Validation of Bayesian Networks

with a case study on fingerprint general patterns

Netherlands Forensic Institute
Supervisors: Dr. **Didier Meuwly**
Mr. **Rudolf Haraksim**

Utrecht University
Supervisor: Dr. **Silja Renooij**

Vikram Doshi
Student No: 3741281

Dep. of Information and Computing Science
Utrecht University
Netherlands

Index

Part I: Case Study (Forensic Science)
Validation of Bayesian networks for fingerprint general pattern

Part II: Computer Science
MAP Computation in Bayesian Networks

to my parents: ma & papa

and grandparents: ba-dada, nana-nani & bhabhoo

‘we live in a beautiful and orderly world...
and not in a formless chaos, as it sometimes seems’
- M. C. Escher

Chapter 1:

General Introduction

1. 1. Outline

A Bayesian network is a graphical model that encodes probabilistic relationships among variables of interest. When used in conjunction with statistical techniques, the graphical model has several advantages for data analysis. One, because the model encodes dependencies among all variables, it readily handles situations where some data entries are missing. Two, a Bayesian network can be used to learn causal relationships, and hence can be used to gain understanding about a problem domain and to predict the consequences of intervention. Three, the model has both a causal and probabilistic semantics; it is an ideal representation for combining prior knowledge (which often comes in causal form) and data. And finally, Bayesian statistical methods in conjunction with Bayesian networks offer an efficient and principled approach for avoiding the overfitting of data [Heckerman, 1996].

Validation of a Bayesian network is done to establish its practical value. To do so, a Bayesian network is typically subjected to an evaluation study using data from the domain of application. Such a study amounts to entering the data available for each problem case into the network and computing the most likely outcome. There exist different measures which help us in establishing a network's quality by comparing its outcome against a given standard of validity.

In Part-I of the thesis, we design and validate Bayesian networks for evaluation of the fingerprint general pattern. These networks have two applications in forensic science. Firstly, it will help the fingerprint examiner reduce the number of reference fingerprints he has to search (to find the donor of the fingermark), by ascertaining from which finger(s) is it more likely for the general pattern(s) of the fingermark(s) to occur. Secondly, it will help the fingerprint examiner to quantify the strength of evidence of the general pattern, in terms of likelihood ratios**.**

To perform validation, we sometimes need to find the best explanation for a set of evidence. In Bayesian network, finding the best explanation amounts to finding a value assignment to some of the variables in the network that has highest posterior probability given the available evidence (i.e. the best explanation is a most likely one).

This problem is known as most probable explanation or maximum a posterior assignment which is NP-hard in general. In Part-II of this thesis, we propose two heuristics and based on experiments on some synthetic data, we show that they converge approximately to the true MAP assignment.

Part I:

Case Study (Forensic Science)

Validation of Bayesian Networks for fingerprint general pattern

Chapter 2:
Introduction and background of study

2. 1. Introduction

Forensic examiners value the contribution of each piece of evidence for forensic evaluation, since it may strengthen the support for the prosecution or the defence hypothesis.

When a fingerprint examiner compares a fingermark recovered from a crime-scene to a reference fingerprint of a suspected person, he exploits all the available information to assess the strength of evidence: for example the properties of the ridge flow (level 1 details), the spatial configuration of minutiae or major ridge path deviations (level 2 details) and of the ridges themselves (level 3 details) (Figure 2.1.1). The fingerprint examiner processes all this information and uses his experienced knowledge to assign the strength of the evidence. While tools have been developed to assist the fingerprint examiner in quantifying the strength of evidence for the spatial configurations of minutiae (level 2 detail) [Neumann et al., 2011], currently no tools exist which are able to assist in the quantification of the evidential value of the general pattern of the ridge flow (level 1 detail).

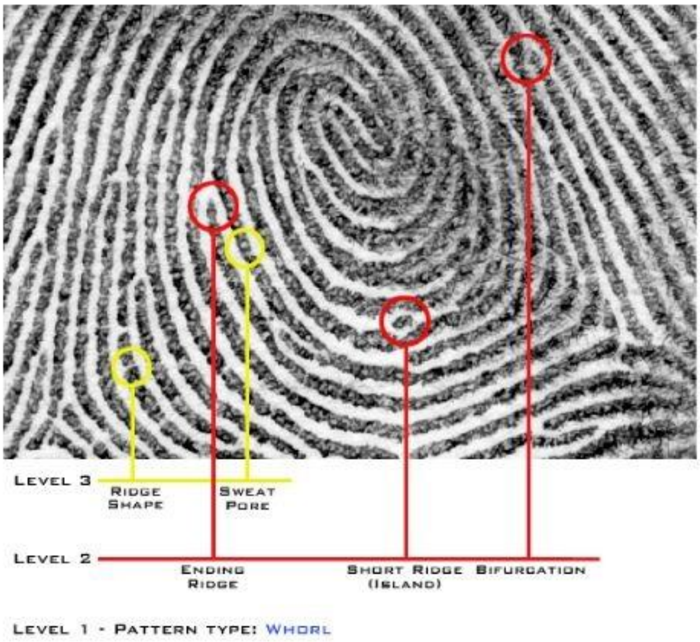


Figure 2.1.1: Example of fingerprint with the 3 levels of fingerprint detail.

In the investigation phase, a fingerprint examiner searches for a fingerprint in the reference dataset which can be linked (identification) to the fingermark found on the crime scene. Before the introduction of the automatic fingerprint identification systems (AFIS), general pattern classifications were only used to narrow down the search space by using fingerprints which had the same general pattern as found on the fingermark.

This is also done with the AFIS. However in the AFIS, to further reduce the number of fingerprints to search, fingerprint examiners also use the general pattern information to infer from which finger it is more likely for that general pattern classification to occur. This can only be performed by Fingerprint examiners, who use their knowledge acquired through experience gained in case work.

Forensic evaluation is a process of describing the evidential value resulting from the study, using a comparative approach, of the similarity and the distinctiveness of a fingermark recovered on a crime scene and a fingerprint of a candidate selected during the forensic investigation process [Stoney, 1991]. In this phase, the strength of evidence of a correspondence between the general pattern of the fingermark and the general pattern of a reference fingerprint is considered implicitly but not quantified since there is no tool to assign the strength of evidence to this.

This chapter focuses on developing such a tool by using Bayesian networks. On one side the networks will assist the fingerprint examiner in the investigation phase to know the distinctiveness of the general pattern (from which finger(s) is it more likely for the general pattern(s) of the fingermark(s) to occur). On the other side, they will help the fingerprint examiner in quantifying the evidential value of the general pattern. Similar to DNA analysis [Meuwly & Veldhuis, 2012], the strength of evidence is expressed as a likelihood ratio which compares the probability of the observation, given two alternative hypotheses. The strength of evidence of the general pattern can in a later stage of forensic evaluation be combined with the evidential values obtained from the 2nd level detail of fingerprints as well as forensic evidence related to other biometric modalities.

In the next section we briefly discuss the terminology used in this thesis. Followed by a description of the two police datasets used. Finally we conclude this chapter by briefly discussing the Bayesian networks created in a previous project.

2. 2. Terminology

The Henry classification method [Henry, 1900], developed by Sir Edward Richard Henry in 1894, allows for categorization of fingerprint records into primary groups, based on fingerprint general pattern types. In the following paragraphs we introduce the labelling convention for finger number and general pattern in accordance with Henry's classification.

Finger Number: The fingers are numbered from 1 to 10, starting from the right thumb (labelled as finger 1) and ending at the left little finger (labelled as finger 10) as shown in the figure below.

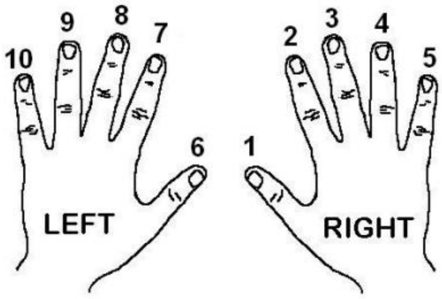
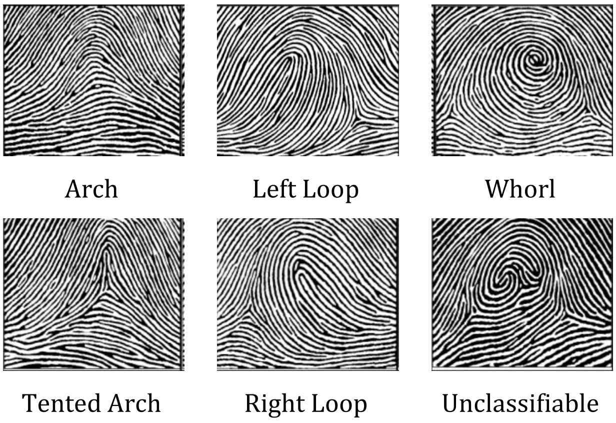


Figure 2.2.1: Fingers numbered according to Henry Classification System

General Pattern: Numerous systems exist to assign a general pattern classification to the shape of the ridge flow of a finger. In this thesis we follow a modified version of the classification codes of the [ANSI/NIST-ITL 1-2011] format, which are named arch, tented arch, left loop, right loop, whorl and unclassifiable (illustrated in Figure 2.2.2).



Arch

Left Loop

Whorl

Tented Arch

Right Loop

Unclassifiable

Figure 2.2.2: General pattern classification

[ANSI/NIST-ITL 1-2000] uses the nomenclature of radial and ulnar loops which relate to the two bones (radius and ulna) in the forearm. This classification makes the general pattern dependent on which hand they come from, which makes it impossible to determine if it is an ulnar or radial loop just from the reference print. For this reason the denomination right loop and left loop is preferred (Table 2.2.1). The direction of the loop (right or left) can be easily determined from the print, thus making the general pattern classification not dependent on which hand it comes from [F.B.I. United States, 1985].

When a print is taken from the finger, the hand is flipped from the front to the back and so a loop is inversed. A radial loop on the left hand is a right loop (Figure 2.2.3) and an ulnar loop on the left hand is a left loop (Figure 2.2.4). Similar observations hold for the right hand i.e. a radial loop on the right hand is a left loop and an ulnar loop on the right hand is a right loop.

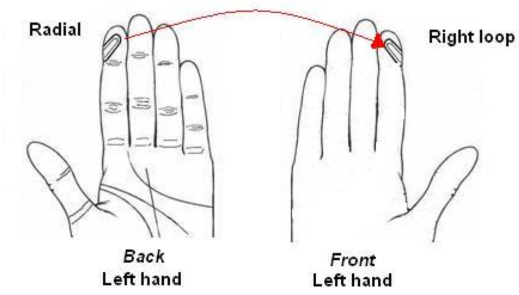


Figure 2.2.3:
Radial loop on left hand is a right loop

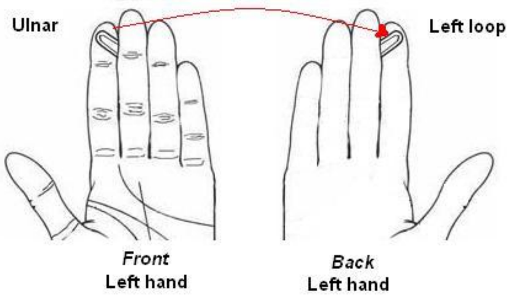


Figure 2.2.4:
Ulnar loop on left hand is a left loop.

	Ulnar Loop	**Radial Loop**
Right hand	Right loop	Left loop
Left hand	Left loop	Right loop

Table 2.2.1: Conversion of Ulnar Loop and Radial loop to Left Loop and Right loop

Sometimes fingerprint examiners face difficulty in assigning a fingerprint general pattern class. This is something caused by ambiguities such as the presence of a scar or fingerprints of people suffering with dysplasia. Another reason could be that the general pattern is too complex to classify, meaning that the general pattern has characteristics of several classes. To reduce the risk of misclassifications, a pattern is classified as ‘unclassifiable’ when it can not be assigned to only one of the other classes.

Fingermark vs. Fingerprint: In the literature, confusion exists between the terms fingerprint and fingermark. This thesis uses the following terminology: the standard rolled inked impressions captured from the finger papillary ridges are named fingerprints whereas recovered traces left by unprotected fingers in uncontrolled conditions are named fingermarks.

2. 3. Datasets

The 'Politie Landelijke Dactyloscopie Eenheid' is the entity that operates the Dutch national police AFIS system and that is the custodian of the criminal fingermark and fingerprint datasets in the Netherlands. The fingermark dataset contains information about the identified fingermarks on the crime scene. Whereas the fingerprint dataset contains the reference fingerprints of criminals collected in the Netherlands. Since a selected subset of the fingerprint dataset is used, it is possible that the same subjects are not present in both the datasets i.e. subjects whose marks are found in the fingermark dataset may not be present in the fingerprint dataset and vice-versa.

2. 3. 1 Fingermark Dataset

The fingermark data used in this research reflects the operational activity on crime scenes as processed by the national police force in the field of fingermark examination in the year of 2010 and 2011. It contains a total of 11,555 identified fingermarks from the years 2010 (4,032 identifications) and 2011 (7,523 identifications). These identified fingermarks belong to 6,523 subjects. It is important to note that in studies of crime-scene fingermarks no formal ground truth exists for an identification (or match) established between a fingermark and a reference fingerprint. Fingerprint examiners in the Netherlands use the numerical standard of 12 points to make a decision of identification [Evett & Williams, 1996; Champod, 2009]. In their decision, they try to minimize the false acceptance ratio[1]. We as researchers consider, that such datasets of pairs of identified fingermark and fingerprint images constitute sets of data with an acceptable ground truth by proxy. Moreover these fingermark latents were found on (1) different crime scenes, or (2) different objects on the same crime scene or (3) different positions on the same object. For this reason we assume that these cases were generated independently.

[1] The false acceptance ratio (FAR) is a unit used to measure the average number of false acceptances within a biometric security system. It measures and evaluates the efficiency and accuracy of a biometric system by determining the rate at which unauthorized or illegitimate users are verified on a particular system.

For each identified fingermark in this dataset, the finger number, general pattern, gender and nationality of the donor is derived from the corresponding reference fingerprint from the police fingerprint dataset (section 2.3.2). The relative frequency of occurrence of fingers over the dataset is summarized in table 2.3.1.1. Considering all the 11,555 identified fingermarks together, the proportions of the general pattern for arch is 5.64%, 22.79% for the right loop, 27.8% for the left loop, 40.32% for the whorl and 3.41% for unclassifiable. No tented arches were observed in this dataset. According to fingerprint examiners this could be because tented arch general pattern (around 5% of all fingers) is uncommon and due to the fact that the data represents the operational activity for only 2 years. Regarding gender, 92.8% of the data originates from male criminals and 2.2% from female criminals. In 5% of the cases the gender was labelled as unknown. A more detailed study of this dataset is presented in [Doekhie, 2012].

	Finger Number									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Police Identified Fingermarks	15.59	16.97	10.64	6.9	2.1	15.26	9.62	11.67	8.07	3.18

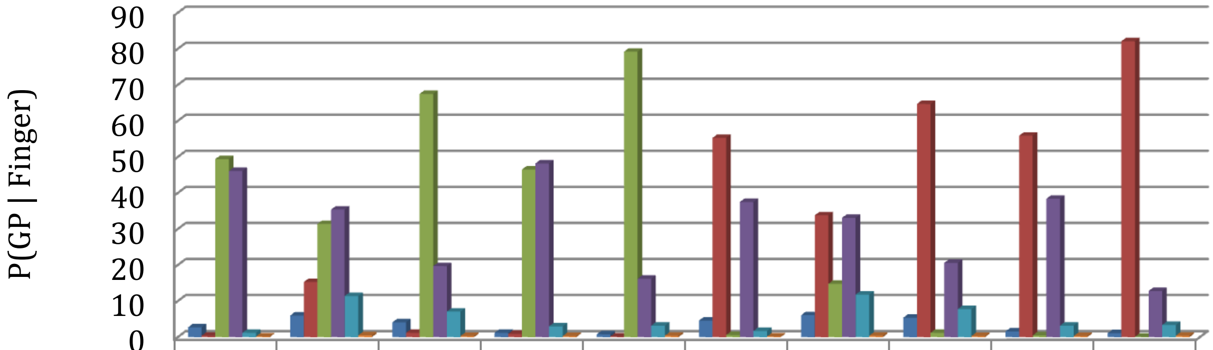
Table 2.3.1.1: Proportions (in percentage) of identified fingermarks in Police identified fingermarks

We will see later that we want to extract the information of the frequency of which finger(s) left the fingermark from this dataset. Though we have more fingermarks than subjects, there is no information in the data which can help us to determine the frequency of occurrence of two (or more) consecutive fingermarks. By consecutivity we mean that the fingermarks where left on the crime scene in a single act of touch (simultaneously imposed by the perpetrator) and that fingers are adjacent to each other and that they come from the same hand (right or left) of the suspect.

2. 3. 2. Fingerprint Dataset

The dataset consists of inked, digitized, automatically encoded and manually checked 10-print cards of the police fingerprint dataset. The general patterns of these prints have been manually assigned by fingerprint examiners. For each print, additional information regarding the finger number, gender and second level details (minutiae) of the donor is available. In this research, 10-print cards from 306,105 criminals have been selected from the original dataset to study the distribution of the general pattern over the 10 fingers (Data selection and refinement is described in Appendix A).

Considering all fingers together, the proportions of the general pattern for arch was 3.35%, 5.40% for the tented arch, 29.09% for the right loop, 30.97% for the left loop, 30.76% for the whorl and 0.43% for unclassifiable. Around 72.5% of the data originated from male donors and 27.1% from female donors. For 0.7% of the data the gender was not stated (unknown). The following table shows the distribution of the general pattern on the different fingers.



	1	2	3	4	5	6	7	8	9	10
🟦 Arch	2.7	6	4.1	1.2	0.8	4.6	6.1	5.4	1.6	1.1
🟥 Left Loop	0.4	15.3	1.2	0.9	0.2	55.3	33.8	64.7	55.9	82.1
🟩 Right Loop	49.4	31.4	67.5	46.5	79.2	0.6	14.8	1.2	0.5	0.1
🟪 Whorl	46.1	35.4	19.7	48.2	16.2	37.5	33.1	20.6	38.4	12.8
🟦 Tented Arch	1.2	11.4	7.1	3	3.2	1.7	11.8	7.8	3.2	3.4
🟧 Unclassifiable	0.2	0.5	0.3	0.3	0.4	0.2	0.4	0.3	0.3	0.4

Table 2.3.2.1: GP distribution (%) on the 10 different fingers.

2. 4. Existing Bayesian networks

Bayesian networks are statistical models that belong to the family of probabilistic graphical models. A Bayesian network has a graphical structure that allows us to represent and reason about an uncertain domain [Korb & Nicholson, 2010]. In particular, each node in the graph represents a random variable, while the arcs between the nodes represent probabilistic dependencies among the corresponding random variables. The strengths of these conditional dependencies in the graph are captured by (conditional) probabilities which are often estimated by human experts or by using statistical and computational methods.

More formally, a Bayesian network B is defined as a pair B = (G, P), where
G = (V(G), A(G)) is an acyclic directed graph with the set of vertices V(G) = {$X_1, \dots, X_n$} and the arcs A(G) ⊆ V(G) × V(G), and P is a joint probability distribution defined on the variables corresponding to the vertices V(G), as follows:

$$P(X_1, X_2, \dots X_n) = \prod_{i=1}^{n} P(X_i \mid \boldsymbol{\pi}(X_i))$$

where $\boldsymbol{\pi}(X_i)$ stands for the set of parents (direct ancestors) of X_i. All variables in a Bayesian network are typically discrete and the conditional probability- $P(X_i \mid \boldsymbol{\pi}(X_i))$ is distribution often represented as a table, listing the probabilities that a child node (X_i) takes on each of its feasible values, for each combination of values of its parents ($\boldsymbol{\pi}(X_i)$).

In a previous project, [Doekhie, 2012] designed two Bayesian to assist the fingerprint examiner in the investigation phase and evaluation phase as described in the introduction. In this section we first briefly discuss the two Bayesian networks designed by Doekie, followed by our observations concerning these two networks.

2. 4. 1. Existing one finger BN

Figure 2.4.1.1 presents the structure of a Bayesian network that was modelled at NFI for assisting the fingerprint examiner in forensic studies when a single fingermark is found on the crime scene [Doekhie, 2012; Haraksim et al., 2013].

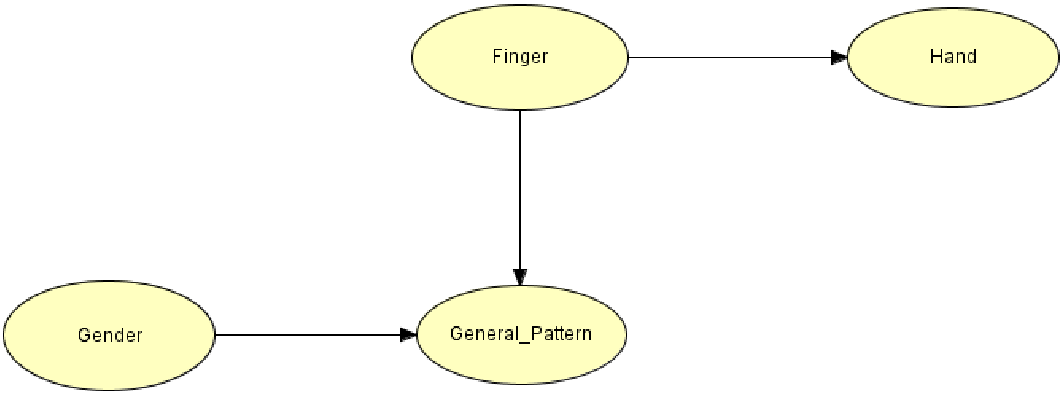


Figure 2.4.1.1: Bayesian network structure for one fingermark

For the *Finger* node, the categorical states 1 through 10 are defined, representing the ten finger numbers. For the *Hand* node, the states Right and Left are defined. The states of the *General_Pattern* node are Arch, Tented arch, Right loop, Left loop, Whorl, and Unclassifiable. The states of the *Gender* node are Male, Female and Unlabelled.

The independencies in the network were modelled by discussions with fingerprint examiners. However, Doekhie did not provide any specific details of this discussion with fingerprint examiners on the qualitative modelling of the network. The probability distribution of the *Finger* is obtained from the fingermark dataset. This information of the frequency of which finger left the fingermark was used to refine the equiprobable prior of 1/10 to a more realistic prior. The conditional probability distribution for Hand given Finger is a deterministic relationship: any finger number between 1 to 5 belongs to the right hand otherwise the fingers belong to the left hand. The prior probability distribution of *Gender* and the conditional probability distributions for *General_Pattern* were extracted from the fingerprint dataset by simple frequency counting.

2. 4. 2. Existing two fingers BN

The previous network was modelled for a single fingermark found on a crime scene. When multiple fingermarks are left on the crime scene in a single act of touch, these fingermarks are conditionally dependent, since they are from the same individual. A multiple consecutive fingermarks network should therefore be able to give stronger inferences than processing each fingermark separately using a single fingermark network.

[Doekhie, 2012] therefore extended the one finger network for two consecutive fingermarks by constructing a structure as illustrated in figure 2.4.2.1. The nodes *Finger* and *General_Pattern* of the one finger Bayesian network were duplicated and re-named to *FingerA* and *FingerB* for the *Finger* node, and *General_PatternA* and *General_PatternB* for the *General_Pattern* node. The arrows from the *Gender* node point towards both general pattern nodes in which the conditional probability distributions are identical to the distribution in the one finger network. In this Bayesian network, the prior on the *Hand* node is obtained from the fingermark dataset (Right_Hand- 52.17%, Left_Hand- 47.81%). We know that fingers 1 to 5 come from the right hand, so P(*FingerMark* = Finger1| *Hand* = Right),

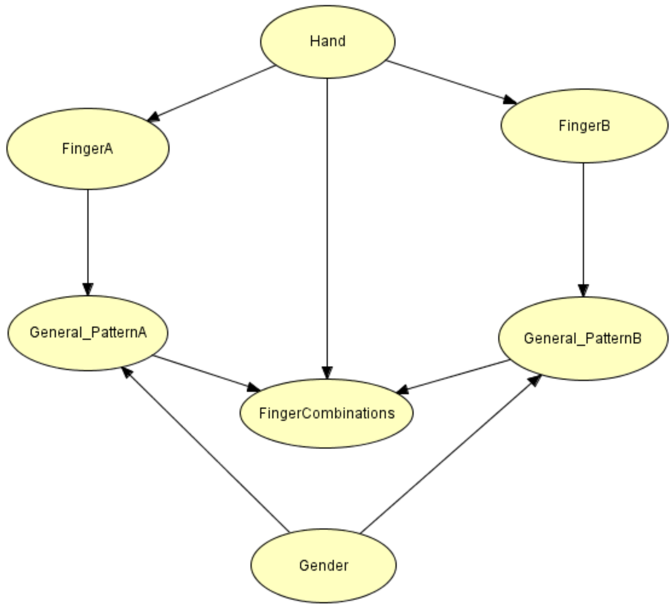


Figure 2.4.2.1: Bayesian network structure for two fingermarks

P(*FingerMark* = Finger2| *Hand* = Right) to P(*FingerMark* = Finger5| *Hand* = Right) are the relative frequency of occurrences of fingers 1 to 5 in the fingermark dataset. While the conditional probability values for the other fingers (finger 6 to 10) given it is the right hand is zero, since these fingers cannot occur on the right hand. Similarly, given that it is a left hand, the conditional probability values for the fingers 6 to 10 are obtained from their relative frequency of occurrence in the fingermark dataset (as shown in Table 2.4.2.1).

		Finger Number (Fingermark)									
Hand		**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
	Right	0.299	0.325	0.204	0.132	0.040	0	0	0	0	0
	Left	0	0	0	0	0	0.319	0.201	0.244	0.169	0.067

Table 2.4.2.1: Conditional probability table of P(Fingermark | Hand)

Sometimes in casework fingerprint examiners can make the assumption that the two fingermarks found on the crime scene were left by two consecutive fingers of the same hand. Under this assumption, there are eight possible finger combinations. [Doekhie, 2012] used finger number 1 and 2, 2 and 3, 3 and 4, 4 and 5, 6 and 7, 7 and 8, 8 and 9, 9 and 10 as the eight possible combinations for the *FingerCombinations* node. The conditional probability values for the *FingerCombinations* are assessed as follows: *fingerX* and *fingerY* can hold values according to the 8 possible combinations of fingers discussed above. Therefore P(*FingerCombinations* = fingerX_and_fingerY | *GeneralPatternA*, *GeneralPatternB, Hand*) is the frequency of the *GeneralPatternA* to occur on *fingerX* and *GeneralPatternB* to occur on *fingerY*, divided by the frequency of obtaining *GeneralPatternA* and *GeneralPatternB* on all the four finger combinations of each hand. These frequencies were obtained from the fingerprint dataset.

2. 4. 3. Observations about existing BNs

We will discuss below our observations concerning Doekhie's two Bayesian networks.

- In Bayesian networks only domain variables whose states are mutually exclusive and collectively exhaustive can be modelled. In the previously designed networks the states of the *Gender* variable were Male, Female and Unlabeled. In this case, unlabeled corresponds to individuals whose gender information is not known. Since in reality people are either male or female, the *Gender* node can only be modelled correctly in the network if the state unlabeled is removed (with the assumption that the gender information is missing at random).

- [Doekhie, 2012] claimed that 'there were no major differences between the general pattern distributions of the different genders' without any statistical tests. If this is true, the Bayesian networks would not require an arc (arrow) connecting the gender and general pattern. On discussion with the forensic examiners, we found out that the *Gender* variable had been incorporated in the network 'just to show in the court' that the additional evidence of gender does not impact the distribution over the fingers given some general pattern. We will perform some statistical tests (section 3.1.2) to get a better understanding of this.

- Recall that the fingermark database does not have any information which can help us in determing the frequency of two or more consecutive fingermarks. Though [Doekhie, 2012] (incorrectly) used information from the fingermark database to assess the probabilities of *FingerA* and *FingerB*- without any motivation for this choice.

- The *General_Pattern* nodes and *Hand* node form a Markov Blanket over the *FingerCombinations* i.e. the *General_Pattern* nodes and *Hand* node shield the *FingerCombinations* node from the rest of the network. Hence hypothetically given the markov blanket of *FingerCombinations*, any information about the *FingerA*, *FingerB* or *Gender* will not affect *FingerCombinations*. Similar to section 2.4.1 if we refine the equiprobable prior which fingers left the fingermarks to a more realistic prior (provided that in future our fingermark dataset contains consecutivity information) then that information will not influence the *FingerCombinations*.

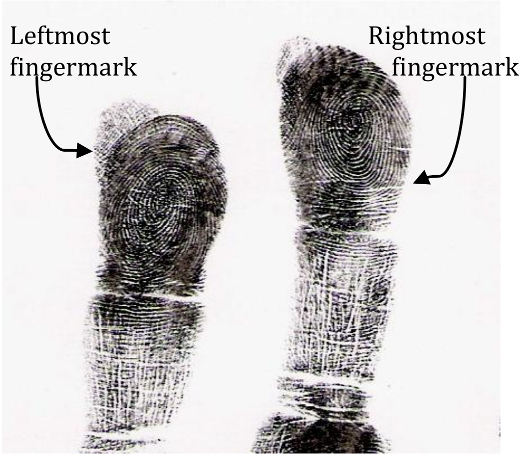


Figure 2.5.3.1: Figure showing the leftmost and rightmost fingermark.

- For real life applications, the combination set to denote consecutive fingers should be according to the spatial position of the fingers as shown in Figure 2.2.1. Since, for example, when two fingermarks (leftmost fingermark and rightmost fingermark as shown in figure 2.5.3.1) are found on the crime scene, then finger numbers 6 and 5 can never be the finger which left the leftmost fingermark. Similarly finger numbers 10 and 1 can never be the finger which left the rightmost fingermark. This spatial arrangement of fingers is not taken into account in the *FingerCombinations* node in [Doekhie, 2012].

- A disadvantage of generalising the 2 fingers BN to more than two fingers is that the conditional probability table (CPT) of the joint node (*FingerCombinations*) will increase exponentially with the number of fingers. For 2 fingers, the size of the CPT of the *FingerCombinations* node is: 8 possible finger combinations x 6 possible general patterns in *FingerA* x 6 possible general patterns in *FingerB* x 2 for the *Hand* = 576. Similarly, for 3 fingers the size of the CPT of the *FingerCombinations* node would become: 6 possible finger combinations x 6 possible general patterns in *FingerA* x 6 possible general patterns in *FingerB* x 6 possible general patterns in *FingerC* x 2 for the *Hand* = 2,592. The size of the CPT of the *FingerCombinations* node for 4 fingers would be 10,368 (4 x 6^4 x 2) and for 5 fingers would be 41,472 (2 x 6^5 x 2). Therefore if we take the same approach as [Doekhie, 2012] for generalization, then the CPT will explode.

In the subsequent chapter, we will try to resolve the above mentioned issues. Moreover we will also extend the problem to multiple (3, 4 and 5) finger networks.

Chapter 3:

Extension to multiple finger networks

3. 1. Motivation

Like software engineering, building a Bayesian network is a cyclic process that iterates over the following tasks:

- Identifying variables and values
- Constructing the acyclic digraph (or directed graph)
- Assessing the probabilities

This process is continued until validation shows that the results of the network are considered satisfactory for the domain.

In this thesis, the problem at hand is to establish what finger(s) is most likely to be the source of some general pattern(s) of a fingermark(s). On discussions with fingerprint examiners at NFI, we understood that when multiple fingermarks are left on the crime scene in a single act of touch, these fingermarks are conditionally dependent, since they are from the same individual. This would mean that a multiple consecutive finger network would be able to give stronger inferences than processing each fingermark separately using a single finger network.

In this chapter we discuss the modelling of multiple finger networks. In the next section we first identify the relevant domain variables. Followed by discussing how we generalised the 1 finger network to multiple finger networks.

3. 2. Relevance of Variables

When modelling Bayesian networks for a certain problem domain, the first step is to identify the essential domain variables and their possible values. This means that before we start constructing a network, we need to know which variables are relevant to solve the problem. Because on the one hand adding irrelevant variables will uselessly increase the complexity of the network, and on the other hand not including the essential variables may lead to incorrect modelling of the problem.

In the project domain, it can be undoubtedly seen that the *General_Pattern* and *Finger_No* variables are relevant. As mentioned in Section 2.2.2, the *General_Pattern* variable can hold six possible values (Arch, Tented_Arch, Whorl, Right_Loop, Left_Loop and Unclassifiable); the state Unclassifiable is assigned only when the fingerprint examiner is unable to assign one of the other five classes defined. And the FingerNo can hold a subset of all 10 possible fingers.

In this section we discuss the relevance of the two other variables *Hand* and *Gender* included in the previous networks. After which we discuss the construction of new networks with the variables we considered relevant.

3. 2. 1. Hand

Firstly if we know from which finger(s) the fingermark is from, then implicitly we also know which hand it is from (by following Henry's convention). However, generally it is not possible for the fingerprint examiners to be able to infer which hand was at the origin of the fingermark(s) recovered from the crime scene from a mere observation of the fingermark(s), though additional traces of the palm print and print of the second joint in the fingers could help the fingerprint examiners make a reliable estimation. In these situations, the fingerprint examiner could narrow down the possibilities from which finger(s) the general pattern belonged to. These situations can be dealt with in a better way using the concept of soft evidence [Bilmes, 2004] (or negative evidence) on the finger nodes, instead of just using another variable for *Hand*. Therefore, we decided not to include the *Hand* node in our networks to keep them simple and compact.

3. 2. 2. Gender

The relevance of the gender variable is important to study since two the datasets (fingermark and fingerprint) have very different distributions of gender. 92.8% of the fingermarks where from male suspects compared to only 72.5% of male criminals in the fingerprint dataset. Also 2.2% of the marks belonged to female suspects, whereas the fingerprint dataset contained around 27.1% female criminals. If we remove the criminals whose gender is unlabelled then we get 97.7% male suspects and 2.3% suspects in fingermark dataset. And 72.75% male criminals and 27.25% female criminals in the fingerprint dataset. It's important to note that we do not take into account the criminals whose gender is unlabelled, since we assume that they are missing at random.

The aim of this project is to obtain a distribution over the fingers (in order to establish the most likely one) given evidence about some general pattern. In this section we analyse how significantly the additional evidence of gender impacts the distribution over the fingers. To perform this analysis we use the fingerprint dataset, since the small number of only 247 female suspects compared to 10,722 male suspects in the fingermark dataset could mean that the dataset is not suitable for performing a gender analysis.

We know that the fingerprint dataset contains all information about all the 10 fingers for both males and females. This means that in this dataset the Gender and Finger_No variables are not correlated (independent). However given some evidence of general pattern, Gender and Finger_No become dependent.

We first analyse the strength of the dependency between gender and general pattern using the chi-square test for independence (section 3.2.2.1 and 3.2.2.2). If the gender and the general pattern are independent, then gender will not influence the distribution over the fingers (since gender is also independent of the distribution over the fingers).

After which we look at how strongly the additional evidence of gender influences the distribution over the fingers, given evidence of the general pattern (section 3.2.2.3).

3. 2. 2. 1. Dependence of general pattern on gender

The issue of the dependency between gender and general pattern has hardly been studied by researchers. We are only aware of the study by [Nithin et al., 2009], who found that 'irrespective of the sexes the fingerprint general pattern did not show any difference'. The problem with this study, however, is that they conclude this from the results of a t-test. Which is not a suitable test for independence when the data consists of nominal variables. We will therefore analyse the dependency between the two variables *Gender* and *General_Pattern* using the chi-square test for independence.

Pearson's Chi-square test of independence assesses whether paired observations on two nominal random variables, expressed in a contingency table, are independent of each other. Table 3.2.2.1 gives an example of a contingency table for the variables *Gender* and *General_Pattern*; the entries for the 12 possible combinations of the variables' values are called *bins*. The Pearson chi-squared statistic for testing the null hypothesis H_0 that there is no association between the two variables, is

$$X^2 = \sum \frac{(O_i - E_i)^2}{E_i},$$

where O_i is the observed frequency for bin *i* and E_i is the expected frequency for bin *i* calculated assuming the null hypothesis of independence is true. In the Chi-Square test of Independence, the degree of freedom (df) equals to (r-1)(c-1)[2], in our situation r is the number of values for gender and c is the number of values for general pattern.

The larger the differences between observed and expected frequencies, the larger the deviation from the null hypothesis, the more unlikely it becomes that the variables are independent. We will reject the null hypothesis if the p-value of the chi-squared test statistic, which indicates the probability that the deviations are due to chance alone, is less than the significance level of 0.05.

[2] This formula is derived from the definition of degree of freedom, which is the difference between the number of parameters under the alternative hypothesis (not independent) and null hypothesis H_0

		General Pattern						
		Arch	**Left_Loop**	**Right_Loop**	**Whorl**	**Tented _Arch**	**Un classifiable**	**Total per gender**
Gender	**Male**	63223	687928	620760	713087	118881	7941	2211820
	Female	38219	253810	264653	224844	45010	2114	828650
	Total per GP	101442	941738	885413	937931	163891	10055	**3040470**

Table 3.2.2.1.1: Contingency table showing the distribution of general pattern on males and females

As mentioned before, we intend to analyse the dependence between gender and the general patterns on all the fingers together. Our null hypothesis in this case is that there is no dependence between gender and the general pattern.

Using the contingency table with observed frequencies from the fingerprint dataset (Table 3.2.2.1.1), convincing evidence was found rejecting the null hypothesis that gender is independent of the general pattern distribution (χ-squared = 13999.82, df = 5, $p < 2.2e-16$). We therefore conclude that Gender and General_Pattern are dependent, i.e. $P(GP) \neq P(GP \mid Gender)$, or equivalently, $P(Gender) \neq P(Gender \mid GP)$. Our results are contrary to that of [Nithin et al., 2009] because they incorrectly used the t-test to perform this study.

3. 2. 2. 2. Dependence of general pattern on gender per finger

In the previous section we saw that the general pattern, measured over all fingers together, depends on the gender of a person. However [Doekhie, 2012] stated that 'they extracted the general pattern distributions of the different fingers for the male, female and unlabeled populations and found no major differences between the general pattern distributions of the different genders'. Since this was stated without any statistical tests, we performed chi-squared tests to gain more insight into the dependence of gender on the general patterns per finger.

For each finger, our null hypothesis in this case is again that there is no dependence between gender and the general pattern. In other words, the null hypotheses assume that the distributions per finger of general pattern for males and females are the same. It should be noted that we are testing the dependence between two variables- gender and general pattern, in context of a third variable- finger. Which is the reason why we perform a chi-squared test for each finger; i.e. 10 chi-squared tests in total

		Finger Number				
Chi-square test results		**1**	**2**	**3**	**4**	**5**
	χ-squared	2875.121	2683.625	2332.715	2570.287	2293.06
	df	5	5	5	5	5
	p-value	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16
		6	**7**	**8**	**9**	**10**
	χ-squared	1221.239	1111.452	1326.545	939.5194	1037.213
	df	5	5	5	5	5
	p-value	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16	< 2.2e-16

Table 3.2.2.2.1: Results of Chi-squared tests for independence between the gender and the distribution of the general patterns per finger performed using R statistical package

Using the 10 contingency tables with observed frequencies from the fingerprint dataset (Appendix B), convincing evidence was found rejecting the null hypotheses that, per finger, gender is independent of the general pattern (values for χ-squared, df and p are summarised in Table 3.2.2.2.1; note that p < 2.2 e-16 for all fingers). We therefore conclude that Gender and General_Pattern are dependent, even per finger, i.e. P(GP | Finger_no) ≠ P(GP | Gender, Finger_no). We obtained results contrary to that of [Deokhie, 2012], since Doekhie may have remarked by just superficially comparing the probability distribution of P(GP | Finger_no) and P(GP | Gender, Finger_no).

3. 2. 2. 3. Influence of gender on the distribution of fingers given some general pattern

In the previous sections we saw that the general pattern (over all or per finger) depends on the gender of a person. Our aim in this project is to obtain a distribution over the fingers given evidence about some general pattern. We now look deeper into this, by analysing how strongly the additional evidence of gender influences the distribution of fingers, given evidence of the general pattern. More formally, we want to know whether or not P(Finger_no|GP) equals P(Finger_no|GP, Gender).

It can be seen from Figures 3.2.2.3.1 to 3.2.2.3.6 that there are only small absolute differences (around 0.01%) in the probability distributions over the fingers given both the general pattern and gender, and the probability distributions over the fingers given just the general pattern. We will test the apparent equalities using the Kullback-Leibler (KL) divergence. We choose to use KL divergence, instead of the chi-square test (used previously), since we want to compare two probability distributions.

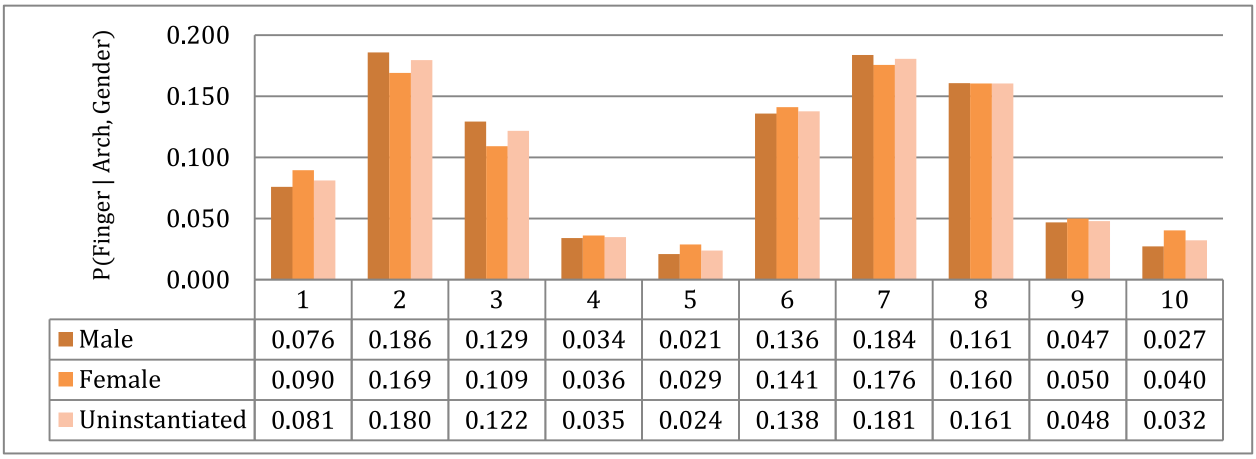
KL divergence can be used to measure the information gain in moving from a prior distribution to a posterior distribution [Cover & Thomas, 1991]. More specifically, the KL divergence measures the information change between distributions before and after applying some influence. The lower the KL divergence score, the more similar the probability distributions are; a zero KL divergence score is obtained for two identical distributions. However in extreme situations, the KL divergence score can also tend to infinity.

We measure the information gained on the distribution over the fingers with the additional evidence of gender i.e. compute
$\mathrm{KL}_{\mathrm{div}}(\, \mathrm{P}(\mathrm{Finger\_no}|\mathrm{GP}, \mathrm{Gender}) \;\|\; \mathrm{P}(\mathrm{Finger\_no}|\mathrm{GP})\,)$ for each general pattern.

Since the KL divergence scores (see Table 3.2.2.3.1) are in the range of 10^{-3}, we can say there is hardly any information gain by the additional evidence of Gender = Male, i.e. P(Finger | General Pattern, Male) ≈ P(Finger | General Pattern). Moreover for the female gender the two distributions are in fact identical for all general patterns except for the Arch, which gets a KL divergence score of 0.003. Hence, given a specific general pattern, the additional evidence of the gender has a very weak influence on the finger distribution. Or in other words there is hardly any information gain in the distribution over the fingers given some additional evidence of gender.

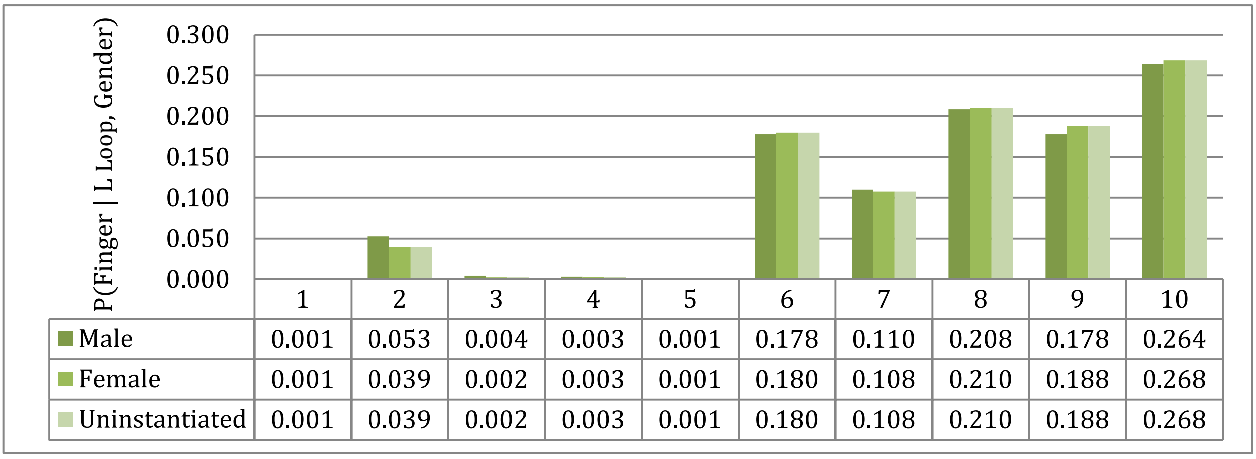
KL Divergence given Male from the Prior (uninstantiated gender)	**Value**		**KL Divergence given Female from the Prior (uninstantiated gender)**	**Value**
Arch	0.0016		Arch	0.0029
Left_Loop	0.0027		Left_Loop	0
Right_Loop	0.0019		Right_Loop	0
Whorl	0.0049		Whorl	0
Tented_Arch	0.0086		Tented_Arch	0
Unclassifiable	0.024		Unclassifiable	0

Table 3.2.2.3.1: KL Divergence measure of information gain i.e.
KL_{div} (P(Finger_no|GP, Gender) || P(Finger_no|GP))



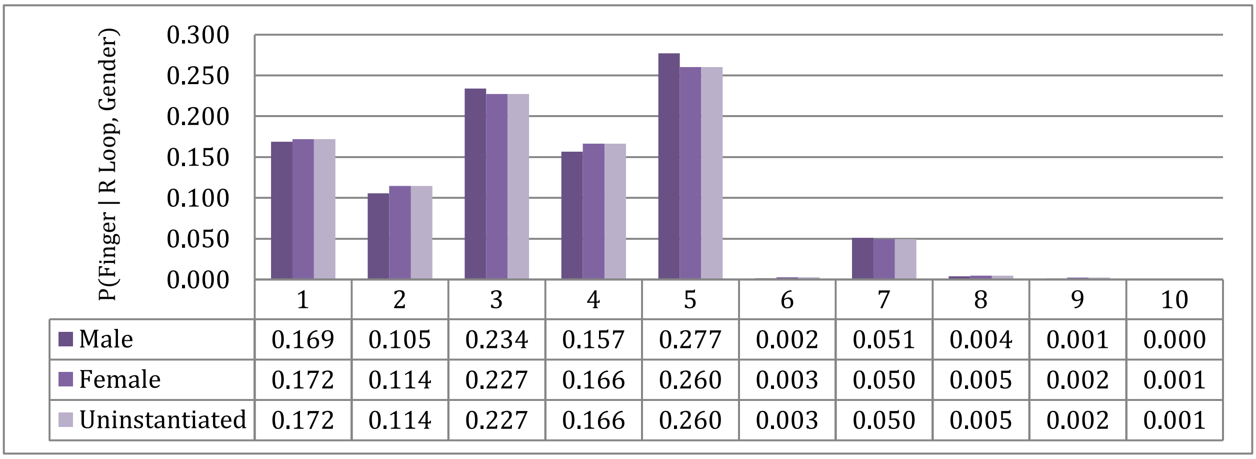
	1	2	3	4	5	6	7	8	9	10
■ Male	0.076	0.186	0.129	0.034	0.021	0.136	0.184	0.161	0.047	0.027
■ Female	0.090	0.169	0.109	0.036	0.029	0.141	0.176	0.160	0.050	0.040
■ Uninstantiated	0.081	0.180	0.122	0.035	0.024	0.138	0.181	0.161	0.048	0.032

Figure 3.2.2.3.1: Distribution of the fingers for all genders, given General Pattern = Arch



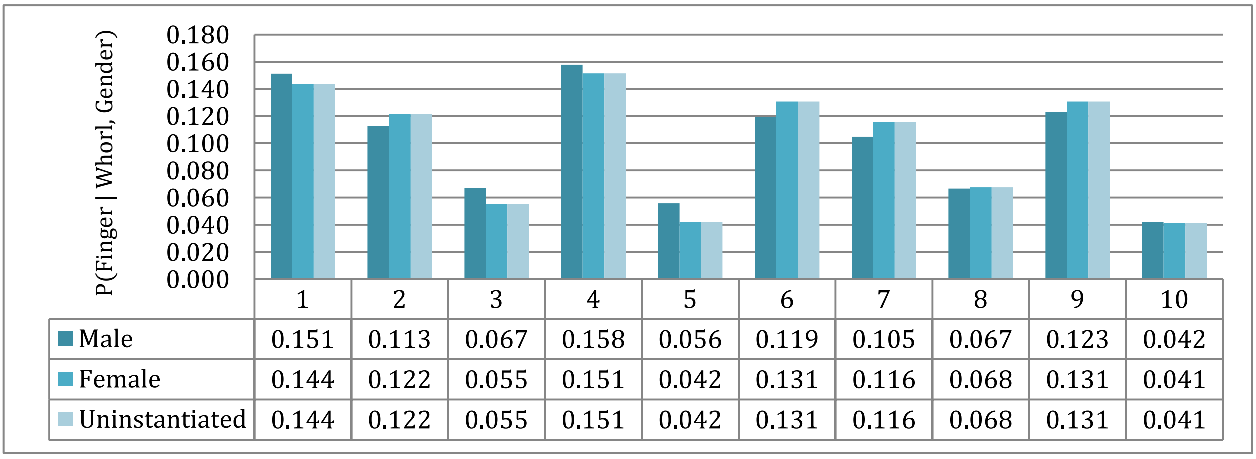
	1	2	3	4	5	6	7	8	9	10
■ Male	0.001	0.053	0.004	0.003	0.001	0.178	0.110	0.208	0.178	0.264
■ Female	0.001	0.039	0.002	0.003	0.001	0.180	0.108	0.210	0.188	0.268
■ Uninstantiated	0.001	0.039	0.002	0.003	0.001	0.180	0.108	0.210	0.188	0.268

Figure 3.2.2.3.2: Distribution of the fingers for all genders, given General Pattern = Left Loop



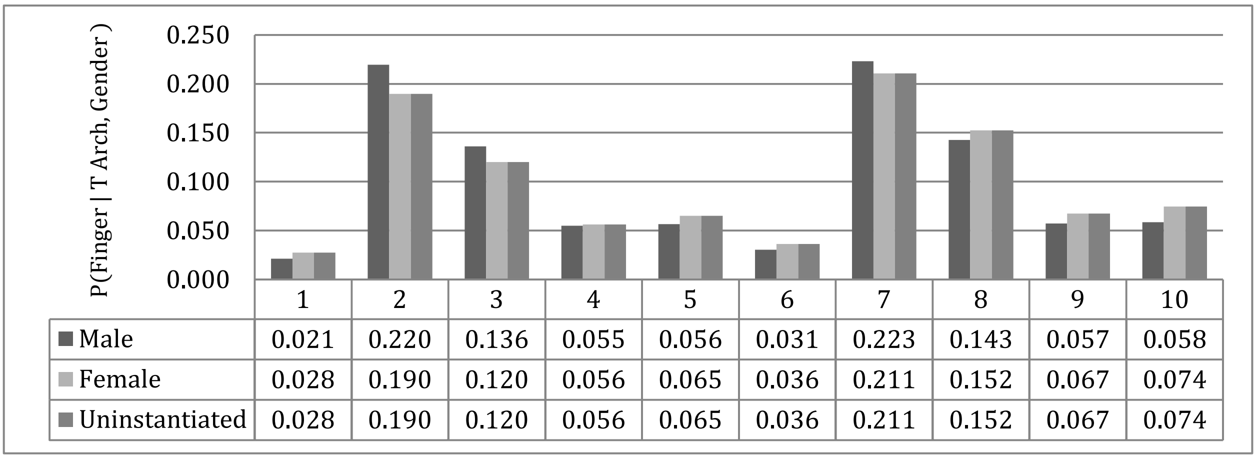
	1	2	3	4	5	6	7	8	9	10
■ Male	0.169	0.105	0.234	0.157	0.277	0.002	0.051	0.004	0.001	0.000
■ Female	0.172	0.114	0.227	0.166	0.260	0.003	0.050	0.005	0.002	0.001
■ Uninstantiated	0.172	0.114	0.227	0.166	0.260	0.003	0.050	0.005	0.002	0.001

Figure 3.2.2.3.3: Distribution of the fingers for all genders, given General Pattern = Right Loop



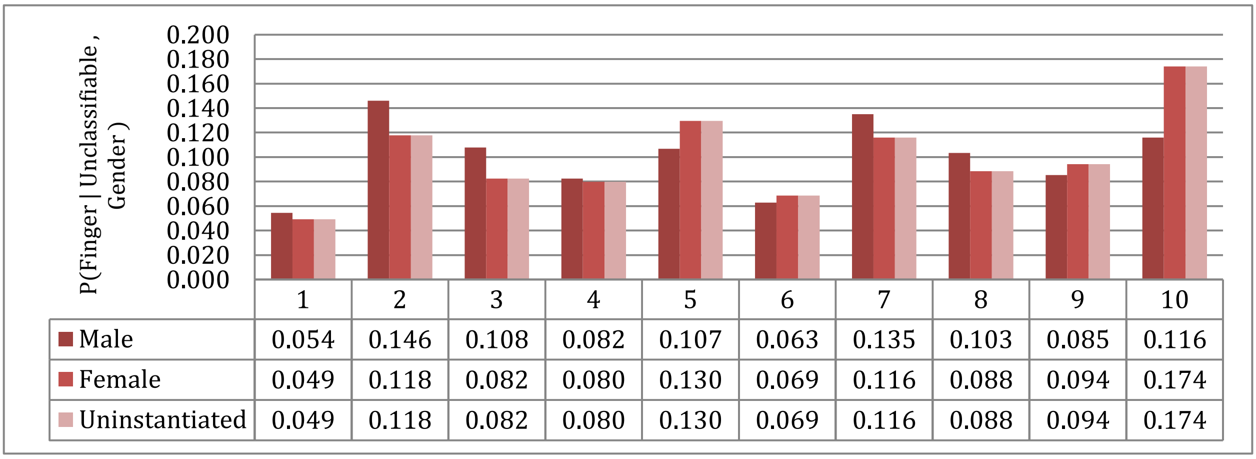
	1	2	3	4	5	6	7	8	9	10
■ Male	0.151	0.113	0.067	0.158	0.056	0.119	0.105	0.067	0.123	0.042
■ Female	0.144	0.122	0.055	0.151	0.042	0.131	0.116	0.068	0.131	0.041
■ Uninstantiated	0.144	0.122	0.055	0.151	0.042	0.131	0.116	0.068	0.131	0.041

Figure 3.2.2.3.4: Distribution of the fingers for all genders, given General Pattern = Whorl



	1	2	3	4	5	6	7	8	9	10
■ Male	0.021	0.220	0.136	0.055	0.056	0.031	0.223	0.143	0.057	0.058
■ Female	0.028	0.190	0.120	0.056	0.065	0.036	0.211	0.152	0.067	0.074
■ Uninstantiated	0.028	0.190	0.120	0.056	0.065	0.036	0.211	0.152	0.067	0.074

Figure 3.2.2.3.5: Distribution of the fingers for all genders, given General Pattern = Tented Arch



	1	2	3	4	5	6	7	8	9	10
■ Male	0.054	0.146	0.108	0.082	0.107	0.063	0.135	0.103	0.085	0.116
■ Female	0.049	0.118	0.082	0.080	0.130	0.069	0.116	0.088	0.094	0.174
■ Uninstantiated	0.049	0.118	0.082	0.080	0.130	0.069	0.116	0.088	0.094	0.174

Figure 3.2.2.3.6: Distribution of the fingers for all genders, given General Pattern = Unclassifiable

3. 2. 2. 4. Discussion

Although the evidence of gender is not found on the crime scene, the *Gender* variable was incorporated in the previous networks [Doekhie, 2012] 'just to show in the court' that, given some general pattern, the additional evidence of gender does not affect the distribution over the fingers.

We have seen in this section that the dependency between gender and general pattern is strong. Once we know the general pattern, however, there is hardly any change in the distribution over the fingers if additional evidence of gender is given. So in the current project context, since the interest is always to get a distribution over the fingers given some general pattern, we can do without the variable gender.

3. 3. Multiple finger networks

We saw in the previous section that for modelling a simple and compact network to predict the most likely finger that left a fingermark with a certain general pattern, we can assume that only the *General_Pattern* and *Finger_No* variables are relevant. This results in the single-finger network illustrated in Figure 3.3.1.

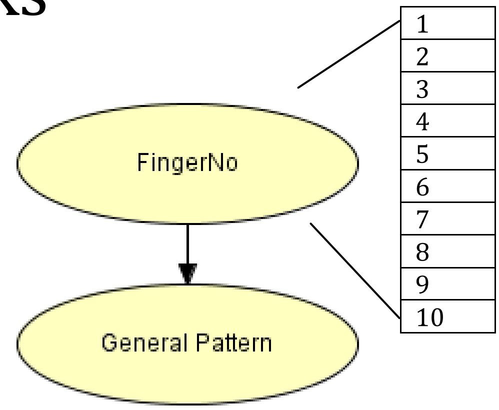


Figure 3.3.1: 1 Finger network

A disadvantage of generalising to multiple finger networks in line with the approach used by Doekhie, is that it would result in the conditional probability table (CPT) of the joint node *(FingerCombinations*) to increase exponentially with the number of fingers. Hence we generalise to multiple finger networks in a different way.

On discussions with fingerprint examiners at NFI, we found that they believe that there is no significant dependency (or correlation) between the general patterns on different fingers. Modelling the *General_Pattern* nodes as independent is not very easy, since firstly we will have to change the direction of causality between *General_Pattern* and *Finger_No*, and secondly like Doekhie's 2 Finger network the CPT for the FingerNo(s) would increase exponentially). For this reason we will model the network in such a way that the *General_Pattern* nodes for the different marks found are independent given the *Finger_No* nodes (i.e. not marginally independent).

3. 3. 1. A straightforward approach

With the above assumption in mind, the easiest way to extend to n>1 fingers is to replicate the single finger network from Figure 3.3.1 n-times, where n is the number of fingermarks found on the crime scene. The two finger network, for example, will then have the structure shown in Figure 3.3.2. As per our assumption there is no arc between the *LeftMost_GeneralPattern* and *RightMost_GeneralPattern*.

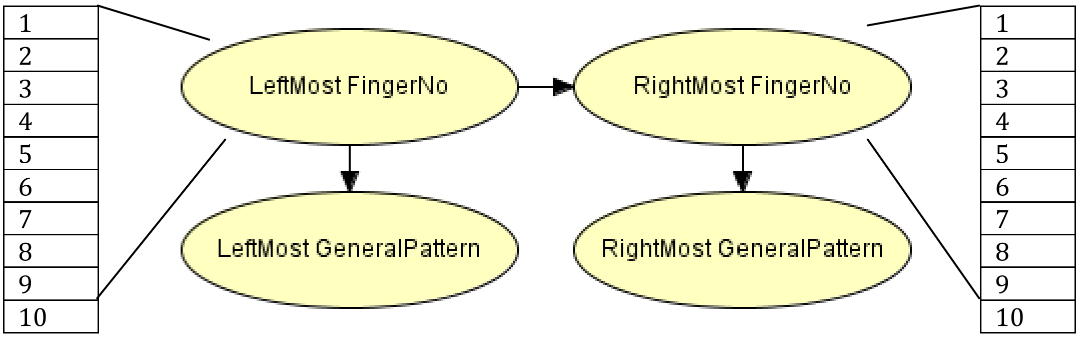


Figure 3.3.2: 2 Finger network made by replicating the 1 Finger network

The advantage of this approach is that the probability distribution P (*GeneralPattern* | *FingerNo*), of the 1 Finger Network can be replicated in P(*LeftMost_GeneralPattern* | *LeftMost_FingerNo*) and P(*RightMost_GeneralPattern* | *RightMost_FingerNo*). Hence we do not need to go back to the data to determine these distributions. We do, however, need to determine the prior probability distribution on the *LeftMost_FingerNo* and the conditional probability distribution P(*RightMost_FingerNo* | *LeftMost_FingerNo*).

If we consider the general case, i.e. when two fingermarks are found on the crime scene and the fingerprint examiner can say that they have been left on the crime scene in a single act of touch and come from the same hand. Then according to the spatial arrangement of the fingers the *LeftMost_FingerNo* can never be the left thumb-6 or right pinky finger-5. Whereas the *RightMost_FingerNo* will depend on the *LeftMost_FingerNo*. For example if the *RightMost_FingerNo* is left ring finger-9, then the *LeftMost_FingerNo* can only be left middle finger-8, index finger-7 or thumb-6.

However for fingerprint examiners, this general case is very difficult to determine. It is more practical and realistic for them to say something about the consecutivity of the two fingermarks. This is the main reason why we focus on consecutive fingermarks from the same hand. Apart from the fact that elicitation of the conditional probability distribution of P(*RightMost_FingerNo* | *LeftMost_FingerNo*) i.e. 100 (10 finger x 10 finger) probability estimates from fingerprint examiners could be a difficult process. Though in reality we would only need to assess the probability estimates for a subset of these cases, since the spatial arrangement of the fingers makes a lot of cases impossible (like the *RightMost_FingerNo* can never be right thumb-1 or left pinky finger-10 etc.).

	8 possible finger combinations (2 Finger network)							
LeftMost_FingerNo	1	2	3	4	7	8	9	10
RightMost_FingerNo	2	3	4	5	6	7	8	9

Table 3.3.1: Possible values of *RightMost_FingerNo* given *LeftMost_FingerNo* for two consecutive fingers.

This assumption of consecutivity of fingers makes the parameterisation of the FingerNo variables quite simple. We use an equiprobable prior represented by the fingerprint dataset (since the fingermark dataset does not have information on the distribution of consecutive fingermarks found on the crime scene). For two consecutive fingers there are 8 possible finger combinations (Table 3.3.1). This means that all fingers in the LeftMost_FingerNo (except finger 5 and 6) have an equal prior probability of 1/8. The conditional probability distribution of P(*RightMost_FingerNo* | *LeftMost_FingerNo*) is deterministic according to the table 3.3.1, i.e. for example if the *LeftMost_FingerNo* is finger 1 than the *RightMost_FingerNo* can only be finger 2.

It should be noted that in the general case, we would want to find the maximum joint assignment to *LeftMost_FingerNo* and *RightMost_FingerNo* which would best explain the evidence of the general patterns. This problem is known as the maximum a posteriori (MAP) or most probable explanation (MPE) in Bayesian networks. This is one of the other reasons why we looked at this problem in more details in part II of this thesis.

3. 3. 2. An alternative approach to consecutivity specific modelling

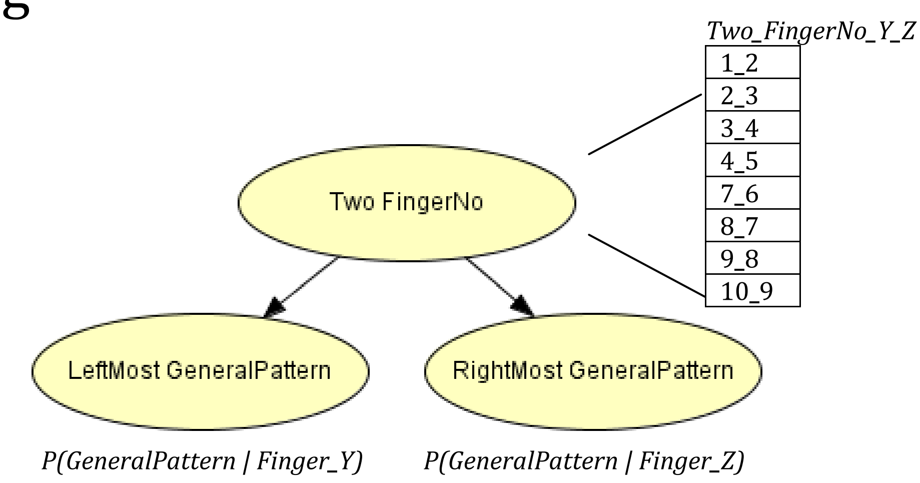


Figure 3.3.3: 2 Finger Network

Since we are only focussing on consecutive fingermarks, a better alternative to the network we just introduced in Figure 3.3.2 would be the network illustrated in Figure 3.3.3. Since in the previous network the relationship between *LeftMost_FingerNo* and *RightMost_FingerNo* is deterministic, we can combine the two variables as *Two_FingerNo*. This would lead to computationally less complex inference in Bayesian networks since we combine the two nodes.

Similar to the previous network (illustrated in Figure 3.3.2) since there are 8 possible consecutive finger combinations, an equiprobable prior probability of 1/8 is assigned to the *Two_FingerNo*. However the conditional probability distributions P(*LeftMost_GeneralPattern* | *Two_FingerNo*) and P(*RightMost_GeneralPattern* | *Two_FingerNo*) are different from the previous network. If the two consecutive fingers are *X* and *Y*, P(*LeftMost_GeneralPattern* | *Two_FingerNo*) would be the probability distribution of general patterns on *Finger_X* and similarly P(*RightMost_GeneralPattern* | *Two_FingerNo*) would be the probability distribution of general patterns on *Finger_Y*. All theses conditional probability distributions are obtained from the fingerprint dataset. Hence we can see that extending to multiple fingers using this network is not that trivial. However this network is computationally less complex and simpler for forensic examiners to understand, since they have to look at the probability distribution of a single variable- *Two_FingerNo* instead of looking at conditional joint distribution of *LeftMost_FingerNo* and *RightMost_FingerNo* (of the network illustrated in Figure 3.3.2).

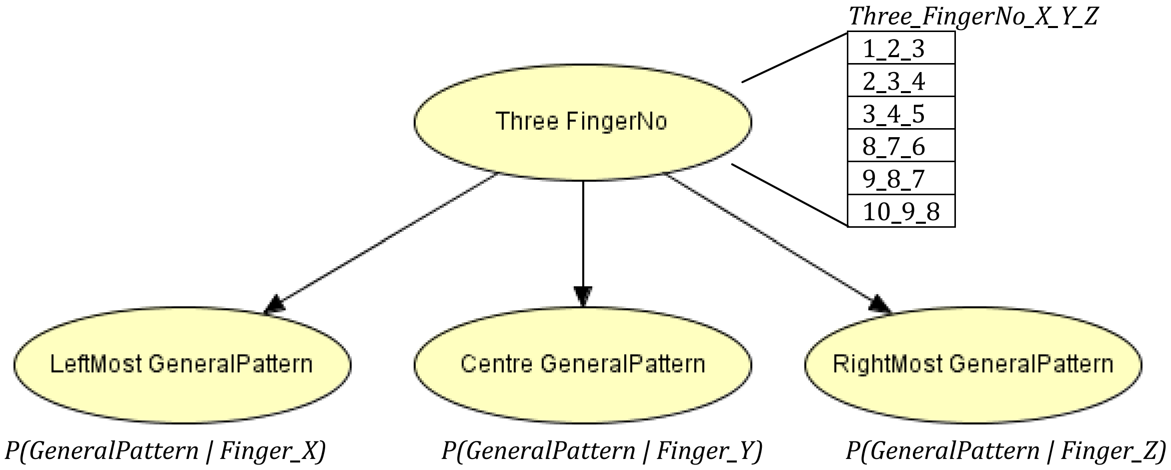


Figure 3.3.4: 3 Finger network

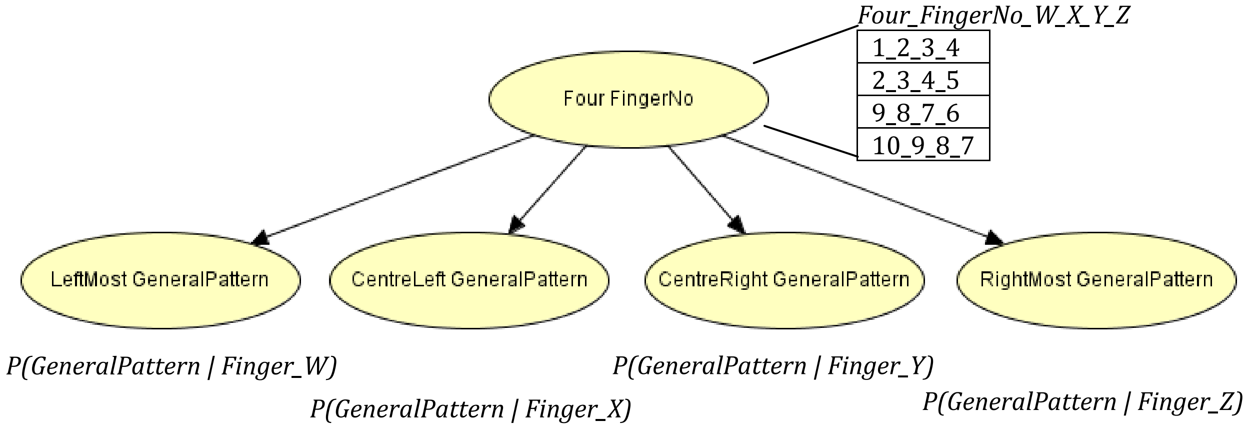

Four_FingerNo_W_X_Y_Z
1_2_3_4
2_3_4_5
9_8_7_6
10_9_8_7
Four FingerNo
LeftMost GeneralPattern
CentreLeft GeneralPattern
CentreRight GeneralPattern
RightMost GeneralPattern
P(GeneralPattern | Finger_W)
P(GeneralPattern | Finger_X)
P(GeneralPattern | Finger_Y)
P(GeneralPattern | Finger_Z)

Figure 3.3.5: 4 Finger network

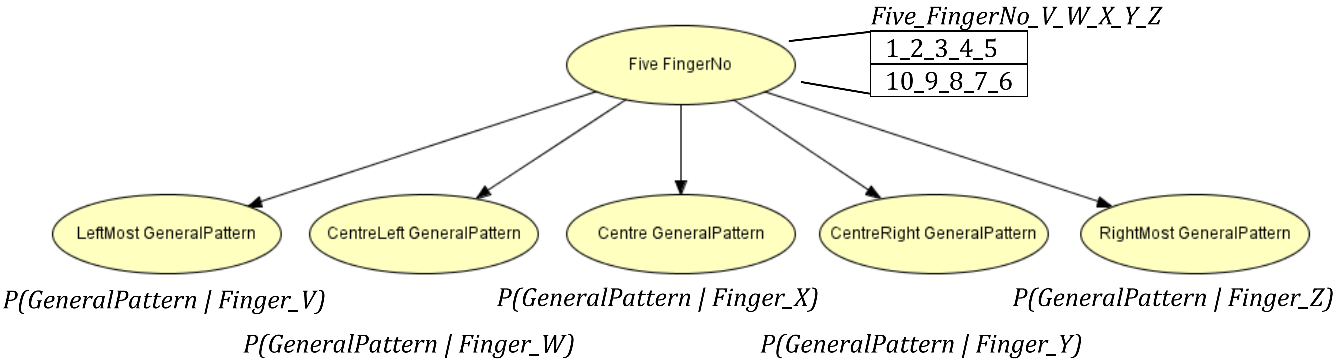

Five_FingerNo_V_W_X_Y_Z
1_2_3_4_5
10_9_8_7_6
Five FingerNo
LeftMost GeneralPattern
CentreLeft GeneralPattern
Centre GeneralPattern
CentreRight GeneralPattern
RightMost GeneralPattern
P(GeneralPattern | Finger_V)
P(GeneralPattern | Finger_W)
P(GeneralPattern | Finger_X)
P(GeneralPattern | Finger_Y)
P(GeneralPattern | Finger_Z)

Figure 3.3.6: 5 Finger network

Similarly we can model the 3 finger network (figure 3.3.4) with 6 possible finger combinations, the 4 finger network (figure 3.3.5) with 4 possible finger combinations and finally the 5 finger network (figure 3.3.6) with 2 possible finger combinations. It should be noted that the 5 finger network basically distinguishes from which hand the general patterns of the five fingermarks originate.

3. 3. 3. Discussion

We have extended the 1 Finger network to multiple (2, 3, 4 and 5) Finger networks. Depending on the number of finger marks found on the crime scene, the appropriate Finger network can be used; for example if three fingermarks considered as consecutive are found on the crime scene then the 3 Finger network will be used. Fingerprint examiners can assess if the fingermarks are consecutive or not, depending on the placement of the prints of the phalanx bones (distal, middle and proximal) in a finger, ridge flow or palm print. If the fingerprint examiners are not able to assess (assume) that the fingermarks are consecutive, then it is best to use the 1 Finger network and treat each mark independently.

Note though that if only 2 fingermarks can be seen on the crime scene, but if from their spacing the fingerprint examiner can say that there should be a another fingermark in between the two obtained- then to get a posterior probability distribution on the *Three_FingerNo*, we can enter the general pattern evidence only for the *LeftMost_GeneralPattern* and *RightMost_GeneralPattern* (leaving the *Centre_GeneralPattern* uninstantiated).

In the next chapter we propose a validation framework for validating these networks.

Chapter 4:
Validation
and
strength of evidence

To be able to use a Bayesian network in practical scenarios, an evaluation study is needed, for example, using data from the domain of the application. Such a study amounts to entering the data available for each problem case into the network and computing the most likely outcome. This outcome is then compared against a given standard of validity.

In the next sections, the validation framework along with the results for this domain will be discussed. First the appropriate measures used to validate the network will be introduced. Followed, by the validation results for the the investiagtive stage and quantifying the strength of evidence at the finger level. We also show how we can quantifying the strength of evidence of a particular general pattern at the person level. Validation of this part is left for future research.

4. 1. Validation framework

In this section, we present a validation framework, which will be used to validate the networks we created. These measures were recommended by forensic scientists

4. 1. 1. Measures using the posterior probability distribution

Bayesian networks are typically used to compute the posterior probability distribution over some variables of interest given a set of observations. The following two measures: (1) Percentage correct using posteriors and (2) Brier score provide two ways of evaluating the practical value of a Bayesian network. The percentage correct treats outcomes as deterministic, whereas the Brier score takes the actual uncertainties into account.

Posterior Accuracy

Bayesian networks can be considered probabilistic forecasters; hence the method similar to forecast verification method can be used to assess the quality of their forecasts.

Let's assume that there are i ∈ {1, ... N} cases in the test data used for validation. Let's assume that the class variable c_i' is the set of class values produced by the model for case i which has the most likely value c_i', and c_i is its true value.

$$Post\_Acc = \frac{1}{N} \sum_{i=1}^{N} \delta(c_i', c_i)$$

Where $\delta(c_i', c_i) = 1$ if $c_i' = c_i$ and 0 otherwise. In other words the percentage of cases where the outcome predicted by the network is correct according to the standard of validity is called the percentage correct.

Brier Score

Bayesian networks do not yield a deterministic outcome. Instead, they produce a posterior probability distribution for their outcome variable(s). This distribution reflects the network's doubt as to the most likely outcome. Since the percentage of correct outputs of a Bayesian network does not take the computed posterior distribution into consideration, it does not reveal the extent of uncertainty in the outcome. To incorporate the network's doubt in the assessment of its practical value, evaluation scores from the field of statistical forecasting are used. In the context of Bayesian networks the use of the Brier score [Panofsky, 1968] was introduced by [van der Gaag & Renooij, 2003].

The uncertainty expressed in the outcome can be taken into account in the evaluation. Let p_{ij} be the probability returned by the network for case i and value j of the outcome variable. Let s_{ij} be a function that returns a 1 if for case i the value j of the outcome variable is correct according to the used standard of validity and 0 otherwise.

The Brier score for case i is $B_i = \sum_j \, (p_{ij} - s_{ij})^2$

The average Brier score over N cases or forecasts is

$$B = \frac{1}{N}\sum_{i=1}^{N} B_i = \frac{1}{N}\sum_{i=1}^{N}\sum_{j} (\, p_{ij} - s_{ij}\,)^2$$

The Brier score lies within the interval [0, 2], where 0 indicates a perfect prediction.

4. 1. 2. Measures using Likelihood Ratios

[Robertson et al., 1995] claim that the presentation of expert evidence should be restricted to a likelihood ratio, which is the ratio of the probability supposing that the evidence for the given hypothesis is true to the probability supposing that the evidence for the contrary mutually exclusive hypothesis is true. This form of presentation of evidence is now becoming more common in many areas of scientific evidence, such as DNA, glass fragment analysis or speaker recognition.

Conditional probabilities can assist when estimating the probability that evidence came from an identified source. The probability estimate is based on calculation of a likelihood ratio [Aitken & Stoney, 1991]. In the likelihood-ratio framework the task of the forensic scientist is to determine the ratio of two probabilities of the same observation (general pattern information in our case) under different hypotheses. The likelihood ratio is calculated from the posterior and prior odds as follows:

$$\underbrace{\frac{p(\,H_1\ |\,E\,)}{p(\,H_2\,|\,E\,)}}_{\text{Posterior probability ratio}} = \underbrace{\frac{p(\,E\,|\,H_1\,)}{p(\,E\,|\,H_2\,)}}_{\text{Likelihood ratio (LR)}} \quad x \quad \underbrace{\frac{p(\,H_1\,)}{p(\,H_2\,)}}_{\text{Prior probability ratio}}$$

where LR is the likelihood ratio, E is the evidence, and in general hypothesis#1 (H_1) and hypothesis#2 (H_2) are a pair of mutually exclusive hypothesis about the origin of the trace material. However more specifically for calculating the evidential value, the first hypothesis H_1 can be considered as $H_p$which is the support by the prosecution and states that the trace material originates from the suspected person. The second hypothesis H_2 can be considered as H_d which is supported by the defence and states that the trace material originates from another individual, randomly chosen within the relevant population of potential sources of the trace. In other words, evidential value is calculated as the ratio of two probabilities: the probability of the evidence when the prosecution hypothesis is true divided by the probability of the evidence when the defence hypothesis is true.

LR Accuracy

The [NRC Report, 2009] and [Koehler, 2008] recommend the use of correct-classification rates/classification-error rates for measuring the validity of a forensic-comparison system. A likelihood ratio greater than one lends support to the hypothesis#1 H_1. Similarly, a likelihood ratio less than one lends support to the hypothesis#2 in the denominator. From the testing dataset, as a ground truth if we know that hypothesis#1 is true, then a LR > = 1 supports hypothesis H_1 and hence is correct. Whereas an LR < 1 supports the other hypothesis (H_2) and hence is classified as an error. In case of assigning strength of evidence to a trace on the crime scene, LR accuracy is the complement of the rate of misleading evidence in favour of the defence (RMED).

Log likelihood ratio cost C_{llr}

An appropriate metric of validity for use within the likelihood ratio framework, is gradient metric based on likelihood ratios, such as the log-likelihood-ratio cost (C_{llr}). The log-likelihood-ratio cost was developed for use in automatic speaker recognition by [Brümmer & Preez, 2006] and [van Leeuwen & Brummer, 2007] and has subsequently been applied to many domains of forensic science.

In order to calculate C_{llr} a test dataset is required from which one can draw a large number of pairs of samples known to have the same origin and a large number of pairs of samples known to have different origins. The pairs of test samples are presented to the system and the knowledge about the same source origin or different source origin status of the test pairs is compared with the output of the system.

The log-likelihood-ratio cost is calculated using the formula

$$C_{llr} = \frac{1}{2}\left(\frac{1}{N_{SO}} \sum_{i=1}^{N_{SO}} \log_2 \left(1 + \frac{1}{LR_{SO_i}}\right) + \frac{1}{N_{DO}} \sum_{j=1}^{N_{DO}} \log_2 \left(1 + LR_{DO_j}\right) \right)$$

where N_{SO} and N_{DO} are the number of same-origin and different-origin comparisons respectively, and LR_{SO} and LR_{DO} are the likelihood ratios derived from test pairs known to be same source origin and different source origin comparisons respectively. Note that part of the equation is the mean of the output of a function applied to all the likelihood ratios derived from same-origin comparisons (left side within the outer brackets), another part is the mean of the output of a function applied to all the likelihood ratios derived from different-origin comparisons (right side within the outer brackets), and C_{llr} is the mean of these two means. A plot of the same-origin and different origin penalty function is shown in Figure 4.1.2.1. Ideally, a same source comparison should result in a large positive log likelihood ratio and would contribute a very small penalty value to C_{llr}. For a same source comparison, a positive log likelihood ratio close to zero would not provide as much support for the same-origin hypothesis as a large positive log likelihood ratio, and it would contribute a somewhat larger penalty value to C_{llr}. For a same-origin comparison, a negative log likelihood ratio would contrary-to-fact lend support to the different-origin hypothesis and would contribute a larger penalty value to C_{llr}, with that penalty value increasing as the magnitude of the negative log likelihood ratio increases and lends greater support to the contrary-to-fact different-origin hypothesis. Mutatis mutandis for the different origin comparison.

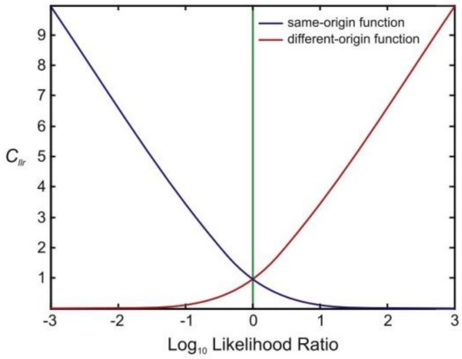


Figure 4.1.2.1: Plot of Cllr functions for same-origin and different-origin comparisons

Tippet Plots

Tippet plots have been classically used for empirical performance assessment [Evett & Buckleton, 1996], and consist of the cumulative distributions of LR values for same-source, and different-source experiments. In particular, two curves are plotted, one for same-source

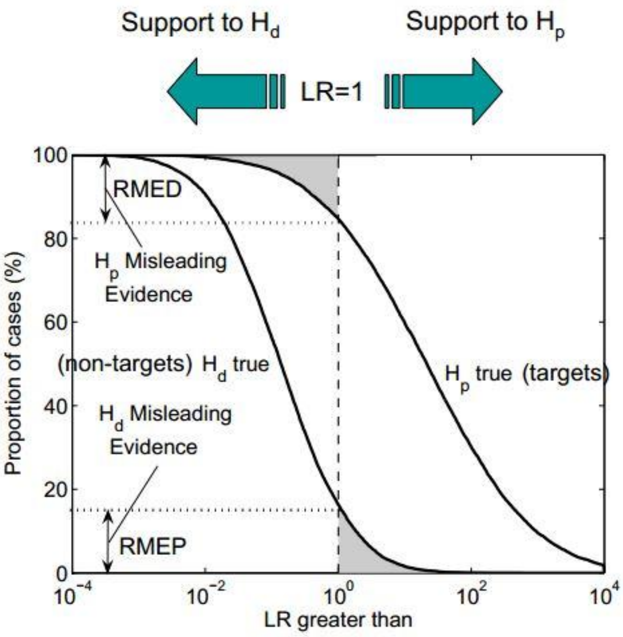


Figure 4.1.2.2: An example tippet plot, which is a graphical representation appropriate for the evaluation of the performance of the forensic recognition system

LR values (Hp is true) and one for different-source LR values (Hd is true). For each curve, and for each value x_0 in the Logarithmic x-axis, the proportion of LR values which are greater than x_0 is represented. It is worth noting that Tippet plots include the values of the rates of misleading evidence, which are determined by the value of each curve at x_0 =0 as shown in figure 4.1.2.2.

This Figure shows also an example of how Tippet plots yield more information about performance than the rates of misleading evidence. RMEP/RMED is the rate of misleading evidence in favour of the prosecution/defence. In an ideal system, the: Hp true curve > LR=1, Hd true curve < LR=1. Moreover the tippet plot shows the discriminative power by the separation between the two curves.

Empirical Cross-Entropy

ECE is a derivation of an information theoretical generalization of the C_{llr} value. These plots have an attractive and simple interpretation: the higher its value, the more information the factfinder needs in order to know the true value of the hypotheses. This information should be interpreted on average over different forensic cases (comparisons in the experimental set). If the LR values of the evidence evaluation process are misleading to the fact finder, then the ECE will grow, and more information on average will be needed in order to know the true values of the hypotheses. The details about the derivation and interpretation of ECE can be found in [Castro Ramos, 2007].

As shown in Figure 4.1.2.3, the solid red curve is the ECE (average information loss) of the LR values computed by the evidence evaluation method under assessment. This is

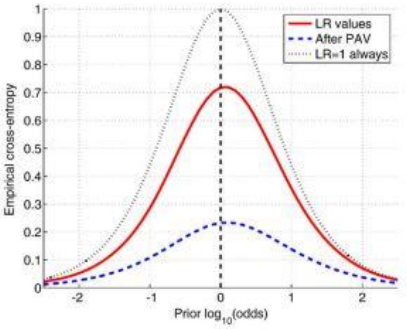


Figure 4.1.2.3: An example ECE plot.

the value which represents the overall performance of a set of LR values, and the lower its value, the better. ECE in terms of information theory is as follows: the higher this ECE curve, the higher the information needed in order to know the true hypotheses on average over cases, and therefore the worse the method. The dotted black curve represents the performance of a method always delivering LR=1, referred to as a neutral method. This performance is achieved when the evidence gives no information about whether H_p or H_d is true. If the ECE curve of the method under analysis presents a value greater than the curve of neutral performance, then the method will lose more information on average than basing the decisions only on the prior information, i.e., not using the evidence at all. In the range of prior probabilities where this happens, the method at hand should not be used for evidence analysis. Finally the dashed blue curve represents the calibrated performance, calculated by transforming the LR values in the experimental set. This calibration transformation can be conducted using a Pool Adjacent Violators algorithm (PAV). Analysis of this is beyond the scope of this thesis.

Detection error trade-off (DET) curves

In the case of biometric systems, a modified ROC curve known as a "detection error trade-off" curve is preferred. A DET curve plots error rates on both axes, giving uniform treatment to both types of error. The graph can then be plotted using logarithmic axes. This spreads out the plot and distinguishes different well performing systems more clearly.

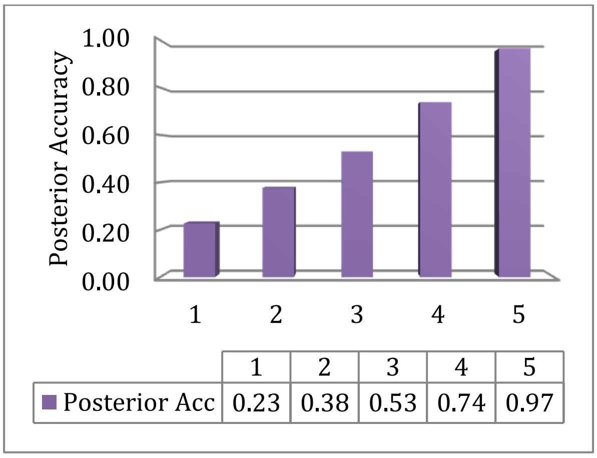
4. 2. Validation results for investigative stage

In this section, we present the results of validation of the network, when used in the investigation phase (i.e. a fingerprint examiner will search for a fingerprint in the reference dataset which can be linked to the fingermark found on the crime scene). Hence, if a fingermark containing a general pattern is recovered from the crime scene, then the Bayesian network calculates the probability for this general pattern to have been left by each finger of a random person (i.e. gives a distribution over the fingers). This distribution will allow searching the dataset per finger number, starting from the most common finger.

To perform validation we use the measures proposed in our validation framework using the posterior probability distribution. We use 20% (61221 individuals) of fingerprint dataset which was kept aside to perform the validation (testing set). This dataset, which contains information of the general pattern of all the 10 fingers, will be used to test the different finger networks.

For example if case-i in our data has a whorl on finger 1. Then we compare the most likely value of the finger or the distribution over fingers (given the general pattern of whorl) with the ground truth (i.e. finger 1).

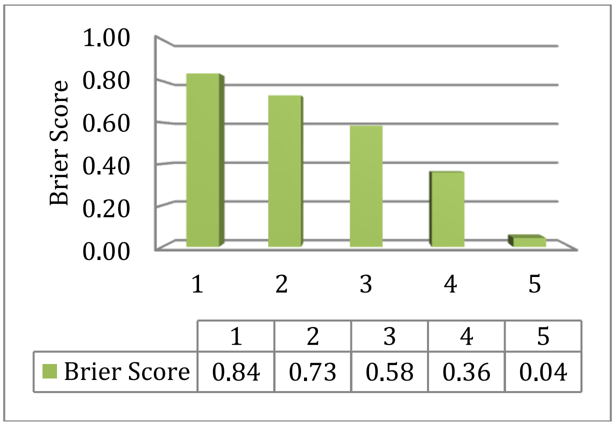
Recall that our testing data contains 61221 individuals with all the 10 fingers. Hence for the 1 Finger network, we have 61221 * 10 cases. However for the 2 Finger network, since we assume that the two fingers are consecutive, we have 8 possible finger combinations possible. Hence for the 2 Finger network, we validate using 61221 * 8 cases. Similarly for the 3 Finger network we validate using 61221 * 6 cases (since for 3 consecutive fingers there are 6 possible combinations). And finally for the 4 Finger network and the 5 Finger network, we validate using 61221 * 4 and 61221 * 2 cases respectively.



	1	2	3	4	5
🟪 Posterior Acc	0.23	0.38	0.53	0.74	0.97

Figure 4.2.1: Graph showing the posterior accuracy for 1, 2, 3, 4 and 5 finger networks.

To compute the posterior accuracy we compare the most likely finger(s) with the ground truth (finger number) in the data. We know that the higher the posterior accuracy of the network, the better the network performs in prediction. We can see from figure 4.2.1 that the posterior accuracy improves with the number of (general pattern of) fingermarks found on the crime scene i.e. more information. The 5 finger network is able to predict the correct hand from which those 5 general patterns originated correctly 97% of the time. On the other hand to compute the Brier score we compare the distribution over the finger(s) with the ground truth in the data. We know that the lower the Brier score, the better is the quality of the network. Similar to the posterior accuracy, as shown in figure 4.2.2, the Brier score improves (decreases) with the number of (general pattern of) fingermarks found on the crime scene i.e. more information. And for the 5 finger network it becomes nearly zero.



	1	2	3	4	5
🟩 Brier Score	0.84	0.73	0.58	0.36	0.04

Figure 4.2.2: Graph depicting the average Brier score for the 1, 2, 3, 4 and 5 finger networks.

4. 3. Validation results for quantifying strength of evidence

Recall that we can quantify the strength of evidence in terms of likelihood ratios (which is the ratio of the probability supposing that the evidence for the given hypothesis is true to the probability supposing that the evidence for the contrary mutually exclusive hypothesis is true). This strength of evidence can be quantified at the level of finger by computing how likely is it for the fingermark to originate from fingerX compared to the other fingers. Moreover this strength of evidence can be quantified at the level of person by computing how likely is it for the suspect to be the donor of the fingermark compared to any random person in the population.

4. 3. 1. At level of finger

At the level of finger the propositions to be tested are:

H_1 : The fingermark was left by the specific finger f_i of a random person

H_2: The fingermark was left by another finger $f_{\sim i}$ of a random person

$$LR_{H_1/H_2} = \frac{P(O \mid H_1)}{P(O \mid H_2)} = \frac{P(\, Whorl \mid Finger_4\,)}{P(\, Whorl \mid Finger_{\sim 4})} = \frac{P(\; Finger_4 \vert\, Whorl\,)/P(\, Finger_4\,)}{P(\, Finger_{\sim 4} \vert\, Whorl\,)/P(\, Finger_{\sim 4}\,)}$$

In our example, it would mean the mark originates from donor’s finger 4 vs. the mark originates from any other finger (1-3, 5-10) of the same donor. The evidential value for a whorl observed on a fingermark paired to the finger number 4 (vs. on any other finger) is 1.67. In other words, it is 1.67 times more likely to observe a whorl if it originates from the finger number 4 than if it originates from any other finger number of a random person.

This concept can be similarly extended to multiple fingers. For example if two consecutive fingermarks were found on the crime scene, then the 2 finger network will be used for inference. The propositions for this would be that

$H_{1:}$ The fingermarks were left by the specific fingers f_i and fj of a random person.

H_2: The fingermarks was left by any other finger combination of a random person.

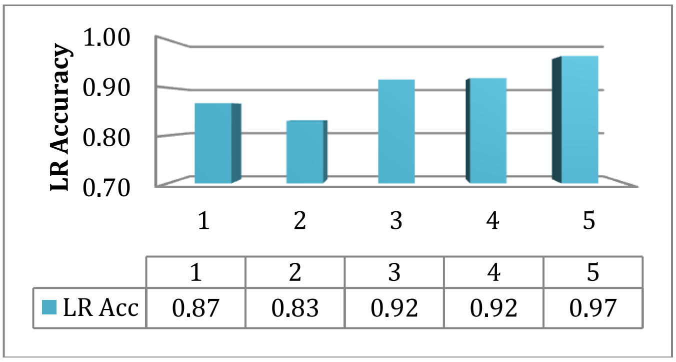
To validate these networks we use the measure proposed in our validation framework using likelihood ratios. To perform experiments using likelihood ratios, we have to assume that we know which finger the general pattern is from (in real life case- we use the 2nd level information etc; in validation- we know which finger has the general pattern from the test dataset).

Similar to the previous section we use 20% (61221 individuals) of fingerprint dataset which was kept aside to perform the validation (testing set). This dataset, which contains information of the general pattern of all the 10 fingers, will be used to test the different finger networks. Similarly, we have 61221 * 10 cases for 1 Finger network, 61221 * 8 cases for 2 Finger network etc.

Though for evaluation for likelihood ratio systems in forensic science we need to be able to compute the same source and the different source likelihood ratios. This was done under the guidance of forensic scientists at NFI. We will explain how we get the same source and different source likelihood ratios with help of an example. Let's assume that a left loop is found on finger 1. Then the same source likelihood ratio would be how likely is it to find the left loop on finger 1 compared to the rest of the fingers $\mathrm{LR}_{\mathrm{SS}} = \frac{P(Left\_Loop \mid Finger_1\,)}{P(Left\_Loop \mid Finger_{2-10})}$. In our situation, we do not have an explicit different source comaprison possible; therefore we implicitly create a different source comparison. In our example the different source comparison would be how likely is it to find a left loop on finger 2 (or other fingers) compared to a left loop on finger 1.

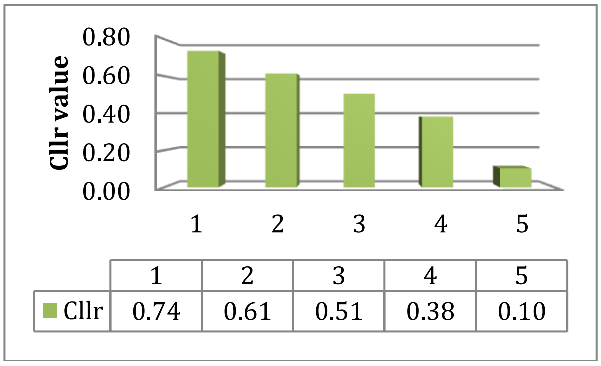
$$\mathrm{LR}_{\mathrm{DS}} = \frac{P(Left\_Loop \mid Finger_2\,)}{P(Left\_Loop \mid Finger_1)}, \frac{P(Left\_Loop \mid Finger_3\,)}{P(Left\_Loop \mid Finger_1)}, \dots \frac{P(Left\_Loop \mid Finger_{10}\,)}{P(Left\_Loop \mid Finger_1)}$$

In ideal scenarios we would get $\mathrm{LR}_{\mathrm{SS}} > 1$ and $\mathrm{LR}_{\mathrm{DS}} < 1$. The measures we presented in the framework tell us how well our network performs by plotting these numbers in different ways (described in the framework). Moreover this concept can be easily extended to multiple fingers.



	1	2	3	4	5
🟦 LR Acc	0.87	0.83	0.92	0.92	0.97

Figure 4.3.1.1: Graph depicting the LR accuracy for the 1, 2, 3, 4 and 5 finger networks.



	1	2	3	4	5
🟩 Cllr	0.74	0.61	0.51	0.38	0.10

Figure 4.3.1.2: Graph depicting the **C**$_{llr}$ values for the 1, 2, 3, 4 and 5 finger networks.

We see in Figure 4.3.1.1, that the LR accuracy (or 1-RMED) is above 80% for all the networks. Though to compute this we do not need the LR_{DS}. Also the C_{llr} value as shown in Figure 4.3.1.2, decreases as we get more evidence (general patterns). As mentioned before the lower the C_{llr} the better the network performs. A C_{llr} of 1 would indicate that the evidence gives no information if H_1 or H_2 is true. Moreover lower the C_{llr} value, more is its discriminatory power.

We also assess the performance of the LR values using tippet plot, ECE plot and DET curves. As we know the discriminatory power of the LR system is depicted in the tippet plot by the distance between the two curves. Moreover in the equal error rate decreasing as we increase the complexity of the network (i.e. find more fingermarks on the crime scene).

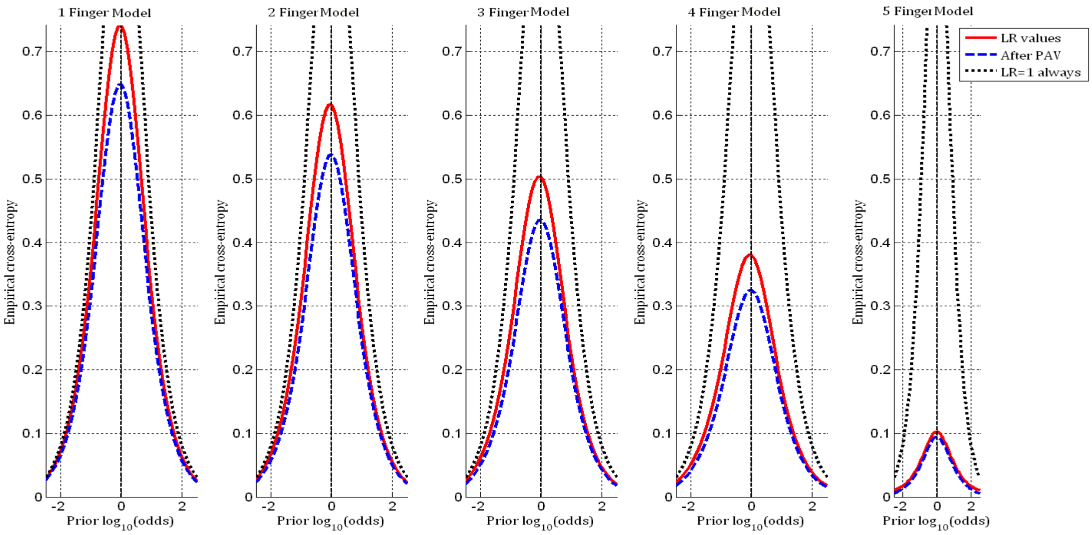
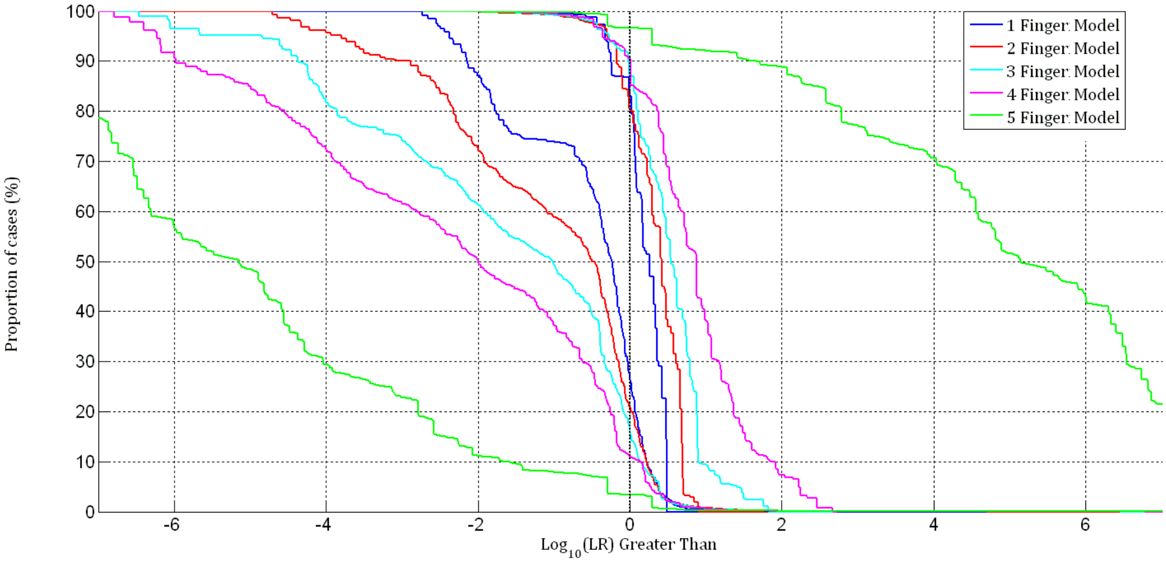


Figure 4.3.1.3: ECE plots for the 1, 2, 3, 4 and 5 finger network.



Misleading Evidence:
1 Finger Model: SS=16.5729%, DS=26.8767%
2 Finger Model: SS=19.7289%, DS=20.9041%
3 Finger Model SS=12.7925%, DS=16.3115%
4 Finger Model SS=9.4359%, DS=11.0795%
5 Finger Model SS=3.3787%, DS=3.3787%

Figure 4.3.1.4: Tippet plot for the 1, 2, 3, 4 and 5 finger network.

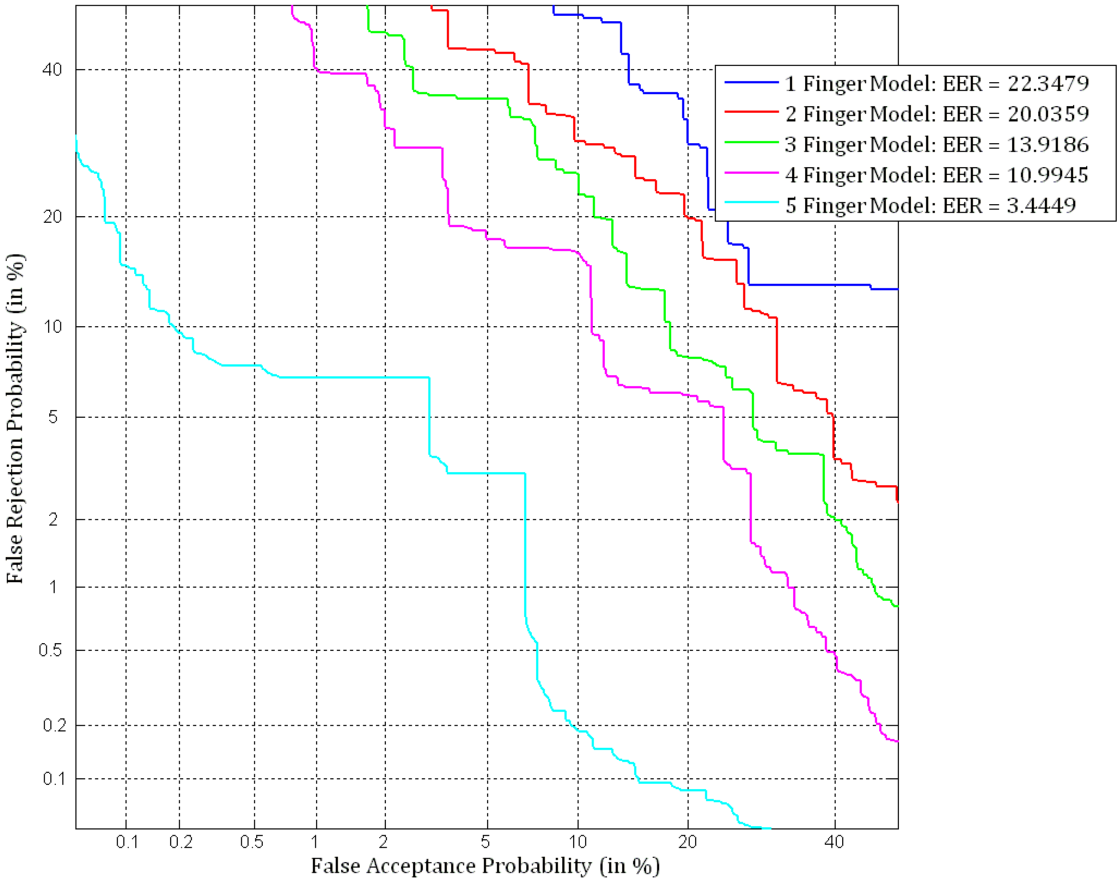


Figure 4.3.1.5: DET curve for the 1, 2, 3, 4 and 5 finger networks.

4. 3. 1. 1. Discussion

In the previous section we have presented the validation results for quantifying the strength of evidence at the level of finger for the 1, 2, 3, 4 and 5 Finger networks. Fingerprint examiners and forensic scientist would now need to assess if these results are satisfactory to be used in case-work analysis.

4. 3. 2. From level of finger to level of person

In the previous section we validated the networks which produced an evidential value at the level of the finger. This evidential value in itself is not of much help since it does not give any information for the prosecution or defence.

However these probabilities can also be used in the method proposed by [Neumann et al., 2011], to extract the evidential value of a fingermark using the first level and the second level details. Though discussion of this method is beyond the scope of the thesis.

In forensic evaluations, rarer evidence has a higher strength of evidence. In future research, this correlation can be somehow used to quantify the strength of evidence at the level of the person.

4. 3. 3. At person level

In this section, we make an attempt to quantify the strength of evidence at the level of the person. Hence according to forensic scientists the hypothesis for this is framed as follows:

H_p: The fingermark was left by the specific finger of the suspect.
H_d: The fingermark was left by any finger of any person.

In forensic science, likelihood ratios are usually constructed with the numerator being the probability of the evidence if the identified person is the source of the evidence, and the denominator being probability of the evidence if an unidentified person is the source of the evidence.

Similar to DNA analysis, in dactyloscopy, the fingerprint examiner can conclude that the suspect's finger 1 matches with the fingermark (it would be obvious that they would have the same general pattern) found on the crime scene. In this case the n-finger networks would be used to determine the probability that a randomly selected other person having the same general pattern will match the fingermark found on the crime scene (i.e. the random match probability- RMP).

Subsequently the examiner will compute the LR, defined as the ratio of the probability that the suspect's finger left the mark (H_p) versus the probability that an unknown person left the mark. Based on this assumption the fingerprint examiner will reason:

- If the suspect is the donor of the trace, then the general pattern on the fingerprint of the suspect and the fingermark match. Then the probability of finding the general pattern on the particular finger of the suspect is therefore equal to 1 if hypothesis H_p is true i.e. $P(Left\_Loop \mid Finger_1\ of\ Suspect)$ =1.
- The chance an unknown man is the donor of the fingermark on the crime scene is the random match probability of any person having a left_loop on his fingers.

Therefore the evidential value as expressed as a likelihood ration would be as follows:

$$\mathrm{LR}_{\mathrm{hp/hd}} = \frac{P(O|H_p)}{P(O|H_d)} = \frac{P(Left\_Loop \mid Finger_1\ of\ Suspect)}{P(Left\_Loop \mid Finger_{1-10})} = \frac{1}{RMP\_left\_loop}$$

This likelihood ratio can be considered as the same source hypothesis. Formulating the different source hypothesis is not so trivial and hence has been left for future research. Without the same source likelihood ratios, we can only generate on curve of the tippet plot (Figure 4.3.3.1). However without the different source results (other curve) we cannot perform validation of these networks.

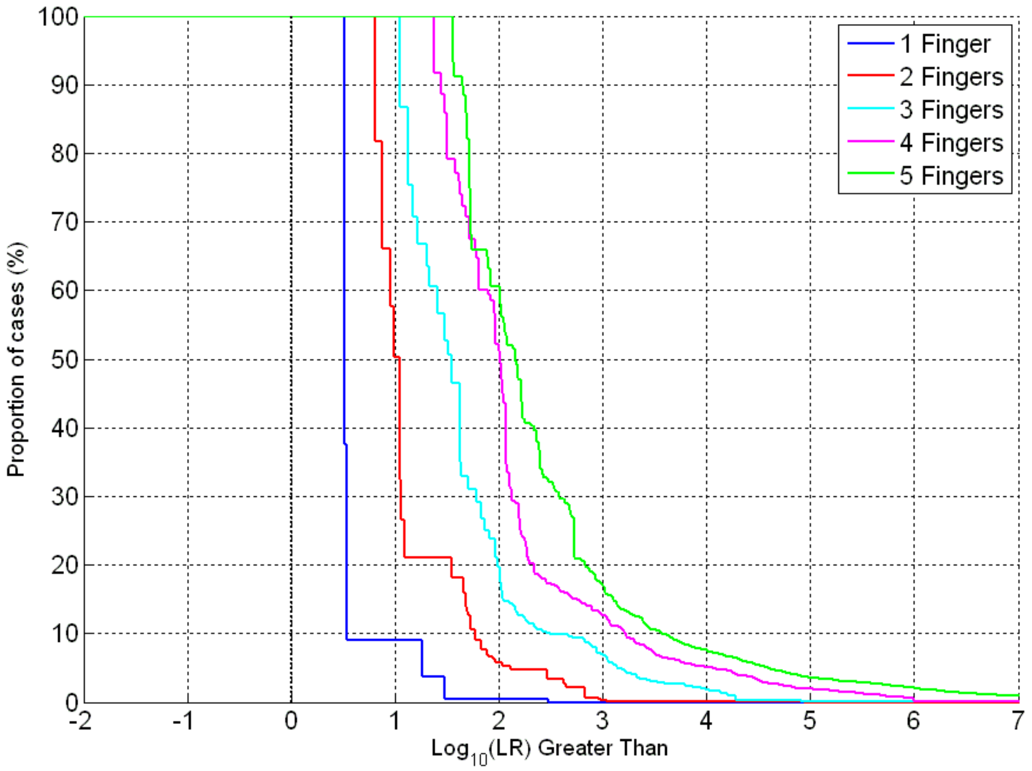


Figure 4.3.3.1: Tippet plot for the 1, 2, 3, 4 and 5 finger network

4. 4. Comparison with previously designed networks

In this section we compare the validation results for Doekhie’s 1 Finger network (Section 2.4.1) with our 1 Finger network.

Since we do not put any evidence on the *Hand* and *Gender* variable, these networks are structurally equivalent (in our project context). Though the difference is that in our network we obtain the prior distribution (equiprobable) from the fingerprint dataset whereas Doekhie’s network prior distribution on the Finger variable is obtained from the fingermark dataset (which is done to refine the equiprobable prior to a more realistic one).

The results are depicted in Figure 4.4.1 and 4.4.2. Out network performs slightly better in terms of measures using posterior probability distribution. However there is hardly any difference when we compare the performance using Likelihood ratios. In the tippet plot we can see that the different source curves are identical, whereas there is some slight change in the same source curve.

	Posterior Accuracy	Avg. Brier Score	C_{llr}
Doekhi’s network	0.17	0.88	0.74
Our network	0.22	0.84	0.74

Figure 4.4.1: Comparison of Doekhie’s 1 Finger network and Our 1 Finger Network

In future, we would like to perform a sensitivity analysis on the FingerNo variable, to see if how sensitive it is to parameter changes in our network. Therefore to know how important is it to use a more realistic prior on the FingerNo.

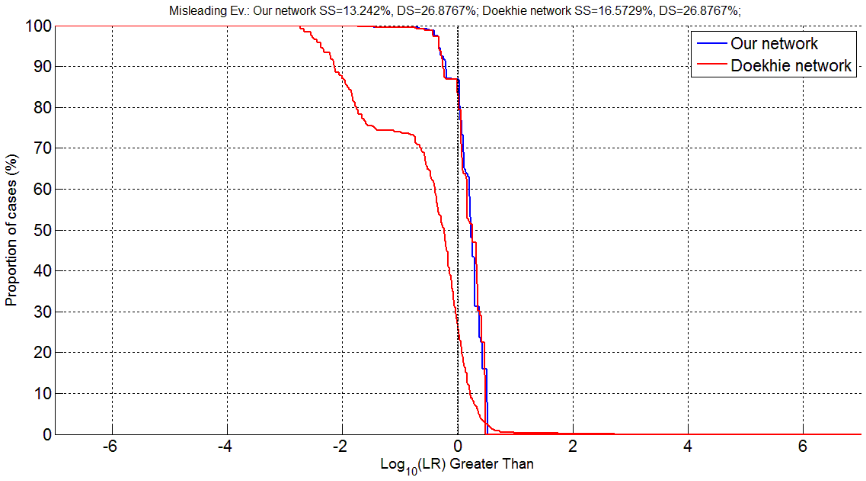


Figure 4.4.2: Tippet plot for of Doekhie's 1 Finger network and Our 1 Finger Network

4. 4. 1. Discussions

We do not compare Doekhie's 2 Finger network with our 2 Finger network, (since as mention in Section 2.4.3) she has used the prior on *FingerA* and *FingerB* from the fingermark dataset. Which we feel is not scientifically correct, since the fingermark database does not have any information about consecutivity and hence should not be used when we want to answer queries regarding consecutive patterns.

Chapter 5: Concluding remarks

5. 1. General Pattern dependence

In this section we look into the assumption used to model the N-finger networks, that the general patterns on different fingers are independent. To do that we learn a network on the general patterns for the 10 fingerprints i.e. let the network model the dependencies in the data with help of a structural learning algorithm.

After this we analyze how strong the dependencies between the fingers are by comparing the learned network to the network where the general patterns on the different fingers are independent. This process will help us in gaining more insight in the domain: i.e. if the general pattern on one finger is (in) dependent of the general pattern of another finger. If we find that the dependencies are considered important, then they can be incorporated into the finger networks (Section 3.3) at a later stage.

5. 1. 1. Learning Bayesian Networks

In data rich applications, data collections that are large and reliable enough can be used to automatically learn the structure of a probabilistic network [Buntine, 1996]. The qualitative part of the network is learnt from the data using an exact Bayesian network structure discovery algorithm [Silander, 2012] for complete discrete data i.e. data with no missing values. We employ the algorithm by [Silander & Myllymaki, 2012], because it is guaranteed to find a globally optimal Bayesian network structure. Moreover, unlike some other exact structure discovery algorithms, the source code for this algorithm is freely available on the web.

To get a better insight in the dependence between the general patterns on the different fingers, we define a variable *General_Pattern_on_FingerX* (X ∈ 1 to 10) for each finger. The *General_Pattern_on_FingerX* variables can hold values as defined in Section 2.2. The structural discovery algorithm finds the graph illustrated in Figure 5.1.1.1 as one of the networks that best fit the data.

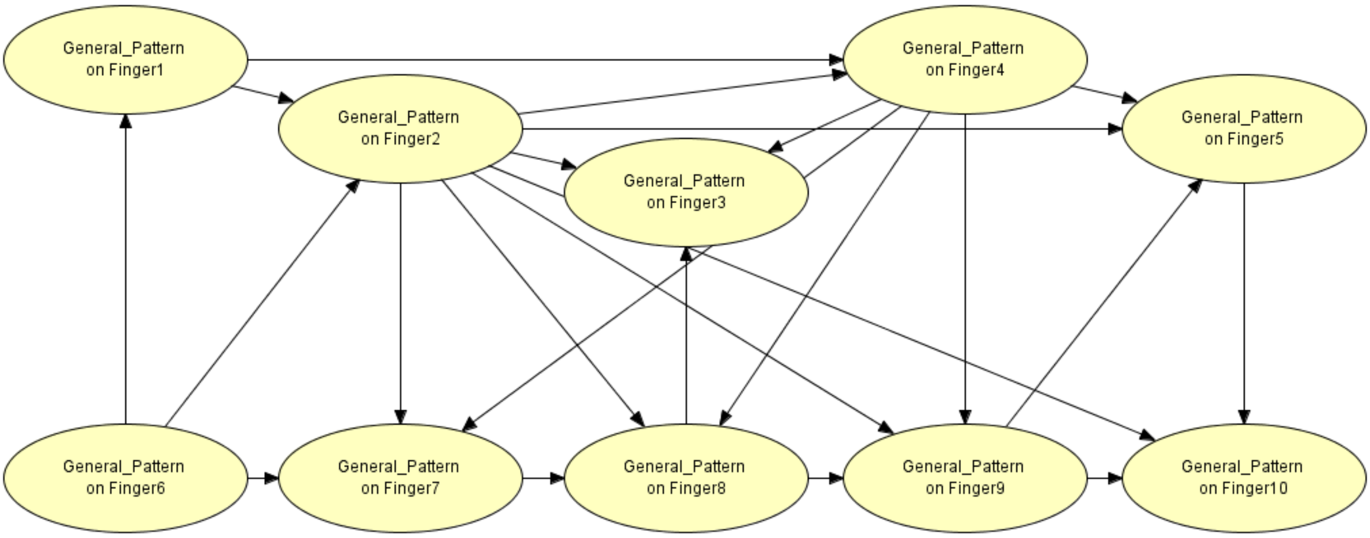

General_Pattern on Finger1
General_Pattern on Finger2
General_Pattern on Finger3
General_Pattern on Finger4
General_Pattern on Finger5
General_Pattern on Finger6
General_Pattern on Finger7
General_Pattern on Finger8
General_Pattern on Finger9
General_Pattern on Finger10

Figure 5.1.1.1: Structural discovery algorithm used to obtain the dependencies between the *General_Pattern* variables of different fingers

5. 1. 1. 1. Structural Learning

In various domains of application like forensic science, data has been collected and maintained over numerous years of every-day problem solving. Such a data collection implicitly contains highly valuable information about the relationships among the variables identified. If a comprehensive dataset is available in the domain for which a Bayesian network is to be developed, the construction of the network's qualitative part may be performed automatically: the basic idea of learning the qualitative part (digraph) from data is to distil information from the dataset and exploit it for constructing a digraph. However it's important to note that these dependencies represent statistical dependencies and not necessarily causal relations.

The total number of Bayesian networks possible with n variables can be calculated by the recursive formula given by [Robinson, 1977]. In our situation, with 10 general pattern variables around 4 * 10^{18} structures are possible. Hence it is computationally infeasible to search through all the possible networks.

In general, finding the best Bayesian network structure[3] is NP-hard [Chickering et al., 2003]. [Silander & Myllymaki, 2012] developed a Bayesian network structure discovery algorithm for complete discrete data which finds a Bayesian network structure without any structural constraints. This algorithm guarantees to give a globally optimal network for networks with less than 29 variables.

An optimal Bayesian network has a structure that is most likely given the data. For a given network it is possible to compute the probability that the network in fact generated the data. The two probabilities are related by Bayes' theorem.

$$P(G|D) = \frac{P(D|G).P(G)}{P(D)}$$

The normalizing denominator P(D) does not depend on the structure G, so provided that the numerator can be calculated, it is possible to compare the relative probabilities of Bayesian network structures without calculating the denominator. This makes it possible to search for the most probable network structure.

The search for the best Bayesian network structure is guided by a scoring function that, given a data, attaches a real number to any given network. The learning problem is now a maximization problem: the better the network, the greater the score. The nature of many common scoring functions is such that several different networks may have equal scores [Chickering, 1995], so instead of 'the best' network a 'set of best' networks can be produced.

[3] Why not use a complete structure (Fully connected): By suitably setting the parameters of a complete Bayesian network (i.e., any Bayesian network with the maximum number n(n-1)/2 of directed edges or arcs), it is possible to present any distribution, so there is never a way to tell for certain that the data did not come from a complete Bayesian network. However, a complete network structure gives little insight to the domain of interest, and from the probabilistic inference point of view, it amounts to listing the probabilities for all the possible combinations of variables. If it is assumed that the data was generated from an unknown Bayesian network, one needs to find the structure of the Bayesian network that generated the data sample [Heckerman, 1996]. Also for a complete network a huge amount of conditional probability values need to be calculated. This could lead to imprecision.

5. 1. 1. 2. Parameter Learning

After we have learnt an optimal structure for the Bayesian network, we are left with parameterizing this network. Theses parameters are assessed from the data using simple frequentist approach (i.e. counting). To perform parameter learning we used GeNIe (Graphical Network Interface) software package, which is the graphical interface to SMILE, a fully portable Bayesian inference engine developed by the Decision Systems Laboratory, University of Pittsburgh.

5. 1. 1. 3. Results

Structural learning on the *General_Pattern* nodes was performed using the algorithm of [Silander & Myllymaki, 2012]. The KLPD fingerprint dataset was divided into a training set (80%, N=244884) and a testing set (20%, N= 61221). The training set was used to learn the structure of the network as well as its parameters.

BDeu (Bayesian Dirichlet equivalence uniform) [Buntine, 1991] score is a popular scoring metric for learning Bayesian network structures for complete discrete data. It corresponds to a set of plausible assumptions under which the posterior odds of different Bayesian network structures can be calculated, thus giving us the opportunity to find the maximum posterior structure for the data. Hence maximizing the BDeu score equals maximizing the posterior probabilities of the structures. BDeu[4] with equivalent sample size 1 was used for this analysis. Equivalent sample size expresses the strength of our prior belief in the uniformity of the conditional distributions of the network.

Silander's algorithm produced 10 best networks (attached in Appendix B). These were coincidently in the same equivalent class (i.e. represented the same independencies and joint distribution). Hence these networks give the same predictive distribution for the data i.e. are equivalent for inference. There was only some variability observed in the direction of the arcs between variables *General_Pattern_on_Finger1, General_Pattern_on_Finger2, General_Pattern_on_Finger4, General_Pattern_on_Finger6*

[4] Though in future, we can use other scoring methods (e.g. information-theorectic scoring function-BIC) and compare the difference.

(though not all combinations are possible). For this reason only one of the networks was used for inference.

We now find try to find a lower bound to the probability of the set of 10 best networks we identified, given the data. We do this to see how well the networks have been identified by the learning algorithm.

Let the probability of a best network (i.e. $P(G_{best} \mid D)$) be denoted by p.
Therefore the sum of the probability of the 10 best networks is 10*p.

The best network was found to be e^{95} times more probable than the 2nd best network in any other equivalence class. If the probability of the second best network is denoted by q (i.e. $P(G_{2nd\_best} \mid D)$) then $q = (1/e^{95}) * p$

It is also known that the probability of all the networks should sum up to 1.
$10 * p + sum\_of\_other\_networks = 1$

Since we want to compute the lower bound, we assume that all the other networks have atleast the same probability as the second best network. N is the number possible structures- in our case around $4*10^{18}$.

$$\Rightarrow 10 * p + (N - 10) * q \geq 1$$

$$\Rightarrow 10 * p + (N - 10) * \left(\frac{1}{e^{95}} * p\right) \geq 1$$

$$\Rightarrow p * \left(10 + (N - 10) * \left(\frac{1}{e^{95}}\right)\right) \geq 1$$

$$\Rightarrow p \geq \frac{1}{\left(10 + (N - 10) * \left(\frac{1}{e^{95}}\right)\right)}$$

The probability of the class of best networks (10 networks)

$$\Rightarrow 10 * p \geq \frac{10}{\left(10 + (N - 10) * \left(\frac{1}{e^{95}}\right)\right)} \approx 1 \; (\textit{almost equal to } 1)$$

Hence the probability of the most probable equivalence class has a lower bound of 1. This shows these 10 networks form the most probable networks and that the learning algorithm has identified the networks pretty well.

5. 1. 2. Comparison with independent network

To check if the dependencies portrayed by the network illustrated Figure 5.1.1.1 are strong, we compare it to the network illustrated without any edges, which is a network constructed assuming all the general patterns on different fingers are independent.

We have no assumption of dependency in the 1 finger network. We make this assumption only in the consecutive finger networks. In this project, since we are interested in 2, 3, 4 and 5 consecutive finger combinations, we only compare the joint distributions of these finger combinations for the dependent (Figure 5.1.1.1) and independent network using KL divergence distance measure. For example the two consecutive finger combinations are Finger 1 and Finger 2, Finger 2 and Finger 3 etc, therefore we compare probability distribution of the general pattern on finger 1 (GP_F_1 for short) and finger 2 in the dependent network- P_{dep} (GP_F_1, GP_F_2) with the independent network-P_{ind} (GP_F_1, GP_F_2). Using the KL divergence, we measure the amount of information lost when the probability distribution of the independent network is used to approximate the probability distribution of the dependent network. The results are illustrated in the Tables 5.1.2.1, 5.1.2.3, 5.1.2.5, 5.1.2.7 for the 2, 3, 4 and 5 finger networks respectively. We observe KL divergence scores of > 0.1 and hence can conclude that there is some loss of information when the probability distribution of independent network is used to approximate the probability distribution of the dependent network.

Apart from KL divergence, we also compare the most likely general pattern combination which occurs on these consecutive fingers for both the networks shown in Tables 5.1.2.2, 5.1.2.4, 5.1.2.6, 5.1.2.8 for the 2, 3, 4 and 5 finger networks respectively.

D$_{KL}$ [P$_{dep}$ (GP_F$_1$, GP_F$_2$) || P$_{ind}$ (GP_F$_1$, GP_F$_2$)]	0.09
D$_{KL}$ [P$_{dep}$ (GP_F$_2$, GP_F$_3$) || P$_{ind}$ (GP_F$_2$, GP_F$_3$)]	0.20
D$_{KL}$ [P$_{dep}$ (GP_F$_3$, GP_F$_4$) || P$_{ind}$ (GP_F$_3$, GP_F$_4$)]	0.16
D$_{KL}$ [P$_{dep}$ (GP_F$_4$, GP_F$_5$) || P$_{ind}$ (GP_F$_4$, GP_F$_5$)]	0.13
D$_{KL}$ [P$_{dep}$ (GP_F$_7$, GP_F$_6$) || P$_{ind}$ (GP_F$_7$, GP_F$_6$)]	0.09
D$_{KL}$ [P$_{dep}$ (GP_F$_8$, GP_F$_7$) || P$_{ind}$ (GP_F$_8$, GP_F$_7$)]	0.21
D$_{KL}$ [P$_{dep}$ (GP_F$_9$, GP_F$_8$) || P$_{ind}$ (GP_F$_9$, GP_F$_8$)]	0.18
D$_{KL}$ [P$_{dep}$ (GP_F$_{10}$, GP_F$_9$) || P$_{ind}$ (GP_F$_{10}$, GP_F$_9$)]	0.13
Avg_D$_{KL}$ [P$_{dep}$ (GP_F$_i$, GP_F$_j$) || P$_{ind}$ (GP_F$_i$, GP_F$_j$)]	0.15

Table 5.1.2.1: KL divergence for 2 finger network

	Dependent Network	Independent Network
arg max P(GP_F$_1$, GP_F$_2$)	{**Whorl**, Whorl}	{**R_Loop**, Whorl}
arg max P(GP_F$_2$, GP_F$_3$)	{**R_Loop**, R_Loop}	{**Whorl**, R_Loop}
arg max P(GP_F$_3$, GP_F$_4$)	{R_Loop, **R_Loop**}	{R_Loop, **Whorl**}
arg max P(GP_F$_4$, GP_F$_5$)	{**R_Loop**, R_Loop}	{ **Whorl**, R_Loop }
arg max P(GP_F$_7$, GP_F$_6$)	{L_Loop, L_Loop}	{L_Loop, L_Loop}
arg max P(GP_F$_8$, GP_F$_7$)	{L_Loop, L_Loop}	{L_Loop, L_Loop}
arg max P(GP_F$_9$, GP_F$_8$)	{L_Loop, L_Loop}	{L_Loop, L_Loop}
arg max P(GP_F$_{10}$, GP_F$_9$)	{L_Loop, L_Loop}	{L_Loop, L_Loop}

Table 5.1.2.2: Most likely general pattern combinations on different finger combinations for 2 finger network for dependent and independent network.

D$_{KL}$ [P$_{dep}$ (GP_F$_1$, GP_F$_2$, GP_F$_3$) || P$_{ind}$ (GP_F$_1$, GP_F$_2$, GP_F$_3$)]	0.29
D$_{KL}$ [P$_{dep}$ (GP_F$_2$, GP_F$_3$, GP_F$_4$) || P$_{ind}$ (GP_F$_2$, GP_F$_3$, GP_F$_4$)]	0.40
D$_{KL}$ [P$_{dep}$ (GP_F$_3$, GP_F$_4$, GP_F$_5$) || P$_{ind}$ (GP_F$_3$, GP_F$_4$, GP_F$_5$)]	0.29
D$_{KL}$ [P$_{dep}$ (GP_F$_8$, GP_F$_7$, GP_F$_6$) || P$_{ind}$ (GP_F$_8$, GP_F$_7$, GP_F$_6$)]	0.30
D$_{KL}$ [P$_{dep}$ (GP_F$_9$, GP_F$_8$, GP_F$_7$) || P$_{ind}$ (GP_F$_9$, GP_F$_8$, GP_F$_7$)]	0.41
D$_{KL}$ [P$_{dep}$ (GP_F$_{10}$, GP_F$_9$, GP_F$_8$) || P$_{ind}$ (GP_F$_{10}$, GP_F$_9$, GP_F$_8$)]	0.32
Avg_D$_{KL}$ [P$_{dep}$ (GP_F$_i$, GP_F$_j$, GP_F$_k$) || P$_{ind}$ (GP_F$_i$, GP_F$_j$, GP_F$_k$)]	0.34

Table 5.1.2.3: KL divergence for 3 finger network

	Dependent Network	Independent Network
arg max P(GP_F$_1$, GP_F$_2$, GP_F$_3$)	{R_Loop, **R_Loop**, R_Loop}	{R_Loop, **Whorl**, R_Loop }
arg max P(GP_F$_2$, GP_F$_3$, GP_F$_4$)	{**R_Loop**, R_Loop,**R_Loop**}	{**Whorl**, R_Loop, **Whorl** }
arg max P(GP_F$_3$, GP_F$_4$, GP_F$_5$)	{R_Loop, **R_Loop**, R_Loop}	{R_Loop, **Whorl**, R_Loop }
arg max P(GP_F$_8$, GP_F$_7$, GP_F$_6$)	{L_Loop, L_Loop, L_Loop }	{L_Loop, L_Loop, L_Loop }
arg max P(GP_F$_9$, GP_F$_8$, GP_F$_7$)	{L_Loop, L_Loop, L_Loop }	{L_Loop, L_Loop, L_Loop }
arg max P(GP_F$_{10}$, GP_F$_9$, GP_F$_8$)	{L_Loop, L_Loop, L_Loop }	{L_Loop, L_Loop, L_Loop }

Table 5.1.2.4: Most likely general pattern combinations on different finger combinations for 3 finger network for dependent and independent network.

$D_{KL}\,[\,P_{dep}\,(GP\_F_1, GP\_F_2, GP\_F_3, GP\_F_4) \Vert P_{ind}\,(GP\_F_1, GP\_F_2, GP\_F_3, GP\_F_4)\,]$	0.51
$D_{KL}\,[\,P_{dep}\,(GP\_F_2, GP\_F_3, GP\_F_4, GP\_F_5) \Vert P_{ind}\,(GP\_F_2, GP\_F_3, GP\_F_4, GP\_F_5)\,]$	0.56
$D_{KL}\,[\,P_{dep}\,(GP\_F_9, GP\_F_8, GP\_F_7, GP\_F_6) \Vert P_{ind}\,(GP\_F_9, GP\_F_8, GP\_F_7, GP\_F_6)\,]$	0.51
$D_{KL}\,[\,P_{dep}\,(GP\_F_{10}, GP\_F_9, GP\_F_8, GP\_F_7) \Vert P_{ind}\,(GP\_F_{10}, GP\_F_9, GP\_F_8, GP\_F_7)\,]$	0.55
$Avg\_D_{KL}\,[\,P_{dep}\,(GP\_F_i, GP\_F_j, GP\_F_k, GP\_F_l) \Vert P_{ind}\,(GP\_F_i, GP\_F_j, GP\_F_k, GP\_F_l)\,]$	0.53

Table 5.1.2.5: KL divergence for 4 finger network

	Dependent Network	Independent Network
arg max P(GP_F_1, GP_F_2, GP_F_3, GP_F_4)	{R_Loop, **R_Loop**, R_Loop, **R_Loop** }	{R_Loop, **Whorl**, R_Loop, **Whorl** }
arg max P(GP_F_2, GP_F_3, GP_F_4, GP_F_5)	{**R_Loop**, R_Loop, **R_Loop**, R_Loop }	{**Whorl**, R_Loop, **Whorl**, R_Loop }
arg max P(GP_F_9, GP_F_8, GP_F_7, GP_F_6)	{L_Loop, L_Loop, L_Loop, L_Loop }	{L_Loop, L_Loop, L_Loop, L_Loop }
arg max P(GP_F_{10}, GP_F_9, GP_F_8, GP_F_7)	{L_Loop, L_Loop, L_Loop, L_Loop }	{L_Loop, L_Loop, L_Loop, L_Loop }

Table 5.1.2.6: Most likely general pattern combinations on different finger combinations for 4 finger network for dependent and independent network.

$D_{KL}\,[\,P_{dep}\,(GP\_F_1, GP\_F_2, GP\_F_3, GP\_F_4, GP\_F_5) \Vert P_{ind}\,(GP\_F_1, GP\_F_2, GP\_F_3, GP\_F_4, GP\_F_5)\,]$	0.67
$D_{KL}\,[\,P_{dep}\,(GP\_F_{10}, GP\_F_9, GP\_F_8, GP\_F_7, GP\_F_6) \Vert P_{ind}\,(GP\_F_{10}, GP\_F_9, GP\_F_8, GP\_F_7, GP\_F_6)\,]$	0.64
$Avg\_D_{KL}\,[\,P_{dep}\,(GP\_F_i, GP\_F_j, GP\_F_k, GP\_F_l, GP\_F_m) \Vert P_{ind}\,(GP\_F_i, GP\_F_j, GP\_F_k, GP\_F_l, GP\_F_m)\,]$	0.66

Table 5.1.2.7: KL divergence for 5 finger network

	Dependent Network	Independent Network
arg max P(GP_F_1, GP_F_2, GP_F_3, GP_F_4, GP_F_5)	{R_Loop, **R_Loop**, R_Loop, **R_Loop**, R_Loop}	{R_Loop, **Whorl**, R_Loop, **Whorl**, R_Loop}
arg max P(GP_F_{10}, GP_F_9, GP_F_8, GP_F_7, GP_F_6)	{L_Loop, L_Loop, L_Loop, L_Loop, L_Loop }	{L_Loop, L_Loop, L_Loop, L_Loop, L_Loop}

Table 5.1.2.8: Most likely general pattern combinations on different finger combinations for 8 finger network for dependent and independent network.

5. 1. 3. Discussion

There is some difference between the most likely general pattern as classified by the dependent network compared to the independent one. In addition we do not get a low (near zero) KL divergence score for all the multiple finger networks. This means that the assumption of fingerprint examiners that the dependencies are not important has to be reconsidered. Though similar to the gender analysis (in Section 3.2.2.3), these dependencies between the general patterns may not influence the finger combination distribution. For this to be done, we will have to somehow learn the dependencies between the general pattern variables in the Bayesian networks we created in Section 3.3.3. After which a similar study as Section 3.2.2.3 needs to be performed to see the impact of the dependencies of the general patterns on the finger combination distribution. We leave this for future work.

5. 2. Conclusion

In this part of the thesis we extended the one finger network to multiple finger networks, making the assumption that the general pattern on different fingers are independent given the fingers the occur on. These networks have two applications for forensic scientists or fingerprint examiners. Firstly, (in investigative stage) it will help to reduce the search space, when searching for fingerprints which can be matched to the fingermark. Secondly, (in evaluation stage) it will help quantify the strength of evidence at the level of finger and level of person.

We have presented validation results for the investigative stage and evaluation stage at the level of finger. However validation of quantifying the strength of evidence at the level of person is left for future research. Moreover, we also need to interact with forensic scientists and fingerprint examiners to somehow understand if the networks we have created are 'good enough' (though there is no clear definition for 'good enough') to be used in practise.

In future, we need to determine how 'sensitive' the network is to changes in the values of the parameter. Especially for the *Finger_No* variable, since it will give us an indication if a more refined prior on the *Finger_No* (i.e. from the fingermark dataset) is useful or not? Moreover if it is useful, it will give motivation to the fingerprint examiners to store information about consecutivity in the fingermark database.

We also saw that the general pattern on different fingers are not independent. Therefore in future we need to learn Bayesian networks from the data (like structural learning in 5.1) and evaluate if the dependencies between the general patterns have an affect on the distribution over finger. In other words, in forensic science terminology, we need to perform a cross validation- i.e. compare the performance of networks created with help of experts (and data) to the networks created from data only.

And finally, we also need to make an attempt to solve an open question in forensic science, 'how to combine the strength of evidence of different forensic evidence- mainly focussing on the level 1 and level 2 details of fingerprints'.

Part II:
Computer Science

MAP Computation
in
Bayesian Networks

Chapter 6:

Introducing MAP

6. 1. Abductive reasoning in Bayesian networks

Since the term Bayesian Network was coined by Judea Pearl [Pearl, 1985] in the mid nineteen-eighties, a lot of attention has been to given to probabilistic inference problems. This has led to a lot of exact and approximate inference algorithms being developed to obtain the posterior probabilities (also known as posterior marginals) of variables given evidence.

More recently, though a more difficult problem known as *abductive inference* has been gaining equal (if not more) attention [Kwisthout, 2011]. Abduction is defined as the process of generating a plausible explanation for a given set of observations or facts [Pople, 1973]. In the context of probabilistic reasoning, abductive inference corresponds to finding the maximum a posteriori probability state of the system variables, given some evidence (observed variables) i.e. the configuration of a set of hypothesis or query variables that is most likely or best explains the evidence [Gamez & Moral, 2004; Neapolitan, 2004].

[Cooper, 1990] had shown that the problem of probabilistic inference in general is NP-hard i.e. from just computing the posterior marginal to a more complex problem of abductive inference becomes intractable even for problems with a small number of variables. Though generally in real-life scenarios, when the tree-width[5] is bounded, just computing the posterior probability is not that computationally hard [Kwisthout et al., 2010]. However it still remains NP-hard to perform abductive inference.

In the last few years, artificial intelligence researchers have devoted increasing attention to the development of abductive reasoning methods in a wide range of applications. Probably the most clear application of abductive reasoning is in the field of diagnosis (see e.g. [Peng & Reggia, 1987; Reiter, 1987; Lucas et al., 2000; van der Gaag et

[5] In graph theory, the treewidth of an undirected graph is a number associated with the graph, which can be defined in several equivalent ways, one of them being that it is the size of the largest clique in a chordal completion of the graph.

al., 2002; Geenen et al., 2006]) where the most likely diagnosis is to be found, given (some) clinical observations and test results. Although other applications exist, such as in weather forecasting, to predict precipitation based on meteorological evidence [Cofino et al., 2002] and in channel coding where messages transmitted over a noisy channel are to be decoded [Frey, 1997]. In the latter case, the receiver observes a sequence of bits received over the channel, and then attempts to find the most likely assignment of input bits that could have generated this observation (taking into account the model of the channel noise). This type of query is much better viewed as an abductive inference query than a standard inference query (i.e. computing the posterior marginal), since the most likely message is more interesting that the most likely values for the individual bits. A similar phenomenon arises in speech recognition [Geoffrey & Russell, 1998], where the most likely utterance given the noisy acoustic channel is more important that the most likely value of individual phonemes uttered. Abductive inference has also gained popularity in cognitive task such as vision or goal inference [Kumar and Desai, 1996; Yuille & Kersten, 2006; Baker et al., 2009], legal reasoning [Thagard, 1989] and computational models of economic process [Gamela, 2001; Demirer et al., 2006]. Though superficially these problems appear to be very different, they solve the same underlying computational problem, which is, given a Bayesian network and a set of evidence, to find the most probable joint value assignment to a set of variables.

Bayesian networks contain information needed to answer any query about the distribution. Inference is the process of answering such queries. In the next section we introduce the three types of queries: *probability query*, *most probable explanation* and *maximum a posteriori hypothesis*. The latter will be used in the following two chapters, where we discuss the two heuristics we have designed to compute the approximate maximum a posteriori hypothesis.

6. 2. Formal definition of queries

Bayesian networks represent a joint probability distribution; which is used to answer any query about the distribution. Subsequently inference is the process of using a joint probability distribution over multiple random variables to answer queries, namely: probabilistic queries, most probable explanation queries and maximum a posteriori probability queries. In this section we will look into these three queries in detail.

In the remainder of this section we consider a Bayesian network $B = (G, P)$ [6] with acyclic digraph $G = (\mathbf{V}, \mathbf{A})$. In addition, we assume that the set of variables $\mathbf{V}$ is divided into a non-empty set of evidence variables $\mathbf{X} \subset \mathbf{V}$, a non-empty set of query variables $\mathbf{Q} = \{ Q_1, Q_2, \dots Q_n \} \subseteq \mathbf{V} \setminus \mathbf{X}$, and possibly a set of remaining variables $\mathbf{Z} = \mathbf{V} \setminus (\mathbf{Q} \cup \mathbf{X})$.

6. 2. 1. Probability Query

Perhaps the most common query type is the probability query. Where the goal is to compute $P(Q_i \mid \mathbf{X} = \mathbf{x})$, which is the posterior probability distribution over the values q_i of Q_i, conditioned on the fact that $\mathbf{X}$ takes on the value $\mathbf{x}$. This is the most standard type of query, which standard inference algorithms are designed to answer. Compared to the other two query types, it is relatively easier to compute this query.

[6] In this thesis we use a standard notation: Variables are denoted by upper-case letters (A) and their values by lower case letters (a). Sets of variables are denoted by bold face upper case letters (**A**) and their instantiations are denoted by bold face lower case letters (**a**).

6. 2. 2. Most Probable Explanation

A second type of query is called the Most Probable Explanation- MPE (Also known as MAP in some literature [Koller & Friedman, 2009]). The MPE in a Bayesian Network is a complete variable assignment which has the highest probability given current evidence [Pearl, 1988; Park & Darwiche, 2004]. In simpler words, the MPE is the most likely assignment to a set of variables given complete evidence about the complement of that set. The difference of MPE from probability query is that instead of a probability we get the most likely value for all remaining variables.

More precisely, if $\mathbf{V} = \mathbf{Q} \cup \mathbf{X}$, then the most likely assignment to the variables $\mathbf{Q}$ given evidence $\mathbf{X} = \mathbf{x}$, is given by

$$\text{MPE}(\mathbf{Q} \mid \mathbf{x}) = \arg\max_q P(\mathbf{q} \mid \mathbf{x}) = \arg\max_q P(\mathbf{q}, \mathbf{x}) / P(\mathbf{x}) = \arg\max_q P(\mathbf{q}, \mathbf{x})$$

where in general $\arg\max_x f(x)$ equals a value of x which maximizes *f(x)*. It’s to be kept in mind that there might be more than one assignment that has the highest probability, in which case MPE returns either a set of possible assignments or an arbitrary member from the set.

Unfortunately, computing the MPE is in general NP-hard [Bodlaender et al., 2002; Kwisthout, 2011], and remains NP-hard when the most probable explanation is to be approximated rather than exactly computed. [De Waal & van der Gaag, 2007] showed that the MPE can be solved in polynomial time if the subgraph of the evidence variables has bounded treewidth and the number of query variables is restricted (see their Theorem 1). They observed that for most applications the number of query variables indeed is much smaller than the number of evidence variables. The number of query variables can in fact often be considered constant in terms of the number of evidence variables. For example calculating MPE for naive Bayesian classifiers and TAN classifiers (which have very small treewidth) can be performed in polynomial time.

6. 2. 3. Maximum a posteriori query

Often we are only interested in finding the most likely assignment to a set of variables, given evidence for only a subset of the complementary set. For example, in medical diagnosis we may not have evidence for all the possible symptom variables. In this case, we are not interested in the most likely assignment to both the disease and the unobserved symptom variables (which is the MPE), but rather in the most likely assignment to the disease variable(s) only. This problem is the main motivation to look further into a more general query type, known as the Maximum a posteriori query- MAP (also known as partial or marginal MAP in some literature [Koller & Friedman, 2009]).

The MAP problem is very similar to the MPE problem, except that the set **Z** of non-evidence, non-query variables is assumed to be non-empty. MPE is therefore a special case of MAP with **Z** = ∅. More formally, the most likely assignment to the variables **Q** given evidence **X** = **x**, is given by

$$\text{MAP}\ (\ \mathbf{Q}\ |\ \mathbf{x}\ ) = \text{arg max}_{q}\ P(\ \mathbf{q}\ ,\ \mathbf{x}\ )$$

Computing MAP is a significantly more difficult problem than computing MPE or the query problem [Park & Darwiche, 2004]. All of these problems are NP-Hard, including their approximations [Cooper, 1990; Paul & Luby, 1993], but the computational resources needed to solve MAP using state-of-the-art algorithms are much greater than those needed to compute the MPE.

6. 3. Existing approaches to computing MAP

In this thesis we will be focussing on computing the MAP hypothesis. We know that in general computing the MAP is NP-hard and remains NP-hard for its approximations also. Though complexity results by [Park & Darwiche, 2004] have shown it to lie in a more harder complexity class[7] NP^{PP}.

MAP consists of two problems that are hard in general- optimization and inference [Park & Darwiche, 2004]. A MAP approximation algorithm can be produced by substituting approximate versions of either the optimization or inference component (or both). The optimization problem is defined over the MAP variables, and the score for each solution candidate instantiation **q** of the MAP variables is the (possibly approximate) probability P(**q**, **x**) produced by the inference method. This allows solutions tailored to the specific problem. For networks whose treewidth is manageable, but contains a hard optimization component (e.g. polytree structure), exact structural inference can be used, coupled with an approximate optimization algorithm. Alternatively, if both components are hard, both the optimization and inference components need to be approximated.

In this section we review a few of the existing approaches to computing MAP. We first discuss algorithms to compute the exact MAP hypothesis. Followed by algorithms which approximate MAP hypothesis.

[7] Defining the complexity class NP^{PP} is outside the scope of this thesis, but it is generally considered very hard, since it is known to contain the entire polynomial hierarchy, of which NP is only the first level.

6. 3. 1. Exact MAP

Variable Elimination: We can solve exact MAP and MPE using a variable elimination algorithm [Zhang & Poole, 1996; Dechter, 1998]. Although any elimination order can be used to compute MPE. Unfortunately only a subset of these orders can be used to compute MAP. A good elimination algorithm for MAP needs to eliminate the non-MAP variables first- which could lead to a significantly larger width. The complexity of a variable elimination algorithm is exponential in the width of the used elimination order. The width of an elimination order with respect to a network is defined as the size of the maximal clique minus one, in join tree constructed based on the elimination order. It can also be equivalently defined as the number of variables minus one, in the largest table constructed when running variable elimination using the order.

Solving MPE using variable elimination is exponential in its treewidth. Whereas solving MAP using variable elimination is exponential in its constrained treewidth- which is the width of the best constrained elimination order.

Other algorithms include solving MAP using systematic search [Park & Darwiche, 2003] and by searching on compiled arithmetic circuits [Huang et al, 2006].

6. 3. 2. Approximating MAP

Since exact MAP computation is often intractable, approximation techniques are needed. We discuss a few approximate techniques below.

6. 3. 2. 1. Inference is easy

In this section we assume that performing exact inference is tractable.

Combination of most likely values: In this crude approximation of MAP, individually the most likely assignment P(Q_i **| X=x**) for all the query variables Q_i are computed. And these individual most likely assignments are considered to be the MAP assignment.

However, the assignment where each variable individually picks its most likely value can be quite different from the most likely joint assignment to all query variables simultaneously [Koller & Friedman, 2009]. The phenomenon can occur in the simplest case, where we have no evidence. For example, consider a 2 node network ($Q_1 \rightarrow Q_2$), where Q_1 and Q_2 are both binary, with conditional probability tables as shown in Table 6.3.1.1. We can see that MAP (Q_1, Q_2 | Ø) = (q_1^0, q_2^1) with a probability of 0.36. However if we apply this heuristic, then the most likely value for Q_1: arg $\max_{q1}$ P(q_1) = (q_1^1) and the most likely value for Q_2: arg $\max_{q2}$ P(q_2) = (q_2^1). Thus we have that
arg $\max_{q1.q2}$ (q_1, q_2) ≠ (arg $\max_{q1}$ (q_1), arg $\max_{q2}$ (q_2)).

The complexity of computing the MAP using this heuristic is the same as the complexity of standard inference **O**(Probability_Query_Inference) or **O**(P_Query)**.** This is simply computing the posterior marginals for all the query variables, which in general is NP-

P(Q_1 **)**	
q_1^0	0.4
q_1^1	0.6

P(Q_2 | Q_1 **)**		
	q_1^0	q_1^1
q_2^0	0.1	0.5
q_2^1	0.9	0.5

P (Q_1, Q_2 **)**		
	q_1^0	q_1^1
q_2^0	0.04	0.3
q_2^1	**0.36**	0.3

P(Q_2 **)**	
q_2^0	0.34
q_2^1	0.66

Table 6.3.1.1: Conditional probability and joint probability tables

hard.

Approximate with MPE: Another heuristic is to approximate this problem using MPE, which is done by finding the most likely configuration of every unknown variable (including the query variables and the non-evidence, non-query variables). And then MAP is approximated by using projection of the MPE assignment on the query set.

It easy to see, that this projection is not usually the most likely configuration of the query variables [Park & Darwiche, 2004]. In other words it is trivial to see that the assignment MAP($\mathbf{Q_y}$ | $\mathbf{x}$) might be completely different from the assignment of $\mathbf{Q_y}$ in MPE({$\mathbf{Q_y}$,$\mathbf{Q_z}$} | $\mathbf{x}$). For example, we again consider the 2 node network ($Q_1 \rightarrow Q_2$) introduced in the previous section. We can see that MAP(Q_1 | ∅) = q_1^1, whereas MPE(Q_1, Q_2 | ∅)= (q_1^0, q_2^1). Hence the assignment MAP(Q_1 | x) might be completely different from the assignment of Q_1 in MPE({Q_1,Q_2} | x). Thus in general the MPE query can't be used to give a correct MAP answer. In other words it may also be said, that MPE is subset of the MAP problem i.e. all MPE problems can be classified as a MAP problem, but not vice-versa.

Other algorithms include genetic algorithm by [de Campos et al. 1999], hill climbing & taboo search [Park & Darwiche 2001], simulated annealing [Yuan et al. 2004] etc.

6. 3. 2. 2. Inference is hard

The algorithms discussed so far depend on the ability to perform exact inference. However in some situations, even performing inference is intractable. In these cases, approximate inference can be substituted in order to produce MAP approximations. **Iterative belief propagation** is a useful approximate inference algorithm, which can be used for approximating MAP. The belief propagation allows all of the techniques for approximating MAP for inference tractable (Section 6.3.2.1) networks to be applied approximately when inference is not tractable.

6. 3. 3. Motivation

This leads to the main motivation behind this part of the thesis. In the following chapters we introduce two heuristics which approximates MAP. These heuristics are computationally tractable, when probability query inference is tractable. Also these heuristics do not have any structural constraints (like bounded tree-width) under which it will run efficiently. We also perform an experimental analysis to investigate the difference in performance of our heuristics and the combination of most likely value heuristic.

Though the problem will still remain in the NP-hard category, more detailed complexity analysis of our heuristic (which approximates MAP) needs to be performed to see if it still lies in the complexity class NP^{PP}.

Chapter 7:

Heuristic I for approximating MAP

In this chapter, we introduce a heuristic for MAP computations involving binary query variables Q_i whose values are denoted by q_i^0 and q_i^1, respectively; the non-query variables (**X** and **Z**) can take on any number of values. Recall that, we use *B* to denote a Bayesian network with variables $\mathbf{V} = \mathbf{X} \cup \mathbf{Q} \cup \mathbf{Z}$, consisting of

- A set of evidence variables **X.**
- A non-empty set of query variables $\mathbf{Q} = \{Q_1, Q_2, \ldots Q_n\}$.
- A set of non evidence and non query variables **Z.**

7. 1. Algorithm

Figure 7.1.1 describes the heuristic we designed to compute the MAP assignment.

```
Heuristic_Algorithm Greedy_MAP_I ( Bayesian Net $\mathbf{B}$, Query Set $\mathbf{Q}$, Evidence $\mathbf{x}$ ):
MAP $\mathbf{m}$, prob_m

$\mathbf{m}$ := true          // MAP assignment
prob_m := 1.0      // contains the probability of the MAP assignment

While $\mathbf{Q} \neq \emptyset$ do
    Compute $P(Q_i \mid \mathbf{x})$ from B, for each $Q_i \in \mathbf{Q}$        // Perform inference on network
    Compute $Q^\top = \arg\max_i \mid P(q_i^0 \mid x) - P(q_i^1 \mid x) \mid$   // find the variable $Q^\top$ which has
                                             // the maximum absolute difference between its
                                             // most likely value and least likely value

    $\mathbf{m} := \mathbf{m} \wedge q^\top$ where $q^\top$ is most likely value of $Q^\top$ // add $q^\top$ to MAP
    prob_m := prob_m * $P(q^\top \mid \mathbf{x})$

    $\mathbf{Q} := \mathbf{Q} \setminus Q^\top$
    $\mathbf{x} := \mathbf{x} \wedge q^\top$
End
```

Figure 7.1.1: Pseudo code for Heuristic I

We find the variable $Q^\top \in \mathbf{Q}$, which has the maximum absolute difference between its most likely value and least likely value (given the current evidence set $\mathbf{x}$). We make a guess that $q^\top$ which is most likely value of $Q^\top$ will be part of the MAP assignment. Since we have made an estimated guess for the variable $Q^\top$, we remove it from the query set and add its most likely value $q^\top$ to the evidence. We continue doing this till the query set is empty. Therefore at each step we are removing one variable from the query set and adding it's most likely value to the MAP and evidence.

For each iteration of the while loop, we need to perform inference on the Bayesian network only once to compute $P(Q_i \mid \mathbf{x})\ \forall Q_i \in \mathbf{Q}$. Therefore since the while loop will run n-times for n query variables, we would need to perform inference only n-times.

In our heuristic if the maximum absolute difference between its most likely value and least likely value of two variables is the same, then we choose the most likely value as a MAP assignment for any one of the variables.

Our heuristic is designed for only binary query variables. For non-binary query variables, just finding the variable $Q^\top$ which has the maximum difference between its most likely value and least likely value may not be of much use. Since these variables have more values and hence we would need to devise a different strategy to decide which variable and its corresponding value to choose. This does not seem to be trivial and hence is left for future research.

7. 2. Intuition behind the heuristic

Given the evidence ($\mathbf{X} = \mathbf{x}$), we evaluate the posterior probability of each query variable Q_i independently i.e. $P(Q_i \mid \mathbf{x})$ where $Q_i \in \mathbf{Q}$. Ignoring the interaction between the query variables (Q_i and Q_j where $i \neq j$), we find out which value of the query variable has the maximum posterior probability or in other words, the maximum confidence in its assignment. For example, if $P(Q_i = q_i^0 \mid \mathbf{x}) = 0.5$ and $P(Q_i = q_i^1 \mid \mathbf{x}) = 0.5$, we have the least possible confidence in the assignment for Q_i, whereas when $P(Q_i = q_i^0 \mid \mathbf{x}) = 0.1$ and $P(Q_i = q_i^1 \mid \mathbf{x}) = 0.9$, we have comparatively more confidence in the posterior probability distribution of Q_i. Therefore of all the query variables we choose the one that has maximum confidence in its assignment, with a belief that this will be the correct MAP assignment for that variable.

We will now mathematically demonstrate why at each step we choose $q^\top$ (which is the most likely value of $Q^\top$) as a MAP assignment for the variable $Q^\top$. Let us consider two query variables A (whose values are denoted by $a^\top$ and $a_\perp$) and B (whose values are denoted by $b^\top$ and $b_\perp$), and evidence $\mathbf{X} = \mathbf{x}$ such that

$$\begin{aligned} P(a^\top \mid \mathbf{x}) &> P(a_\perp \mid \mathbf{x}) \\ P(b^\top \mid \mathbf{x}) &> P(b_\perp \mid \mathbf{x}) \end{aligned} \quad \text{... (i)}$$

In other words, we assume $a^\top$ and $b^\top$ are the most likely values and $a_\perp$ and $b_\perp$ are the least likely values of A and B respectively given the evidence $\mathbf{X} = \mathbf{x}$.

From the definition of probability distribution, we know

$$P(a^\top \mid \mathbf{x}) + P(a_\perp \mid \mathbf{x}) = 1,$$
$$P(b^\top \mid \mathbf{x}) + P(b_\perp \mid \mathbf{x}) = 1.$$

If we assume that, $P(a^\top \mid \mathbf{x}) - P(a_\perp \mid \mathbf{x}) > P(b^\top \mid \mathbf{x}) - P(b_\perp \mid \mathbf{x})$ then

$\Leftrightarrow 1 - P(a_\perp \mid \mathbf{x}) - P(a_\perp \mid \mathbf{x}) > 1 - P(b_\perp \mid \mathbf{x}) - P(b_\perp \mid \mathbf{x})$

[Substituting $P(a^\top \mid \mathbf{x})$ by $P(a_\perp \mid \mathbf{x})$ and $P(b^\top \mid \mathbf{x})$ by $P(b_\perp \mid \mathbf{x})$]

$\Leftrightarrow P(a_\perp \mid \mathbf{x}) < P(b_\perp \mid \mathbf{x})$ ... (ii)

In a similar way using the other substitution we can arrive at

$\Leftrightarrow P(a^\top \mid \mathbf{x}) > P(b^\top \mid \mathbf{x})$ ... (iii)

Therefore combining (i), (ii) and (iii), we get
$P(a^{\top} \mid \mathbf{x}) > P(b^{\top} \mid \mathbf{x}) > P(b_{\perp} \mid \mathbf{x}) > P(a_{\perp} \mid \mathbf{x})$.

This shows that $A = a^{\top}$ has the maximum (and $A = a_{\perp}$ has the minimum) posterior probability, and hence it is possibly the best MAP assignment for query variable A (given our assumptions). In a similar way, this technique can be applied to more than 2 binary variables to get a descending order of all the posterior probabilities given the current evidence. In other words using standard Bayesian inference technique, we estimate the posterior probability of the query variables in **Q.** In Q^{T} , we store the query variable that has maximum posterior probability. The most likely assignment $q^{\top}$ for this variable is our best guess for the MAP assignment.

7. 3. Complexity

The heuristic, assigns a value to the MAP assignment at each step of the algorithm. Hence if there are n query variables, then the algorithm will run n-times. And at each step we update the evidence set and therefore need to perform inference n-times.
Therefore the complexity of Heuristic-I is:

$$\mathbf{O}\ (n * \text{P\_Query})$$

where P_Query is the cost of inference i.e. computing the posterior marginals for all the query variables. This shows that our Heuristic-I is linear in the cost of probabilistic query inference and therefore it becomes tractable if the cost of probabilistic query inference is tractable.

7. 4. Formal proof of correctness for two variables

In this section, we present a formal proof to show that for two binary MAP variables this heuristic guarantees to give a correct MAP assignment.

Proposition: For two binary query variables the Heuristic I is guaranteed to return the correct MAP assignment.

Lemma 0[8]**:** In case of MAP of two binary variables, the combination of the most likely values is always greater than the combination of the least likely values.

Proof:

Let $a^{\top}, a_{\perp}, b^{\top}, b_{\perp}$ be such that $P(a^{\top} \mid \mathbf{x}) > P(a_{\perp} \mid \mathbf{x})$ and $P(b^{\top} \mid \mathbf{x}) > P(b_{\perp} \mid \mathbf{x})$

[$a^{\top}$ and $b^{\top}$ being the most likely value of A and B resp.]

[$a_{\perp}$ and $b_{\perp}$ being the least likely value of A and B resp.]

$P(a^{\top} \mid \mathbf{x}) > P(a_{\perp} \mid \mathbf{x})$ [assumption]

$\Leftrightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) + P(a^{\top}, b_{\perp} \mid \mathbf{x}) > P(a_{\perp}, b^{\top} \mid \mathbf{x}) + P(a_{\perp}, b_{\perp} \mid \mathbf{x})$ [Marginalising]

$\Leftrightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) - P(a_{\perp}, b_{\perp} \mid \mathbf{x}) > P(a_{\perp}, b^{\top} \mid \mathbf{x}) - P(a^{\top}, b_{\perp} \mid \mathbf{x})$ (i)

$P(b^{\top} \mid \mathbf{x}) > P(b_{\perp} \mid \mathbf{x})$ [assumption]

$\Leftrightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) + P(a_{\perp}, b^{\top} \mid \mathbf{x}) > P(a^{\top}, b_{\perp} \mid \mathbf{x}) + P(a_{\perp}, b_{\perp} \mid \mathbf{x})$ [Marginalising]

$\Leftrightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) - P(a_{\perp}, b_{\perp} \mid \mathbf{x}) > P(a^{\top}, b_{\perp} \mid \mathbf{x}) - P(a_{\perp}, b^{\top} \mid \mathbf{x})$ (ii)

(i) + (ii)

$\Leftrightarrow 2 \cdot [\, P(a^{\top}, b^{\top} \mid \mathbf{x}) - P(a_{\perp}, b_{\perp} \mid \mathbf{x}) \,] >$

$P(a_{\perp}, b^{\top} \mid \mathbf{x}) - P(a^{\top}, b_{\perp} \mid \mathbf{x}) + P(a^{\top}, b_{\perp} \mid \mathbf{x}) - P(a_{\perp}, b^{\top} \mid \mathbf{x})$

$\Leftrightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) - P(a_{\perp}, b_{\perp} \mid \mathbf{x}) > 0$

$\Leftrightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) > P(a_{\perp}, b_{\perp} \mid \mathbf{x})$

[8] We start with lemma number 0, since this lemma is true in general (MAP of 2 variables), whereas the lemmas after this make some assumptions.

Result: $P(\, a^{\top}, b^{\top} \mid \mathbf{x} \,) > P(\, a_{\perp}, b_{\perp} \mid \mathbf{x} \,)$

We introduce the following lemma to know if the combination of most likely value of A and least likely value of B is greater or lesser than the least likely value of A and the most likely value of B.

Lemma 0.1: $P(\, b^{\top} \mid \mathbf{x} \,) - P(\, b_{\perp} \mid \mathbf{x} \,) > P(\, a^{\top} \mid \mathbf{x} \,) - P(\, a_{\perp} \mid \mathbf{x})$
$\Leftrightarrow P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,) > P(\, a^{\top}, b_{\perp} \mid \mathbf{x} \,)$

Proof:

$P(\, b^{\top} \mid \mathbf{x} \,) - P(\, b_{\perp} \mid \mathbf{x} \,) > P(\, a^{\top} \mid \mathbf{x} \,) - P(\, a_{\perp} \mid \mathbf{x})$

$\Leftrightarrow\ P(\, a^{\top}, b^{\top} \mid \mathbf{x} \,) + P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,) - P(\, a^{\top}, b_{\perp} \mid \mathbf{x} \,) - P(\, a_{\perp}, b_{\perp} \mid \mathbf{x} \,) >$

$P(\, a^{\top}, b^{\top} \mid \mathbf{x} \,) + P(\, a^{\top}, b_{\perp} \mid \mathbf{x} \,) - P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,) - P(\, a_{\perp}, b_{\perp} \mid \mathbf{x} \,)$ [Marginalising]

$\Leftrightarrow\ 2 \cdot P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,) > 2 \cdot P(\, a^{\top}, b_{\perp} \mid \mathbf{x} \,)$

$\Leftrightarrow\ P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,) > P(\, a^{\top}, b_{\perp} \mid \mathbf{x} \,)$

Result: $P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,) > P(\, a^{\top}, b_{\perp}` \mid \mathbf{x} \,)$

Since $b^{\top}$ is the most likely value of B, and the difference between the most likely value and least likely value of B is greater than that of A- as per our heuristic, we enter the evidence of $b^{\top}$. Subsequently in the following two lemma we find out which value of A becomes more likely (with additional evidence of $b^{\top}$). And we see that value of most likely of A (with additional evidence of $b^{\top}$) and $b^{\top}$ is the correct MAP assignment.

Lemma 0.1.1: $P(\, a^{\top} \mid b^{\top}, \mathbf{x} \,) > P(\, a_{\perp} \mid b^{\top}, \mathbf{x} \,) \Rightarrow P(\, a^{\top}, b^{\top} \mid \mathbf{x} \,) > P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,)$

Proof:

$P(\, a^{\top} \mid b^{\top}, \mathbf{x} \,) > P(\, a_{\perp} \mid b^{\top}, \mathbf{x} \,)$

$\Rightarrow P(\, a^{\top} \mid b^{\top}, \mathbf{x} \,) \cdot P(\, b^{\top} \mid \mathbf{x} \,) > P(\, a_{\perp} \mid b^{\top}, \mathbf{x} \,) \cdot P(\, b^{\top} \mid \mathbf{x} \,)$

[Multiplying both sides by $P(\, b^{\top} \mid \mathbf{x} \,)$]

$\Leftrightarrow P(\, a^{\top}, b^{\top} \mid \mathbf{x} \,) > P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,)$ [By definition of cond. prob]

Result: $P(\, a^{\top}, b^{\top} \mid \mathbf{x} \,) > P(\, a_{\perp}, b^{\top} \mid \mathbf{x} \,)$

MAP Result:

$P(a^{\top}, b^{\top} \mid \mathbf{x}) > P(a_{\perp}, b_{\perp} \mid \mathbf{x})$ [Lemma 0]

$P(a_{\perp}, b^{\top} \mid \mathbf{x}) > P(a^{\top}, b_{\perp} \mid \mathbf{x})$ [Lemma 0.1]

$P(a^{\top}, b^{\top} \mid \mathbf{x}) > P(a_{\perp}, b^{\top} \mid \mathbf{x})$ [Lemma 0.1.1]

MAP Assignment $\Rightarrow a^{\top}, b^{\top}$

Lemma 0.1.2: $P(a^{\top} \mid b^{\top}, \mathbf{x}) < P(a_{\perp} \mid b^{\top}, \mathbf{x})$

$\Rightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) < P(a_{\perp}, b^{\top} \mid \mathbf{x})$ and $P(a_{\perp}, b_{\perp} \mid \mathbf{x}) < P(a^{\top}, b_{\perp} \mid \mathbf{x})$

Proof:

$P(a^{\top} \mid b^{\top}, \mathbf{x}) < P(a_{\perp} \mid b^{\top}, \mathbf{x})$

$\Rightarrow P(a^{\top}, b^{\top} \mid \mathbf{x}) < P(a_{\perp}, b^{\top} \mid \mathbf{x})$

[see proof of previous lemma with > replaced by <]

$\Leftrightarrow P(b^{\top} \mid a^{\top}, \mathbf{x}) \cdot P(a^{\top} \mid \mathbf{x}) < P(b^{\top} \mid a_{\perp}, \mathbf{x}) \cdot P(a_{\perp} \mid \mathbf{x})$ [By def. of cond. prob]

$\Leftrightarrow [\, P(a^{\top} \mid \mathbf{x}) \,/\, P(a_{\perp} \mid \mathbf{x}) \,] < [\, P(b^{\top} \mid a_{\perp}, \mathbf{x}) \,/\, P(b^{\top} \mid a^{\top}, \mathbf{x}) \,]$

[Assuming $P(a_{\perp} \mid \mathbf{x}) > 0$ and $P(b^{\top} \mid a^{\top}, \mathbf{x}) > 0$]

Now recall that $P(a^{\top} \mid \mathbf{x}) > P(a_{\perp} \mid \mathbf{x})$ i.e.

$\Leftrightarrow 1 < [\, P(a^{\top} \mid \mathbf{x}) \,/\, P(a_{\perp} \mid \mathbf{x}) \,] < [\, P(b^{\top} \mid a_{\perp}, \mathbf{x}) \,/\, P(b^{\top} \mid a^{\top}, \mathbf{x}) \,]$

$\Rightarrow P(b^{\top} \mid a^{\top}, \mathbf{x}) < P(b^{\top} \mid a_{\perp}, \mathbf{x})$

$\Leftrightarrow 1 - P(b_{\perp} \mid a^{\top}, \mathbf{x}) < 1 - P(b_{\perp} \mid a_{\perp}, \mathbf{x})$ [Taking the complement of $b^{\top}$]

$\Leftrightarrow P(b_{\perp} \mid a^{\top}, \mathbf{x}) > P(b_{\perp} \mid a_{\perp}, \mathbf{x})$

$\Rightarrow P(b_{\perp} \mid a^{\top}, \mathbf{x}) \cdot P(a^{\top} \mid \mathbf{x}) > P(b_{\perp} \mid a_{\perp}, \mathbf{x}) \cdot P(a_{\perp} \mid \mathbf{x})$

[Since $P(a^{\top} \mid \mathbf{x}) > P(a_{\perp} \mid \mathbf{x})$]

$\Leftrightarrow P(a^{\top}, b_{\perp} \mid \mathbf{x}) > P(a_{\perp}, b_{\perp} \mid \mathbf{x})$

Result: $P(a^{\top}, b^{\top} \mid \mathbf{x}) < P(a_{\perp}, b^{\top} \mid \mathbf{x})$ and $P(a_{\perp}, b_{\perp} \mid \mathbf{x}) < P(a^{\top}, b_{\perp} \mid \mathbf{x})$

MAP Result:

$P(a^{\top}, b^{\top} \mid \mathbf{x}) > P(a_{\perp}, b_{\perp} \mid \mathbf{x})$ [Lemma 0]

$P(a_{\perp}, b^{\top} \mid \mathbf{x}) > P(a^{\top}, b_{\perp} \mid \mathbf{x})$ [Lemma 0.1]

$P(a^{\top}, b^{\top} \mid \mathbf{x}) < P(a_{\perp}, b^{\top} \mid \mathbf{x})$ [Lemma 0.1.2]

$P(a_{\perp}, b_{\perp} \mid \mathbf{x}) < P(a^{\top}, b_{\perp} \mid \mathbf{x})$ [Lemma 0.1.2]

MAP Assignment $\Rightarrow a_{\perp}, b^{\top}$

---

7. 4. 1. Discussion

In Figure 7.4.1, we use the orange boxes to depict the intermediary steps taken according to our heuristic. For each of those intermediary steps we verify the action of our heuristic using all the Lemmas presented in the previous sections and is depicted as blue boxes in Figure 7.4.1. In this figure we see that for each of the four possible combinations, the Lemma agrees with the action of our heuristic. Therefore we conclude that this heuristic guarantees a correct MAP assignment for a MAP problem involving two binary variables.

It should be noted that the proof does not take into consideration any equality. It can be easily seen that if equality exists at any stage then the MAP will contain both the assignments of its subtrees. Furthermore, in the previous section, we do not introduce the lemmas 0.2, 0.2.1 and 0.2.2 because the query variables A and B are arbitrary. These Lemmas can be easily proven by interchanging A and B and vice-versa (i.e. $a^{\top} \leftrightarrow b^{\top}$, $a_{\perp} \leftrightarrow b_{\perp}$) in the Lemmas presented in the previous section.

Based on the formal analysis in the previous section, we also can present the following two corollaries.

Corollary 1: In case of MAP of 2 binary variables, the combination of the most likely values is always greater than the combination of the least likely values.

Proof: As proved in lemma 0 in Section 7.4.

Corollary 2: In case of MAP of 2 binary variables, the combination of the least likely values can never be the MAP assignment (or in other words the MAP assignment always consists of at least one of the most likely values).

Proof: Following from Lemma 0, since the probability of the combination of most likely values is always greater than the probability of the combination of least likely values, the least likely combination can never be the MAP assignment.

In the next sections we analyse the behaviour of this heuristic and investigate if these two corollaries hold for 3 or more variables. These two corollaries will help us in analysing the properties of such networks. Moreover in the next sections we will be able to see how these properties change when we increase the number of query variables.

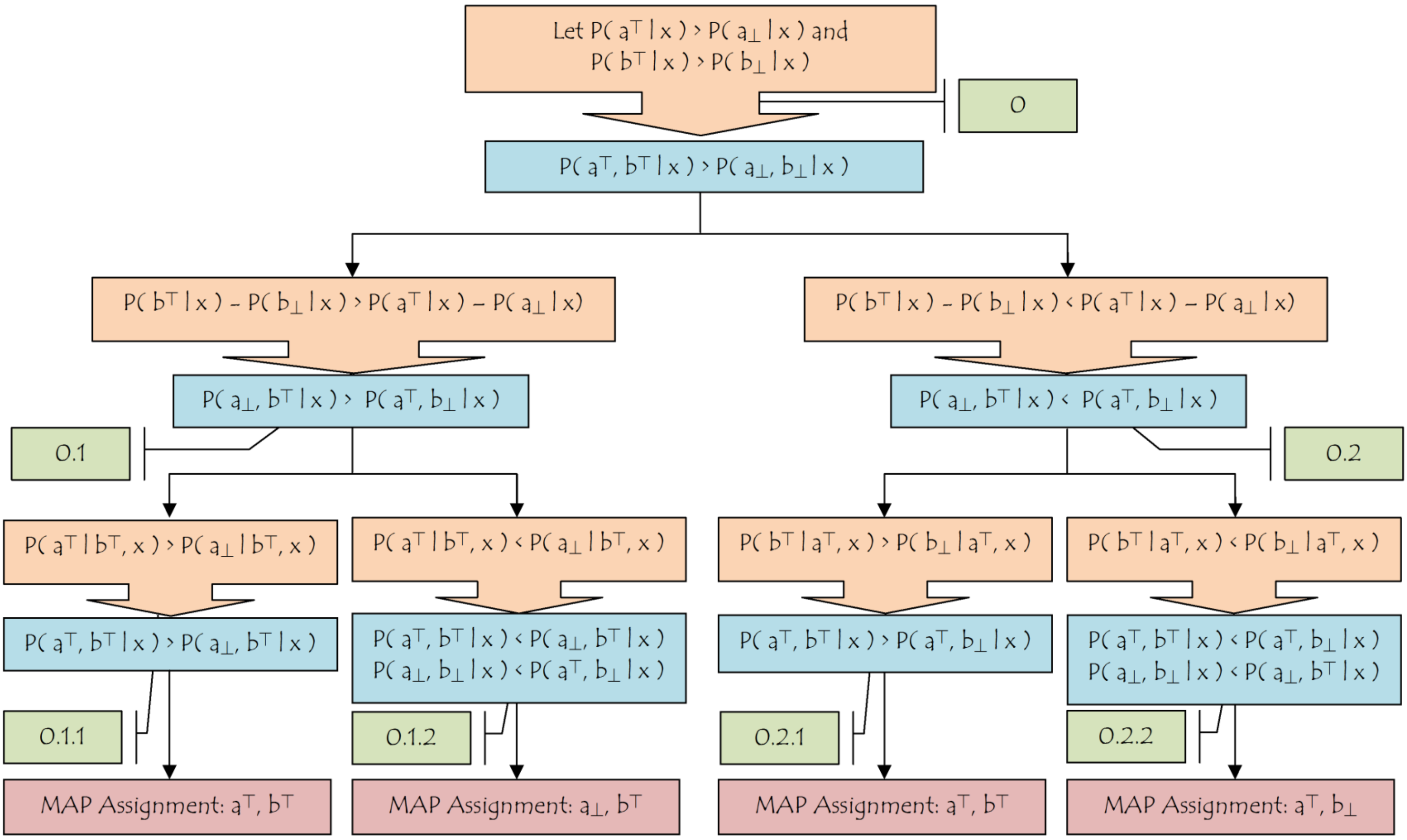

Let $P(a^\top \mid x) > P(a_\perp \mid x)$ and
$P(b^\top \mid x) > P(b_\perp \mid x)$
0
$P(a^\top, b^\top \mid x) > P(a_\perp, b_\perp \mid x)$
$P(b^\top \mid x) - P(b_\perp \mid x) > P(a^\top \mid x) - P(a_\perp \mid x)$
$P(b^\top \mid x) - P(b_\perp \mid x) < P(a^\top \mid x) - P(a_\perp \mid x)$
$P(a_\perp, b^\top \mid x) > P(a^\top, b_\perp \mid x)$
$P(a_\perp, b^\top \mid x) < P(a^\top, b_\perp \mid x)$
0.1
0.2
$P(a^\top \mid b^\top, x) > P(a_\perp \mid b^\top, x)$
$P(a^\top \mid b^\top, x) < P(a_\perp \mid b^\top, x)$
$P(b^\top \mid a^\top, x) > P(b_\perp \mid a^\top, x)$
$P(b^\top \mid a^\top, x) < P(b_\perp \mid a^\top, x)$
$P(a^\top, b^\top \mid x) > P(a_\perp, b^\top \mid x)$
$P(a^\top, b^\top \mid x) < P(a_\perp, b^\top \mid x)$
$P(a_\perp, b_\perp \mid x) < P(a^\top, b_\perp \mid x)$
$P(a^\top, b^\top \mid x) > P(a^\top, b_\perp \mid x)$
$P(a^\top, b^\top \mid x) < P(a^\top, b_\perp \mid x)$
$P(a_\perp, b_\perp \mid x) < P(a_\perp, b^\top \mid x)$
0.1.1
0.1.2
0.2.1
0.2.2
MAP Assignment: $a^\top, b^\top$
MAP Assignment: $a_\perp, b^\top$
MAP Assignment: $a^\top, b^\top$
MAP Assignment: $a^\top, b_\perp$

Figure 7.4.1: Pictorial representation of the proposition (*for two binary variables the algorithm is guaranteed to return the correct MAP assignment*). The lemmas are in orange, the results in blue and the MAP assignment in red.

7. 5. Analysis for 3 or more variables

Instead of a formal proof (as done in Section 7.4) we show a counter example where this heuristic fails. This example is the evidence that our heuristic does not always guarantee to give the correct MAP assignment for a MAP problem involving 3 or more variables.

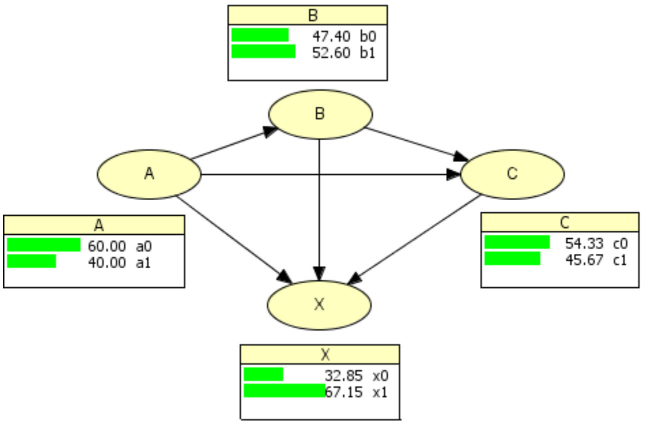


Figure 7.5.1: An example Bayesian network where the heuristic identifies an incorrect MAP assignment.

Let us consider a Bayesian Network with the graph structure as shown in Figure 7.5.1 and the conditional probability tables as shown in Table 7.5.1. In this network A, B and C are the query variables and X is the evidence variable, all of which are binary. In this case, a MAP assignment problem would seek the most likely assignment to the query variables given the evidence. This example can be also considered as an MPE problem because we do not have any non-query and non-evidence variables.

P(A)	
a^0	0.6
a^1	0.4

P(B | A)		
	a^0	a^1
b^0	0.63	0.24
b^1	0.37	0.76

P(C | A, B)				
	a^0		a^1	
	b^0	b^1	b^0	b^1
c^0	0.5	0.31	0.06	0.92
c^1	0.5	0.69	0.94	0.08

P(X | A, B, C)								
	a^0				a^1			
	b^0		b^1		b^0		b^1	
	c^0	c^1	c^0	c^1	c^0	c^1	c^0	c^1
x^0	0.53	0.1	0.88	0.47	0.06	0.13	0.21	0.25
x^1	0.47	0.9	0.12	0.53	0.94	0.87	0.79	0.75

Table 7.5.1: Conditional probability tables for the example Bayesian network from Figure 7.5.1.

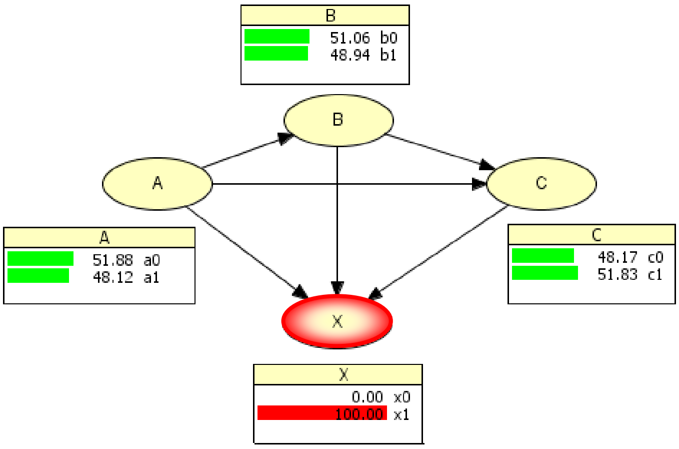
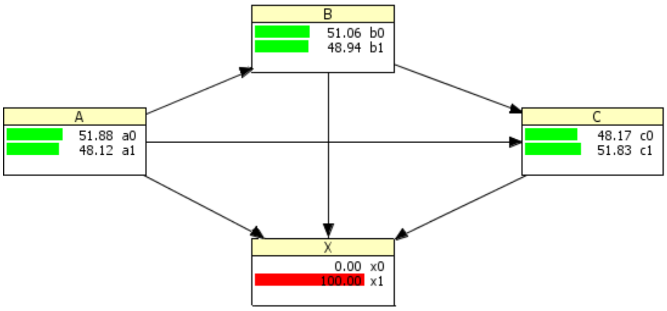


Figure 7.5.2: Posterior probabilities of A, B and C given evidence X = x^1.

$P(A, B, C \mid X = x^1)$	
$P(a^0, b^0, c^0 \mid x^1)$	0.132
$P(a^0, b^0, c^1 \mid x^1)$	0.253
$P(a^0, b^1, c^0 \mid x^1)$	0.012
$P(a^0, b^1, c^1 \mid x^1)$	0.121
$P(a^1, b^0, c^0 \mid x^1)$	0.008
$P(a^1, b^0, c^1 \mid x^1)$	0.117
$P(a^1, b^1, c^0 \mid x^1)$	**0.329**
$P(a^1, b^1, c^1 \mid x^1)$	0.027

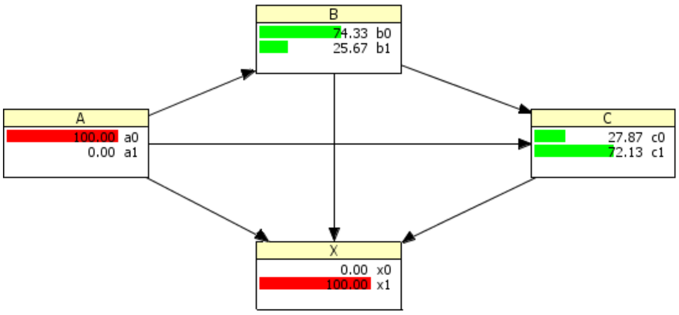
Table 7.5.2: Joint probability distribution of A, B, C given X = x^1

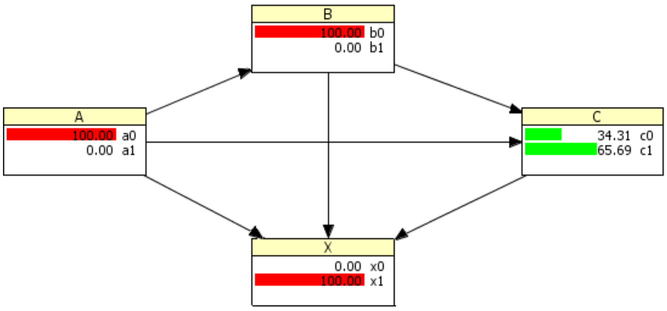
Given the evidence X = x^1, the posterior probabilities of A, B and C is shown in Figure 7.5.2. In Table 7.5.2 we show the joint probability distribution conditioned on the evidence i.e. $P(A, B, C \mid X = x^1)$. From Table 7.5.2, we see the most likely assignment of the query variables (i.e. MAP assignment) is (a^1, b^1, c^0) with a probability of 0.329. However, our heuristic (as illustrated in Figure 7.5.3) identifies an incorrect MAP assignment of (a^0, b^0, c^1), which has a probability of 0.253 only.



Step 1: Given the evidence X = x^1, we see that the $P(a^0|x^1)$ has the maximum posterior probability and $P(a^1|x^1)$ has the minimum posterior probability. Therefore according to the heuristic, a^0 is considered a part of the MAP assignment and hence additional evidence of a^0 is introduced.

Step 2: Given the current evidence, we see that $P(b^0|a^0,x^1)$ has the maximum posterior probability. Therefore according to the heuristic, b^0 is also considered to part of the MAP assignment and hence additional evidence of b^0 is introduced.





Step 3: Finally given the current evidence, we see that $P(c^1|a^0,b^0,x^1)$ has the maximum posterior probability. Therefore we consider c^1 to be part of the MAP assignment.

As a result the **MAP assignment** according to the **heuristic** is (a^0, b^0, c^1)

Figure 7.5.3: Step by step working of heuristic for the example Bayesian network discussed in section 7.5.

7. 5. 1. Discussion

For 3 variables we have shown that there is at least one case where our heuristic fails to arrive at the correct MAP assignment. Hence the heuristic approximates MAP for more than 2 query variables.

The two corollaries discussed in Section 7.4.1 are no longer valid for 3 or more variables. For example, if we consider the example Bayesian network (illustrated in Figure 7.5.1), we can see that given the evidence X = x^1, the most likely value assignments are a^0, b^0 and c^1; therefore the least likely values are a^1, b^1 and c^0. As shown in Table 7.3.2.2, $P(a^1, b^1, c^0 \mid x^1) = 0.329 > P(a^0, b^0, c^1 \mid x^1) = 0.253$. Which shows that the combination of the most likely values is not always greater that the combination of the least likely values. Therefore *Corollary 1* can not be generalised to more than 2 variables. Furthermore, in the same example, we also see that the combination of the least likely values assignment is the correct MAP assignment. This contradicts *Corollary 2* which states that, in case of MAP of 2 binary variables, the combination of the least likely values can never be the MAP assignment.

Next, we show that our heuristic is never worse than the combination of most likely values heuristic (discussed in Section 6.3.2.1).

Corollary 3: In case of MAP of 3 binary variables, whenever the combination of the most likely values heuristic gives a correct MAP assignment, then our heuristic will also give a correct MAP assignment.

Proof: Hypothetically, let's assume that the combination of most likely values heuristic will give a correct MAP assignment. Then this heuristic, in the first step chooses a most likely value which will be part of the correct MAP assignment. After which we are left with 2 query variables, and we have seen in the previous section 'for two binary variables the heuristic is guaranteed to return the correct MAP assignment'. Hence the heuristic will always give a correct MAP assignment, if the combination of most likely values heuristic is able to do so.

7. 6. Experiments

In this section, we perform experiments to compare the performance of our heuristic and the commonly used combination of most likely values heuristic (discussed in Section 6.3.2.1). For this study, we consider two, three and four binary query variables. For simplicity we also assume a single binary evidence variable X ∈ $\{x^0, x^1\}$. As we have a single binary evidence variable, we perform two MAP queries on each network i.e. P(**Q** | X = x^0) and P(**Q** | X = x^1). We compute accuracy by counting the number of correct MAP assignments. Furthermore, we only consider an MPE problem (special case of MAP), where we do not have any non-query and non-evidence variables.

7. 6. 1. MAP of 2 variables

In this section we discuss the two experiments we performed, on networks containing 2 query variables. In these experiments, we modified the parameters of the network for a densely and sparsely connected network.

Experiment 1: MAP of 2 binary variables in a densely connected network

In this experiment we use a fully-connected Bayesian network structure as illustrated in Figure 7.6.1.1. In this network there are two binary query variables: A and B and one evidence binary variable X.

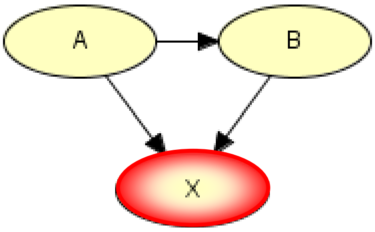


Figure 7.6.1.1: Bayesian network graph for experiment 1

The set of parameter probabilities for this graph are $P(a^0)$; $P(b^0 \mid A)$ – i.e. $P(b^0 \mid a^0)$, $P(b^0 \mid a^1)$; $P(x^0 \mid A, B)$ – i.e. $P(x^0 \mid a^0, b^0)$, $P(x^0 \mid a^0, b^1)$, $P(x^0 \mid a^1, b^0)$, $P(x^0 \mid a^1, b^1)$. We iteratively change these 7 parameters in steps of 0.1 from probability value of zero to probability value of one. We avoid any parameter assignments of zero (or one) to the network parameters. Since zero (or one) probability is a very special value in probability theory, as it denotes an impossible event. For simplicity of our experiments, we avoid such cases by changing parameter assignment of 0 to 0.00001 or parameter assignment of 1 to 0.99999. In other words, each parameter probability takes a value from the following 11 numbers: {0.00001, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99999}.

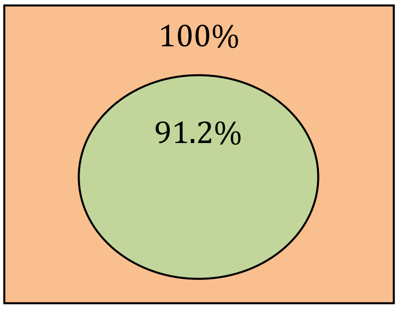


Figure 7.6.1.2: Accuracy

🟧 Our Heuristic I- 100%

🟩 Combination of most likely values Heuristic- 91.2%

In this network there are 7 parameters, and each parameter can take any of the 11 probability values. Therefore in this experiment we generate 11^7= 19,487,171 networks with different set of parameter probabilities. As mentioned before, since we have a single binary evidence variable X, we can perform two MAP queries on each network i.e. MAP(A, B | x^0) and MAP(A, B | x^1). Therefore with 11^7 networks, we can perform 11^7 x 2 = 38,974,342 MAP queries.

To calculate the accuracy we use the number of correct MAP assignments. We have seen in Section 7.3.1 that our heuristic always gives a correct assignment for 2 binary variables. However the combination of most likely values heuristic gave a correct MAP assignment around 91.2% times (35,527,609 out of 38,974,342 queries) as shown in Figure 7.6.1.2.

Experiment 2: MAP of 2 binary variables in a sparsely connected network

In this experiment we use a sparsely-connected Bayesian network structure as illustrated in Figure 7.6.1.3. In this network there are two binary query variables: A and B and one binary evidence variable X. The query variables are marginally independent.

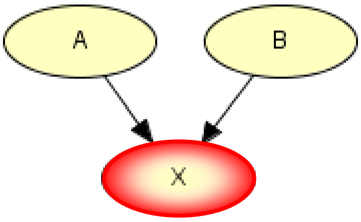


Figure 7.6.1.3: Bayesian network graph for experiment 2

The set of parameter probabilities for this graph are $P(a^0)$; $P(b^0)$; $P(x^0 \mid A, B)$ – i.e. $P(x^0 \mid a^0, b^0)$, $P(x^0 \mid a^0, b^1)$, $P(x^0 \mid a^1, b^0)$, $P(x^0 \mid a^1, b^1)$. Similar to experiment 1, these 6 parameters probabilities can take a value from the following 11 numbers: {0.00001, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99999}. Similar to experiment 1 we also avoid any parameter assignments of zero or one to the network parameters.

In this network there are 6 parameters, and each parameter can take any of the 11 probability values. Therefore in this experiment we generate 11^6= 1,771,561 networks with different set of parameter probabilities. As mentioned before, since we have a single binary evidence variable X, we can perform two MAP queries on each network i.e. MAP(A, B | x^0) and MAP(A, B | x^1). Therefore with 11^6 networks, we can perform 11^6 x 2 = 3,543,122 MAP queries.

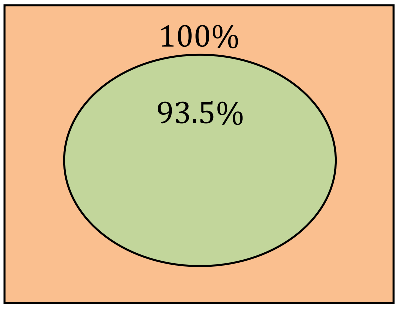


Figure 7.6.1.4: Accuracy

🟧 Our Heuristic I- 100%

🟩 Combination of most likely values Heuristic- 93.5%

Similar to experiment 1 we use the number of correct MAP assignments to compute the accuracy. We have seen in Section 7.3.1 that our heuristic always gives a correct assignment for 2 binary variables. However the combination of most likely values heuristic gave a correct MAP assignment around 93.5% times (3,310,947 out of 3,543,122 queries) as shown in Figure 7.6.1.4.

7. 6. 2. MAP of 3 variables

In this section we discuss the two experiments we performed on networks containing 3 query variables. In these experiments, we modified the parameters of the network for a densely and sparsely connected network.

Experiment 3: MAP of 3 binary variables in a densely connected network

In this experiment we use a fully-connected Bayesian network structure as illustrated in Figure 7.6.2.1. In this network there are three binary query variables: A, B and C and one evidence variable X.

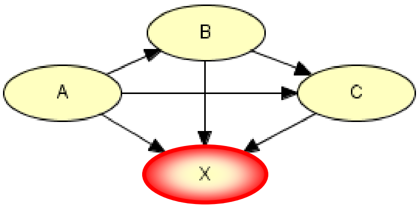


Figure 7.6.2.1: Bayesian network graph for experiment 3

The set of parameter probabilities for this graph are one parameter for $P(a^0)$; two parameters for $P(b^0 \mid A)$; four parameters for $P(c^0 \mid A, B)$ and finally eight parameters for $P(x^0 \mid A, B, C)$.

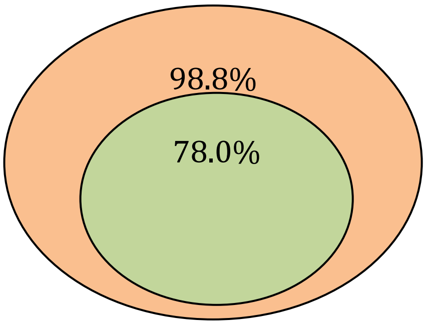


Figure 7.6.2.2: Accuracy

🟧 Our Heuristic I- 98.8%

🟩 Combination of most likely values Heuristic- 78.0%

We generate 1,000,000 networks by assign pseudo-random[9] numbers to these 15 parameters. Similar to the previous experiments we avoid any assignments of zero or one to the network parameters. As mentioned before, since we have a single binary evidence variable X, we can perform two MAP queries on each network i.e. MAP(A, B, C | x^0) and MAP(A, B, C | x^1). Therefore with 10^6 networks, we can perform $2 \times 10^6 = 2{,}000{,}000$ MAP queries.

[9] To generate random numbers a pseudorandom number generator (PRNG) of C++ is used. PRNG uses an algorithm for generating a sequence of numbers that approximates the properties of random numbers. The sequence is not truly random in the sense that it is completely determined by a seed state- which is used by the algorithm to generate the random series. The same seed state will always produce the same sequence of random numbers.

We use the number of correct MAP assignments to compute the accuracy. Our heuristic gave a correct MAP assignment around 98.8% times (1,976,921 out of 2 x 10^6) compared to the combination of most likely values heuristic which gave a correct MAP assignment around 78.0% times (1,560,478 out of 2 x 10^6 queries) as depicted in Figure 7.6.2.2.

Experiment 4: MAP of 3 binary variables in a sparsely connected network

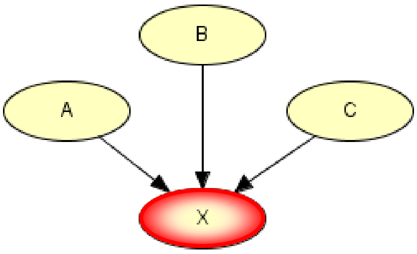


Figure 7.6.2.3: Bayesian network graph for experiment 4

In this experiment we use a sparsely-connected Bayesian network structure as illustrated in Figure 7.6.2.3. In this network there are three binary query variables: A, B and C and one evidence variable X. The query variables are marginally independent.

The set of parameter probabilities for this graph are one parameter for P(a^0); one parameter for P(b^0); one parameter for P(c^0) and finally eight parameters for P(x^0 | A, B, C).

Similar to experiment 3, we generate 1,000,000 networks by assign pseudo-random numbers to these 11 parameters. We also avoid any assignments of zero or one to the network parameters. As mentioned before, since we have a single binary evidence variable X, we can perform two MAP queries on each network i.e. MAP(A, B, C | x^0) and MAP(A, B, C | x^1). Therefore with 10^6 networks, we can perform 2 x 10^6 = 2,000,000 MAP queries.

We use the number of correct MAP assignments to compute the accuracy. Our heuristic gave a correct MAP assignment around 99.7% times (1,994,357 our of 2 x 10^6) compared to the combination of most likely values heuristic which gave a correct MAP assignment around 85.4% times (1,701,090 out of 2 x 10^6 queries) as depicted in Figure 7.6.2.4.

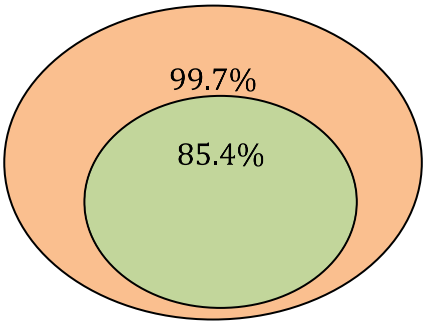


Figure 7.6.2.4: Accuracy

🟧 Our Heuristic I- 99.7%

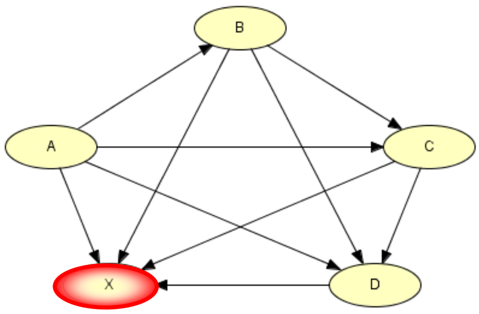
🟩 Combination of most likely values Heuristic- 85.4%

7. 6. 3. MAP of 4 variables

Similar to the previous sections, in this section we discuss the two experiments we performed on networks containing 4 query variables. In these experiments, we modified the parameters of the network for a densely and sparsely connected network.

Experiment 5: MAP of 4 binary variables in a densely connected network

In this experiment we use a fully-connected Bayesian network structure as illustrated in Figure 7.6.3.1. In this network there are four binary query variables: A, B, C and D and one evidence variable X.



The set of parameter probabilities for this graph are one parameter for P(a^0); two parameters for P(b^0 | A); four parameters for P(c^0 | A, B); eight parameters for P(d^0 | A, B, C) and finally sixteen parameters for P(x^0 | A, B, C, D).

Figure 6.6.3.1: Bayesian network graph for experiment 5

We generate 1,000,000 networks by assign pseudo-random numbers to these 31 parameters. Similar to the previous experiments we avoid any assignments of zero or one to the network parameters. As mentioned before, since we have a single binary evidence variable X, we can perform two MAP queries on each network i.e. MAP(A, B, C, D | x^0) and MAP(A, B, C, D | x^1). Therefore with 10^6 networks, we can perform 2 x 10^6 = 2,000,000 MAP queries.

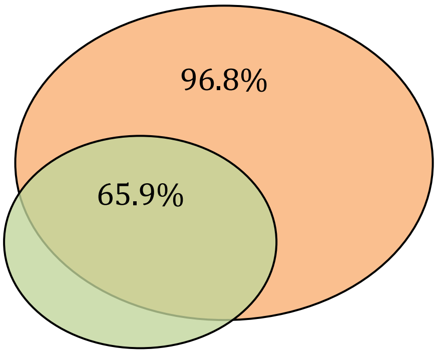


Figure 7.6.3.2: Accuracy

🟧 Our Heuristic I- 96.8%

🟩 Combination of most likely values Heuristic- 65.9%

We use the number of correct MAP assignments to compute the accuracy. Our heuristic gave a correct MAP assignment around 96.8% times (1,935,285 out of 2 x 10^6) compared to the combination of most likely values heuristic which gave a correct MAP assignment around 65.9% times (1,318,570 out of 2 x 10^6 queries) as shown in Figure 7.6.3.2.

Contrary to Corollary 3, in our experiment we found in 0.06% (1207) of the MAP queries, that the combination of most likely values heuristic gave correct MAP assignments whereas our heuristic failed to do so. Hence for more than 3 binary query variables the combination of most likely values heuristic $\nsubseteq$ our heuristic.

Experiment 6: MAP of 4 binary variables in a sparsely connected network

In this experiment we use a sparsely-connected Bayesian network structure as illustrated in Figure 7.6.3.3. In this network there are four binary query variables: A, B, C, D and one evidence variable X. The query variables A and B are marginally independent. Whereas A, C and D are conditionally independent given evidence of B.

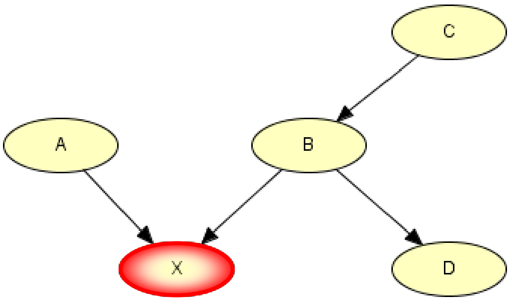


Figure 7.6.3.3: Bayesian network graph for experiment 6

The set of parameter probabilities for this graph are one parameter for $P(a^0)$; two parameters for $P(b^0 \mid C)$; one parameter for $P(c^0)$; two parameters for $P(d^0 \mid B)$ and finally four parameters for $P(x^0 \mid A, B)$.

Similar to the previous experiments, we generate 1,000,000 networks by assign pseudo-random numbers to these 10 parameters. We also avoid any assignments of zero or one to the network parameters. As mentioned before, since we have a single binary evidence variable X, therefore with 10^6 networks, we can perform 2×10^6 = 2,000,000 MAP queries.

We use the number of correct MAP assignments to compute the accuracy. Our heuristic gave a correct MAP assignment around 98.7% times (1,973,935 our of 2×10^6) compared to the combination of most likely values heuristic which gave a correct MAP assignment around 75.4% times (1,507,789 out of 2×10^6 queries) as shown in Figure 7.6.3.4.

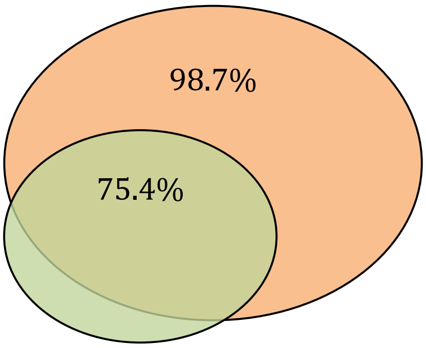


Figure 7.6.3.4: Accuracy

🟧 Our Heuristic I- 98.7%

🟩 Combination of most likely values Heuristic- 75.4%

Similar observations as in experiment 5 were found which proved that for more than 3 binary query variables the combination of most likely values heuristic $\nsubseteq$ our heuristic. In 0.03% (658) of the MAP queries, the combination of most likely values heuristic gave correct MAP assignments whereas our heuristic failed to do so.

7. 6. 4. Discussion

From the experimental results in the previous sections, we conclude that our heuristic performs better than the combination of most likely values heuristic (discussed in 6.3.2.1). However the applicability of this claim is constrained by the assumptions of our experimental methodology (for example, fixed graph structure, the size of the query set is less than 5, MPE assumption, etc).

Both heuristics (our heuristic and combination of most likely values heuristic) seem to perform better on sparse graphs than on dense graphs. However we could not find any reason for it.

Chapter 8:
Heuristic II for approximating MAP

In this chapter we discuss a different variation of the Heuristic I we proposed in the previous chapter. Though this heuristic is computationally more complex, it gives 100% accurate MAP assignment predictions in our simple experiments. Though we need to perform more extensive experiments to say something concrete.

8. 1. Algorithm

In Heuristic-I we are using a greedy approach to get the MAP assignment i.e. at each stage we are making a guess based on the posterior marginals. However, if we start with a wrong guess then we will never get the correct MAP assignment. Our belief is that if we start with a correct MAP assignment to one of the query variables, then the greedy approach of Heuristic-I may lead to a correct MAP assignment to the complete query set. Hence for example, if we choose the query variable Q_i having two values (q_i^0, q_i^1), then

1. We assume q_i^0 is part of the correct MAP assignment. Therefore we add q_i^0 to MAP assignment denoted by $\mathbf{m\_q_i^0}$, remove Q_i from query set $\mathbf{Q}$ and add q_i^0 to the evidence **x.** Subsequently we call Heuristic-I on the current query set ($\mathbf{Q}$ **-** Q_i) and evidence ($\mathbf{x} \wedge q_i^j$) to get the most likely assignments for these query variables.
2. We then assume that the other value of Q_i - q_i^1 is part of the correct MAP assignment and proceed in a similar way to get the MAP assignment $\mathbf{m\_q_i^1}$.

Moreover instead of just performing this for one query variable, we do this for all the query variables so that our chance for finding a MAP increases. In the heuristic described in Figure 8.1.1, $\mathbf{m\_q_i^j}$ denotes the MAP assignment, where we assume the value q_i^j of variable Q_i is part of the MAP assignment, and the rest of the most likely assignments (i.e. for $\mathbf{Q}$ - Q_i) are found by applying Heuristic-I. We store the probability of $\mathbf{m\_q_i^j}$ in the variable prob_ $m\_q_i^j$.

With n binary query variables, we get 2 * n MAP assignments to choose from. Since we have stored the probability of the 2 * n possible MAP assignments, we assume that the correct MAP assignment would be the one which has the maximum probability.

```
Heuristic_Algorithm Greedy_MAP_II ( Bayesian Net **B**, Query Set **Q**, Evidence **x** ):
MAP **m**, prob_m
---------------------------------------------------------------------------

For all $Q_i$ in query set **Q**          // $i \in \{1, \ldots, n\}$
    For all $q_i^j := \{ q_i^0, q_i^1 \}$ do    // $j \in \{0, 1\}$
        **x' := x ; Q' := Q**      // Make a copy of the query set and evidence
        **m_$q_i^j$** := true
        //  **m_$q_i^j$** contains the MAP assignment when we assume that value $q_i^j$ of
        // variable $Q_i$ is part of the MAP assignment

        **m_$q_i^j$** := **m_$q_i^j$** $\wedge$ $q_i^j$
        prob_ m_$q_i^j$ := $P(q_i^j \mid \mathbf{x})$
        **x** := **x** $\wedge$ $q_i^j$
        **Q** := **Q** \ $Q_i$

        (**m_$q_i^j$**, prob_ m_$q_i^j$)  $\wedge$  := Heuristic_Algorithm Greedy_MAP_I (**B**, **Q**, **x** )
        //  **m_$q_i^j$** contains the approximate MAP assignment, where we guess that
        //  value $q_i^j$ of $Q_i$ is part of it.
        //  prob_ m_$q_i^j$ is the probability of the MAP assignment **m_$q_i^j$**

        **x** := **x'; Q** := **Q'**  // Restore the query and evidence set to original set of
                           //  observations
    End
End

**m** := arg $\max_{i,j}$ P( **m_$q_i^j$** | **x** ) // We don't need to perform inference again for this since
                              //   we have calculated it before- P( **m_$q_i^j$** | **x** ) = prob_ m_$q_i^j$
prob_m := $\max_{i,j}$  prob_ m_$q_i^j$
```

Figure 8.1.1: Pseudo code for Heuristic II

8. 1. 1. Complexity

When we are inside the two for-loops, we initially perform probability query inference to obtain P(q_i^j | **x**). And we remove the variable Q_i from the query set. Hence we are left with n-1 query variables. We call Heuristic-I on these n-1 query variables. Recall that P_Query is the cost of inference i.e. computing the posterior marginals for all the query variables. Therefore complexity of these statements is:

O(P_Query + Complexity_of_Heuristic_I_for_n-1_queryvariables)
⇒ **O**(P_Query + {[n-1] * P_Query})
⇒ **O**({1 + [n-1]} * P_Query)
⇒ **O**(n * P_Query)

Since we have n query variables, each having only two values, we perform this 2 * n times. Therefore the complexity of Heuristic-II is:

⇒ **O**(2 * n * {n * P_Query})
⇒ **O**(2 * n^2 * P_Query)

This means that our Heuristic-II is polynomial in the complexity of probabilistic query inference. And still remains tractable if probabilistic query inference is tractable.

8. 1. 2. Theoretical analysis of Heuristic II

In this section we theoretically analyse the heuristic for two and three binary MAP variables. We have seen in the previous chapter that the Heuristic-I is guaranteed to give a correct MAP assignment for 2 query variables. Therefore it is trivial to see that this heuristic will also give a correct MAP assignment for 2 query variables.

For 3 query variables, this heuristic will always give a correct MAP assignment. Initially this heuristic iteratively considers both the possible values for a query variable to be part of the MAP assignment- therefore we are bound to start with a correct MAP assignment to that query variable. After which we call Heuristic-I to find the most likely assignments to the other two query variables. We know from the previous chapter that for two query variables, Heuristic-I is guaranteed to return a correct MAP assignment. Therefore this heuristic will always find a correct MAP assignment for three query variables.

8. 2. Experiments

In the previous section we saw that the Heuristic II is guaranteed to give a correct MAP assignment for 2 and 3 query variables. Therefore we only repeat experiment 5 and 6 (in chapter 7) with the new heuristic.

8. 2. 1. MAP of 4 variables

Experiment 7: MAP of 4 binary variables in a densely connected network

Similar to experiment 5, in this experiment we use a fully-connected Bayesian network structure as illustrated in Figure 8.2.1.1. In this network there are four binary query variables: A, B, C and D and one evidence variable X.

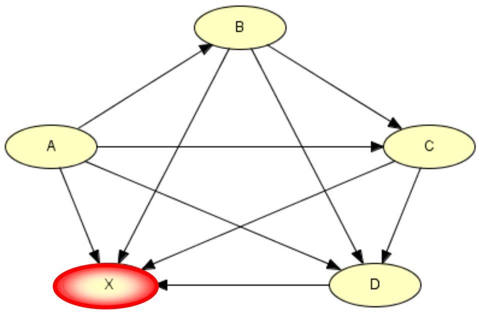


Figure 8.2.1.1: Bayesian network graph for experiment 7

The set of parameter probabilities for this graph are one parameter for $P(a^0)$; two parameters for $P(b^0 \mid A)$; four parameters for $P(c^0 \mid A, B)$; eight parameters for $P(d^0 \mid A, B, C)$ and finally sixteen parameters for $P(x^0 \mid A, B, C, D)$.

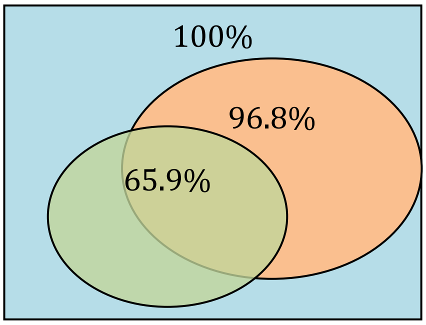


Figure 8.2.1.2: Accuracy

🟦 Our Heuristic II– 100%

🟧 Our Heuristic I- 96.8%

🟩 Combination of most likely values Heuristic- 65.9%

We can generate the same series of pseudo random numbers by using the same seed value. Hence we generate 1,000,000 networks by assigning the same pseudo-random numbers (as in experiment 5) by setting the same seed value. Similar to the previous experiments we avoid any assignments of zero or one to the network parameters. As mentioned before, since we have a single binary evidence variable X, we can perform two MAP queries on each network i.e. MAP(A, B, C, D | x^0) and MAP(A, B, C, D | x^1). Therefore with 10^6 networks, we can perform 2×10^6 = 2,000,000 MAP queries.

We use the number of correct MAP assignments to compute the accuracy. In this experiment our new heuristic always gave a correct MAP assignment compared to the combination of most likely values heuristic which gave a correct MAP assignment around 65.9% times (1,318,570 out of 2,000,000 inferences) as shown in figure 8.2.1.2.

Experiment 8: MAP of 4 binary variables in a sparsely connected network

Similar to experiment 6, in this experiment we use a sparsely-connected Bayesian network structure as illustrated in Figure 8.2.1.3. In this network there are four binary query variables: A, B, C and D and one evidence variable X. The query variables A and B are marginally independent. Whereas A, C and D are conditionally independent given evidence of B.

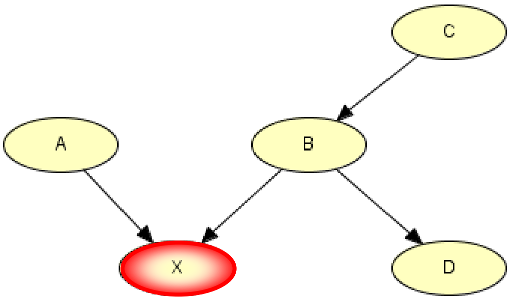


Figure 8.2.1.3: Bayesian network graph for experiment 8

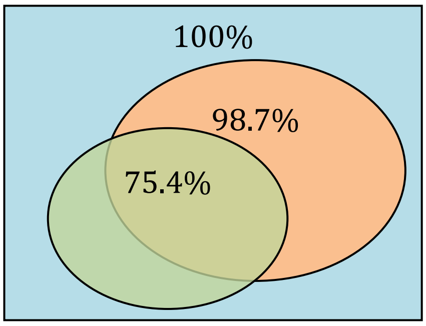


Figure 8.2.1.4: Accuracy

🟦 Our Heuristic II– 100%

🟧 Our Heuristic I– 98.7%

🟩 Combination of most likely values Heuristic- 75.4%

The set of parameter probabilities for this graph are one parameter for $P(a^0)$; two parameters for $P(b^0 \mid C)$; one parameter for $P(c^0)$; two parameters for $P(d^0 \mid B)$ and finally four parameters for $P(x^0 \mid A, B)$.

Similarly we generate the same pseudo random numbers as in experiment 6 using a common seed value. These pseudo random numbers are used to generate 1,000,000 networks. We also avoid any assignments of zero or one to the network parameters. As mentioned before, since we have a single binary evidence variable X, therefore with 10^6 networks, we can perform 2 x 10^6 = 2,000,000 MAP queries.

We use the number of correct MAP assignments to compute the accuracy. In this experiment our new heuristic always gave a correct MAP assignment compared to the combination of most likely values heuristic which gave a correct MAP assignment around 75.4% times (1,507,789 out of 2,000,000 inferences) as shown in figure 8.2.1.4.

8. 2. 2. Discussion

In our experiments we saw that for four query variables our Heuristic-II always gave a correct assignment. Though theoretically we are not able to find any reasons for this behaviour. We need to also see if the heuristic continues to always give an accurate prediction if we increase the number of query variables.

Chapter 9:

Concluding remarks

9. 1. Conclusion

In our experiments we tried to understand the difference between our heuristics and the combination of most likely values heuristic. A MAP query finds the most likely joint assignment to **Q**, whereas a probability query finds the posterior marginals for a single query variable Q_i i.e. computes P(Q_i | **X** = **x**). And we saw that the assignment where each variable picks its most likely value (combination of most likely values heuristic) can be quite difference from the most likely joint assignment to all variables simultaneously.

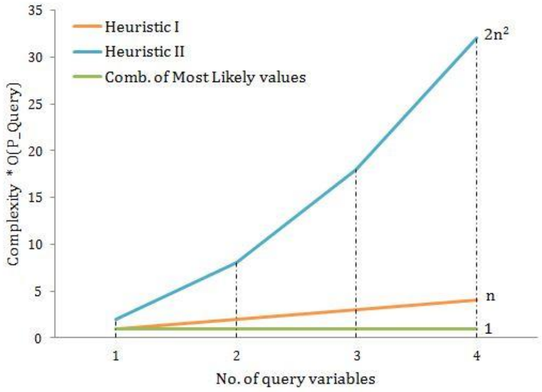


Figure 9.1.1: Complexity comparison for the heuristics

In the Figure 9.1.1 we have depicted the complexity of the heuristics. Though Heuristic I and II are computationally more complex, our simple experiments gave an indication that both our heuristics perform better than the combination of the most likely values heuristic. However to be able to say something stronger, a more detailed study needs to be done. We need to analyse the performance by generating random network structures and also compare its performance with other approximate MAP (abductive) inference algorithms. Also since this problem of abductive inference is similar to multi dimensional classification, we need to evaluate the performance of the heuristic using performance metrics proposed by [Bielza et al., 2011].

In our experiments we have considered MPE problem, which is a special case of MAP. We need to also evaluate how our heuristic performs when the non-evidence, non-query variables are non-empty.

In our heuristics, we have assumed only binary query variables. Future work will include generalising the heuristic to non binary query variables. Also for Heuristic II, we need to theoretically analyse why the heuristic always gives a correct MAP assignment for 4 query variables.

References

[Aitken, 1991] -[Aitken & Stoney, 1991] Aitken, Colin GG, and David A. Stoney. *The use of statistics in forensic science*. CRC Press, 1991.

[ANSI/NIST-ITL 1-2011] American National Standard for Information Systems – Data Format for the Interchange of Fingerprint, Facial& Other Biometric Information, ANSI/NIST-ITL 1-2011

[Baker, 2009] [Baker et al., 2009] Baker, Chris L., Rebecca Saxe, and Joshua B. Tenenbaum. "Action understanding as inverse planning." *Cognition* 113.3 (2009): 329-349.

[Bielza et al., 2011] Bielza, Concha, Guangdi Li, and Pedro Larrañaga. "Multi-dimensional classification with Bayesian networks." International Journal of Approximate Reasoning 52.6 (2011): 705-727.

[Bilmes, 2004] Bilmes, Jeff. "On soft evidence in Bayesian networks." *Dept. of EE, U. of Washington, Tech. Rep. UWEETR-2004-0016* (2004).

[Bodlaender et al., 2002] Bodlaender, Hans L., Frank van den Eijkhof, and Linda C. van der Gaag. "On the complexity of the MPA problem in probabilistic networks." *ECAI*. 2002.

[Brümmer & Preez, 2006] Brümmer, Niko, and Johan du Preez. "Application-independent evaluation of speaker detection." *Computer Speech & Language* 20.2 (2006): 230-275.

[Buntine, 1996] Buntine, Wray. "A guide to the literature on learning probabilistic networks from data." *Knowledge and Data Engineering, IEEE Transactions on* 8.2 (1996): 195-210.

[Champod, 2009] Champod, Christophe. "Identification and individualization." *Wiley Encyclopedia of Forensic Science* (2009).

[Chickering, 1995] Chickering, David Maxwell. "A transformational characterization of equivalent Bayesian network structures." *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1995.

[Chickering et al., 2003] Chickering, David Maxwell, David Heckerman, and Christopher Meek. "Large-sample learning of Bayesian networks is NP-hard." *The Journal of Machine Learning Research* 5 (2004): 1287-1330..

[Cofino et al., 2002] Cofiño, Antonio S., et al. "Bayesian networks for probabilistic weather prediction." *In Proceedings of the 15th Eureopean Conference on Artificial Intelligence, ECAI'2002*. 2002.

[Cooper, 1990] Cooper, Gregory F. "The computational complexity of probabilistic inference using Bayesian belief networks." *Artificial intelligence* 42.2 (1990): 393-405.

[Cover & Thomas, 1991 Cover, Thomas M., and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[Dechter, 1998] Dechter, Rina. "Bucket elimination: A unifying framework for probabilistic inference." *Learning in graphical models*. Springer Netherlands, 1998. 75-104.

[Demirer et al., 2006] Demirer, Riza, Ronald R. Mau, and Catherine Shenoy. "Bayesian networks: a decision tool to improve portfolio risk analysis." *Journal of applied finance* 16.2 (2006): 106.

[Doekhie, 2012] Gina Doekhie. "A Bayesian Network for Assigning Probabilities on which Fingerleft a Mark", MSc Thesis, University of Amsterdam, 2012.

[Evett & Williams, 1996a] Evett, I., and R. Williams. "A review of the sixteen point fingerprint standard in England and Wales." *Journal of Forensic Identification* 46 (1996): 49-73.

[Evett & Buckleton, 1996b] Evett, I. W., and J. S. Buckleton. "Statistical analysis of STR data." *16th Congress of the International Society for Forensic Haemogenetics*

(Internationale Gesellschaft für forensische Hämogenetik eV), Santiago de Compostela, 12–16 September 1995. Springer Berlin Heidelberg, 1996.

[Frey, 1997] Frey, Brendan J. *Bayesian networks for pattern classification, data compression, and channel coding*. Diss. University of Toronto, 1997.

[F.B.I. United States, 1985] United States. Federal Bureau of Investigation., The Science of fingerprints : classification and uses. Rev. 12-84. ed. 1985, Washington, D.C.: U.S. Dept. of Justice For sale by the Supt. of Docs., U.S. G.P.O. v, 211 p. 30

[Gamela, 2001] Gemela, Jozef. "Financial analysis using Bayesian networks." *Applied Stochastic Models in Business and Industry* 17.1 (2001): 57-67.

[Gamez & Moral, 2004] Gamez, Jose A., and Serafin Moral. *Advances in Bayesian networks*. New York: Springer, 2004.

[Geenen et al., 2006] Geenen, P. L., et al. "Development of a probabilistic network for clinical detection of classical swine fever." *Proceedings of the Eleventh Symposium of the International Society for Veterinary Epidemiology and Economics*. 2006.

[Geoffrey & Russell, 1998] Zweig, Geoffrey, and Stuart J. Russell. "Probabilistic modeling with Bayesian networks for automatic speech recognition." *ICSLP*. Vol. 98. 1998.

[Haraksim et al., 2013] Haraksim, Rudolf, et al. "Assignment of the evidential value of a fingermark general pattern using a Bayesian network." *Biometrics Special Interest Group (BIOSIG), 2013 International Conference of the*. IEEE, 2013.

[Heckerman, 1996] Heckerman, David. *A tutorial on learning with Bayesian networks*. Springer Berlin Heidelberg, 2008.

[Henry, 1900] E. R. Henry. *Classification and uses of fingerprints*. George Boutledge and Sons limited Broadway, London, 1900.

[Koehler, 2008] Koehler, Jonathan J. "Fingerprint error rates and proficiency tests: what they are and why they matter." *Hastings LJ* 59 (2007): 1077.

[Koller & Friedman, 2009] Kollar, Daphne, and Nir Friedman. *Probabilistic graphical models: principles and techniques*. The MIT Press, 2009.

[Korb & Nicholson, 2010] Korb, Kevin B., and Ann E. Nicholson. *Bayesian artificial intelligence*. CRC press, 2003.

[Kumar and Desai, 1996] Kumar, V. P., and Uday B. Desai. "Image interpretation using Bayesian networks." *Pattern Analysis and Machine Intelligence, IEEE Transactions on*18.1 (1996): 74-77.

[Kwisthout et al., 2010] Kwisthout, Johan, Hans L. Bodlaender, and Linda C. van der Gaag. "The Necessity of Bounded Treewidth for Efficient Inference in Bayesian Networks."*ECAI*. Vol. 215. 2010.

[Kwisthout, 2011] Kwisthout, Johan. "Most probable explanations in Bayesian networks: Complexity and tractability." *International Journal of Approximate Reasoning*52.9 (2011): 1452-1469.

[Lucas et al., 2000] Lucas, Peter JF, et al. "A probabilistic and decision-theoretic approach to the management of infectious disease at the ICU." *Artificial Intelligence in medicine*19.3 (2000): 251-279.

[Meuwly & Veldhuis, 2012] Meuwly, Didier, and Raymond Veldhuis. "Forensic biometrics: From two communities to one discipline." *Biometrics Special Interest Group (BIOSIG), 2012 BIOSIG-Proceedings of the International Conference of the*. IEEE, 2012.

[Neapolitan, 2004] Neapolitan, Richard E. *Learning bayesian networks*. Upper Saddle River: Pearson Prentice Hall, 2004.

[Neumann et al., 2011] Neumann, Cedric, et al. "Quantitative assessment of evidential weight for a fingerprint comparison I. Generalisation to the comparison of a mark with set of ten prints from a suspect." *Forensic science international* 207.1 (2011): 101-105.

[Nithin et al., 2009] Nithin, M. D., et al. "Study of fingerprint classification and their gender distribution among South Indian population." *Journal of Forensic and Legal Medicine* 16.8 (2009): 460-463.

[NRC Report, 2009] National Research Council, Strengthening Forensic Science in the United States: A Path Forward, National Academies Press, Washington, DC, 2009.

[Park & Darwiche, 2004] Park, James D., and Adnan Darwiche. "Complexity Results and Approximation Strategies for MAP Explanations." *J. Artif. Intell. Res.(JAIR)* 21 (2004): 101-133.

[Paul & Luby, 1993] Dagum, Paul, and Michael Luby. "Approximating probabilistic inference in Bayesian belief networks is NP-hard." *Artificial intelligence* 60.1 (1993): 141-153.

[Pearl, 1985] Pearl, Judea. "Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning”. *Proceedings of the 7th Conference of the Cognitive Science Society, University of California*,. 1985.

[Pearl, 1988] Pearl, Judea. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[Peng & Reggia, 1987] Peng, Yun, and James A. Reggia. *Abductive inference models for diagnostic problem-solving. Symbolic Computation*. Springer-Verlag New York, Inc, 1990.

[Pople, 1973] Pople, Harry E. "On the mechanization of abductive logic." *Proceedings of the 3rd international joint conference on Artificial intelligence*. Morgan Kaufmann Publishers Inc., 1973.

[Castro Ramos, 2007] Castro, Daniel Ramos. *Forensic evaluation of the evidence using automatic speaker recognition systems*. Diss. Universidad autónoma de Madrid, 2007.

[Reiter, 1987] Reiter, Raymond. "A theory of diagnosis from first principles." *Artificial intelligence* 32.1 (1987): 57-95.

[Robertson et al., 1995] Robertson, Bernard, G. Anthony Vignaux, and John Buckleton. *Interpreting evidence: evaluating forensic science in the courtroom*. New York: J. Wiley, 1995.

[Robinson, 1977] Robinson, Robert W. "Counting unlabeled acyclic digraphs." *Combinatorial mathematics V*. Springer Berlin Heidelberg, 1977. 28-43.

[Silander & Myllymaki, 2012] Silander, Tomi, and Petri Myllymaki. "A simple approach for finding the globally optimal Bayesian network structure." *arXiv preprint arXiv:1206.6875* (2012).

[Silander, 2012] Silander, Tomi. "The most probable Bayesian network and beyond." (2009).

[Stoney, 1991] Stoney, David A. "What made us ever think we could individualize using statistics?." *Journal of the Forensic Science Society* 31.2 (1991): 197-199.

[Thagard, 1989] Thagard, Paul. "Explanatory coherence." *Behavioral and Brain sciences* 12.3 (1989): 435-502.

[van der Gaag et al., 2002] van der Gaag, Linda C., et al. "Probabilities for a probabilistic network: a case study in oesophageal cancer." *Artificial Intelligence in medicine* 25.2 (2002): 123-148.

[van der Gaag & Renooij, 2003] van der Gaag, Linda C., and Silja Renooij. "Probabilistic networks as probabilistic forecasters." *Artificial Intelligence in Medicine*. Springer Berlin Heidelberg, 2003. 294-298.

[van Leeuwen & Brummer, 2007] van Leeuwen, David A., and Niko Brümmer. "An introduction to application-independent evaluation of speaker recognition systems." *Speaker Classification I*. Springer Berlin Heidelberg, 2007. 330-353.

[Yuille & Kersten, 2006] Yuille, Alan, and Daniel Kersten. "Vision as Bayesian inference: analysis by synthesis?." *Trends in cognitive sciences* 10.7 (2006): 301-308.

[Zhang & Poole, 1996] Zhang, Nevin Lianwen, and David Poole. "Exploiting causal independence in Bayesian network inference." *arXiv preprint cs/9612101* (1996).

Appendix A: Data selection and refinement of Police fingerprint dataset

Few years ago, the dataset at police consisted of around 1,100,000 ten-print cards of criminals. The data from 772,577 individuals having a Dutch nationality were handed over to the NFI for research purposes. For security reasons the data of international criminals were not handed over. This data had been encoded in an internal security based format.

For a previous project, the data of 312,484 individuals from the dataset were manually checked by fingerprint examiners at the police [Doekhie, 2012].

- 1,477 entries were duplicates [Ids: 10736929-10738473]
- For 3,621 individuals all the fingerprint information was missing. Whereas for an additional 1,281 individuals part of the fingerprint information was missing. (By missing information we mean that there was no corresponding fingerprint image present in the dataset or the image was without a fingerprint).

For the current version of the dataset we remover all duplicate entries and all individuals with incomplete information. And hence finally the refined dataset contained information from 306,105 individuals.

Appendix B: Contingency tables showing the distribution of general pattern on males and females per finger

		General Pattern on Finger 1						
		Arch	Left_Loop	Right_Loop	Whorl	Tented _Arch	Un classifiable	Total per gender
Gender	Male	4803	856	104718	107852	2521	432	221182
	Female	3421	326	45475	32301	1238	104	82865
	Total per GP	8224	1182	150193	140153	3759	536	**304047**

		General Pattern on Finger 2						
		Arch	Left_Loop	Right_Loop	Whorl	Tented _Arch	Un classifiable	Total per gender
Gender	Male	11751	36297	65472	80401	26102	1159	221182
	Female	6464	9978	30301	27336	8537	249	82865
	Total per GP	18215	46275	95773	107737	34639	1408	**304047**

		General Pattern on Finger 3						
		Arch	Left_Loop	Right_Loop	Whorl	Tented _Arch	Un classifiable	Total per gender
Gender	Male	8172	3072	145214	47701	16166	857	221182
	Female	4170	627	60103	12381	5410	174	82865
	Total per GP	12342	3699	205317	60082	21576	1031	**304047**

		General Pattern on Finger 4						
		Arch	**Left_Loop**	**Right_Loop**	**Whorl**	**Tented _Arch**	**Un classifiable**	**Total per gender**
Gender	**Male**	2155	2106	97156	112564	6547	654	221182
	Female	1383	712	44013	34059	2529	169	82865
	Total per GP	3538	2818	141169	146623	9076	823	**304047**

		General Pattern on Finger 5						
		Arch	**Left_Loop**	**Right_Loop**	**Whorl**	**Tented _Arch**	**Un classifiable**	**Total per gender**
Gender	**Male**	1319	543	171899	39860	6713	848	221182
	Female	1100	169	68902	9485	2935	274	82865
	Total per GP	2419	712	240801	49345	9648	1122	**304047**

		General Pattern on Finger 6						
		Arch	**Left_Loop**	**Right_Loop**	**Whorl**	**Tented _Arch**	**Un classifiable**	**Total per gender**
Gender	**Male**	8581	122449	1097	84931	3626	498	221182
	Female	5388	45614	710	29373	1635	145	82865
	Total per GP	13969	168063	1807	114304	5261	643	**304047**

		General Pattern on Finger 7						
		Arch	**Left_Loop**	**Right_Loop**	**Whorl**	**Tented _Arch**	**Un classifiable**	**Total per gender**
Gender	**Male**	11609	75522	31667	74811	26500	1073	221182
	Female	6715	27307	13131	25983	9484	245	82865
	Total per GP	18324	102829	44798	100794	35984	1318	**304047**

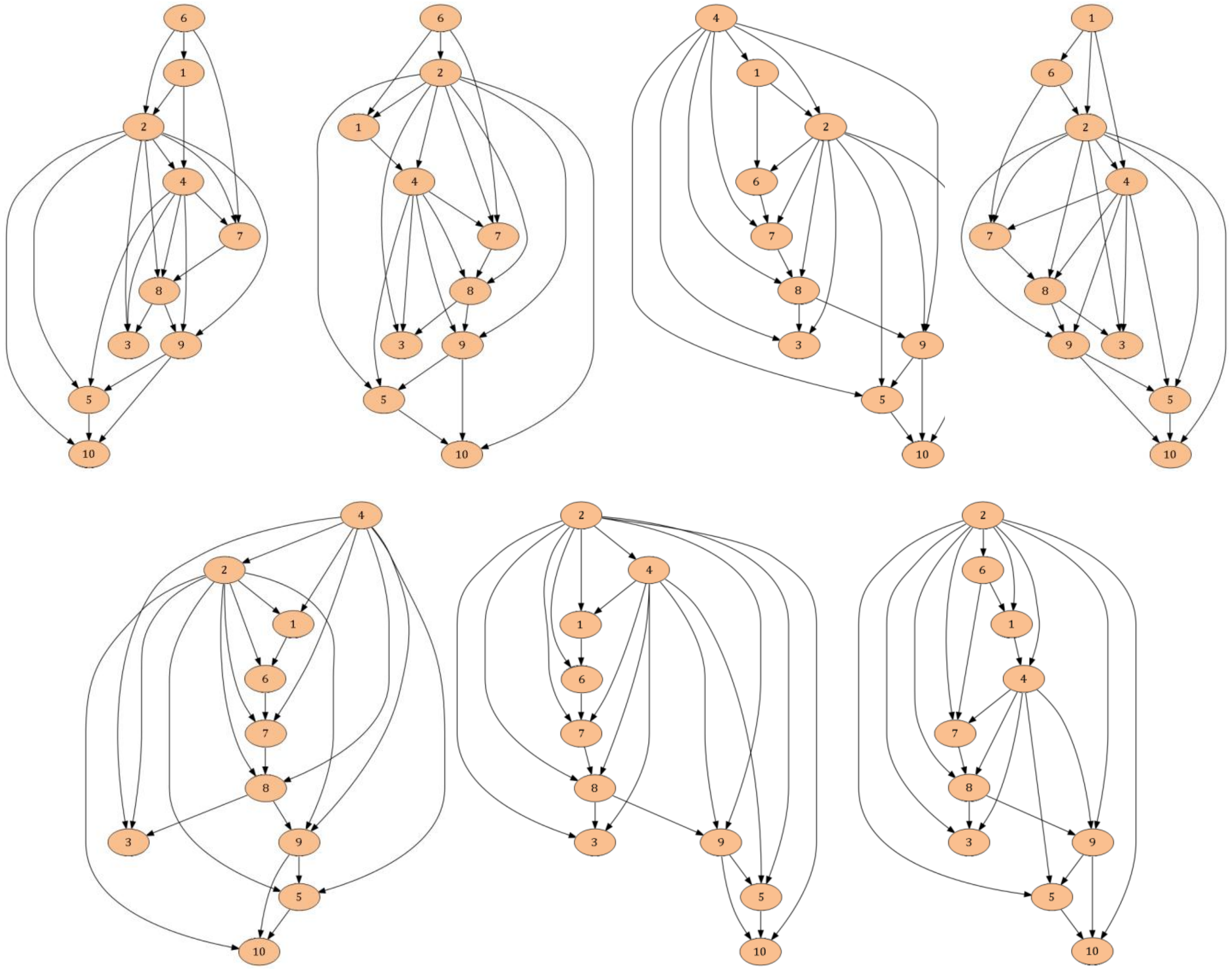
		General Pattern on Finger 8						
		Arch	Left_Loop	Right_Loop	Whorl	Tented _Arch	Un classifiable	Total per gender
Gender	Male	10151	143399	2386	47455	16970	821	221182
	Female	6131	53260	1232	15196	6859	187	82865
	Total per GP	16282	196659	3618	62651	23829	1008	**304047**

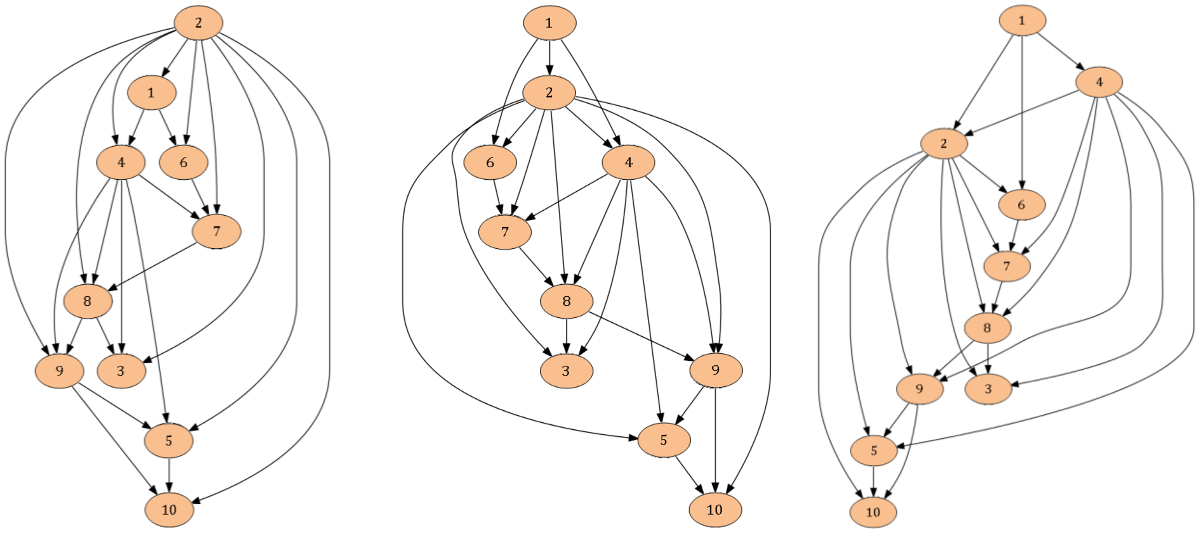
		General Pattern on Finger 9						
		Arch	Left_Loop	Right_Loop	Whorl	Tented _Arch	Un classifiable	Total per gender
Gender	Male	2955	122223	847	87686	6793	678	221182
	Female	1909	47673	641	29412	3031	199	82865
	Total per GP	4864	169896	1488	117098	9824	877	**304047**

		General Pattern on Finger 10						
		Arch	Left_Loop	Right_Loop	Whorl	Tented _Arch	Un classifiable	Total per gender
Gender	Male	1727	181461	304	29826	6943	921	221182
	Female	1538	68144	145	9318	3352	368	82865
	Total per GP	3265	249605	449	39144	10295	1289	**304047**

Appendix C: 10 best equivalent networks

These following 10 best networks were learnt from the fingerprint dataset using the algorithm by [Silander & Myllymaki, 2012] as discussed in Section 5.1. These networks are coincidently in the same equivalent class. The variables general pattern on fingerX has been denoted by only X in the following graphical structures (for examples general pattern on finger1 is denoted by only 1).





Structural Learning results: 10 best networks

Acknowledgement

The research was motivated by the work carried out at the University of Lausanne, presented by Prof. dr. Christophe Champod at the ENFSI WG 2010 entitled *"The Use of Probabilistic Networks in the Area of Fingerprints"*.

My first and sincere appreciation goes to Dr. Didier Meuwly and Mr. Rudolf Haraksim, my supervisors at the Netherlands Forensic Institute for giving me an opportunity to work on this project. Forensic science was a completely new domain for me, but it was not difficult with their continuous help, guidance and support in all stages of this thesis. I would also like to thank him for being open to ideas, and for encouraging and helping me to shape them.

I would like to express my deep gratitude and respect to my supervisor Dr. Silja Renooij from Utrecht University, who agreed to supervise me, despite all odds. Her advices and insight were invaluable. Especially in the computer science part of this project, it was her guidance and enthusiasm, which helped us design new heuristics.

I would also like to thank my study advisor Dr. Ad Feelders, who was a continuous source of support. Moreover I could leverage on his experience on learning of Bayesian Networks. I would like to thank Prof. dr. Linda van der Gaag, for examining and reviewing my thesis.

I also express my deep gratitude to the fingerprint experts at NFI, Mr. Anko Lubach and Ms. Sheryl Lie Kwie for their invaluable expert advice. I am also grateful to Prof. dr. Arian van Asten, Dr. Peter Vergeer and Prof. dr. Charles Berger for their guidance and support.

Moreover finally I express my gratitude towards Utrecht University and Utrecht University Fonds for funding my MSc. education.