Adam J. Grove, and Dale Schuurmans. “Learning Bayesian Nets That Perform Well.” In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 198–207. Morgan Kaufmann Publishers, 1997.

[Goo50] Irving J. Good. *Probability and the Weighing of Evidence.* Charles Griffin, London, 1950.

[Goo83] Irving J. Good. *Good Thinking: The Foundations of Probability and Its Applications.* University of Minnesota Press, Minneapolis, 1983.

[GR01] Ashutosh Garg and Dan Roth. “Understanding Probabilistic Classifiers.” In *Proceedings of the Twelfth European Conference on Machine Learning (ECML)*, pp. 179–191. Springer-Verlag, 2001.

[HBM] S. Hettich, C. L. Blake, and C. J. Merz. “UCI Repository of Machine Learning Databases.” URL: `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[HP03] Joseph Y. Halpern and Riccardo Pucella. “A Logic for Reasoning About Evidence.” In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 297–304. Morgan Kaufmann Publishers, 2003.

[Jef65] Richard C. Jeffrey. *The Logic of Decision.* McGraw-Hill, New York, 1965. 2nd edition, University of Chicago Press, Chicago, 1983; paperback correction, 1990.

[Jef92] Richard C. Jeffrey. *Probability and the Art of Judgement.* Cambridge University Press, Cambridge, 1992.

[Jen99] Finn V. Jensen. “Gradient Descent Training of Bayesian Networks.” In *Proceedings of the Fifth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU)*, pp. 190–200. Springer-Verlag, 1999.

[Jen01] Finn V. Jensen. *Bayesian Networks and Decision Graphs.* Springer-Verlag, New York, 2001.

[JLO90] Finn V. Jensen, Steffen L. Lauritzen, and Kristian G. Olesen. “Bayesian Updating in Causal Probabilistic Networks by Local Computations.” *Computational Statistics Quarterly*, **5**:269–282, 1990.

[Kjae94] Uffe Kjærulff. "Reduction of Computational Complexity in Bayesian Networks Through Removal of Weak Dependences." In *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 374–382. Morgan Kaufmann Publishers, 1994.

[KL51] S. Kullback and R. A. Leibler. "On Information and Sufficiency." *Annals of Mathematical Statistics*, **22**:79–86, 1951.

[Kv00] Uffe Kjærulff and Linda C. van der Gaag. "Making Sensitivity Analysis Computationally Efficient." In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 317–325. Morgan Kaufmann Publishers, 2000.

[Las95] Kathryn B. Laskey. "Sensitivity Analysis for Probability Assessments in Bayesian Networks." *IEEE Transactions on Systems, Man, and Cybernetics*, **25**:901–909, 1995.

[LIT92] Pat Langley, Wayne Iba, and Kevin Thompson. "An Analysis of Bayesian Classifiers." In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI)*, pp. 223–228. AAAI Press, 1992.

[Pea88] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers, San Francisco, 1988.

[Pea01] Judea Pearl. "On Two Pseudo-Paradoxes in Bayesian Analysis." *Annals of Mathematics and Artificial Intelligence*, **32**:171–177, 2001.

[PHP96] Malcolm Pradhan, Max Henrion, Gregory Provan, Brendan Del Favero, and Kurt Huang. "The Sensitivity of Belief Networks to Imprecise Probabilities: An Experimental Investigation." *Artificial Intelligence*, **85**:363–397, 1996.

[Poo98] David Poole. "Context-Specific Approximation in Probabilistic Inference." In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 447–454. Morgan Kaufmann Publishers, 1998.

[RBK95] Stuart Russell, John Binder, Daphne Koller, and Keiji Kanazawa. "Local Learning in Probabilistic Networks with Hidden Variables." In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1146–1152. Morgan Kaufmann Publishers, 1995.

[Som] Fabio Somenzi. "CU Decision Diagram Package." URL: `http://vlsi.colorado.edu/~fabio/CUDD/cuddIntro.html`.

[SS86] Prakash P. Shenoy and Glenn Shafer. "Propagating Belief Functions with Local Computations." *IEEE Expert*, **1**:43–52, 1986.

[van97] Robert A. van Engelen. "Approximating Bayesian Belief Networks by Arc Removal." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**:916–920, 1997.

[vR01] Linda C. van der Gaag and Silja Renooij. "Analysing Sensitivity Data from Probabilistic Networks." In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 530–537. Morgan Kaufmann Publishers, 2001.

[Wag02] Carl Wagner. "Probability Kinematics and Commutativity." *Philosophy of Science*, **69**:266–278, 2002.

[WD00] Haiqin Wang and Marek J. Druzdzel. "User Interface Tools for Navigation in Conditional Probability Tables and Elicitation of Probabilities in Bayesian Networks." In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 617–625. Morgan Kaufmann Publishers, 2000.