Forensic speaker recognition

Based on text analysis of transcribed speech fragments

Nelleke Scheijen

Delft University of Technology

Forensic speaker recognition

Based on text analysis of transcribed speech fragments

by

Nelleke Scheijen

to obtain the degree of
Master of Science in Applied Mathematics
Specialisation Stochastics

at the Delft University of Technology,
to be defended publicly on Wednesday June 24, 2020 at 14:00.

Student number:	4323874	
Project duration:	Oktober 7, 2019 – June 24, 2020	
Thesis committee:	Dr. J. Söhl,	TU Delft, supervisor
	Dr. ir. M. Keijzer,	TU Delft
	Prof. dr. ir. G. Jongbloed,	TU Delft
	A. J. Leegwater, MSc.	NFI, daily supervisor

Nederlands Forensisch Instituut
Ministerie van Justitie en Veiligheid

TU Delft

Abstract

Speaker recognition is an important subject and a constantly developing field in forensic science. Currently, speaker recognition research is mainly based on phonetics and speech signal processing. This research addresses speaker recognition from a new perspective, analysing the transcription of a fragment of speech with text analysis methods. Since text analysis is based on the transcription text only, it can be assumed independent from current automatic speaker recognition software. Hence, it would contribute significantly to the overall evidential value. The analysis is based on the frequencies of non-content, highly frequent words. We study whether information about the identity of the speaker is contained in the transcription of spoken text.

The value of evidence is quantified using a score-based likelihood ratio. The score-based approach is chosen because in most forensic cases, there is not enough data from the suspect or of the disputed speech fragment available to model a robust feature-based likelihood ratio. Different methods to model the system from feature vector over score to likelihood ratio have been compared. As a baseline, a distance based method is used, where the score is the distance between the feature vectors. To improve upon this baseline, machine learning algorithms are implemented. The results from SVM and XGBoost are explored. As a third method a feature-based likelihood ratio is calculated and used as a score instead of as a direct likelihood ratio. With this method, both similarity and typicality are taken into account.

The model is trained and tested on the FRIDA data set from the Netherlands Forensic Institute, consisting of Dutch conversations from a homogeneous group of 250 individuals. The performance of the likelihood ratio system is evaluated through computing the cost log-likelihood-ratio C_{llr}, which is a measure for the accuracy and quality of the likelihood ratios, and the accuracy A of the likelihood ratios solely. The performance is also evaluated by inspecting the Tippett, empirical cross-entropy and pool-adjacent-violators plots. Different values for parameters used in the calculation of the likelihood ratios are investigated: the length of the sample N, the number of frequent words (number of features) $F_{\#}$ and the number of samples $S_{\#}$ needed to train the model.

The distance method showed a strong baseline, with good performance for large sample lengths. The SVM method outperformed the distance method for all parameter settings, with a peak performance of $A = 0.94$ and $C_{llr} = 0.24$. The XGBoost method showed promising results for smaller samples lengths, but a too large amount of data is needed to obtain good performance for larger sample lengths. The LR score method showed moderate results, but no improvements due to the necessity to estimate high-dimensional distributions.

This thesis shows that information about the identity of the speaker is contained in transcriptions of speech. The complete process from data to likelihood ratio is constructed, where the likelihood ratio quantifies the evidential value of a transcribed speech fragment.

Preface

With handing in this thesis, my life as a student comes to an end. This thesis is the final requirement to obtain my Master's degree in Applied Mathematics in the specialisation Stochastics. During the last 9 months, I worked on the subject of forensic speaker recognition, based on text analysis of transcribed speech fragments. I am really glad to have studied this topic, as it perfectly fits my idea of an interesting mathematical problem with a clear application in real life. I had no previous experience with forensic science and learned a lot about quantifying evidence as well as machine learning and programming. It sparked my enthusiasm about forensic science and I appreciate the opportunity to have gained knowledge about this research area. Without my supervisors, friends and family around me, this thesis would not have been as it is now. Therefore, I would like to thank some people in particular.

First of all I would like to thank my supervisor Jakob Söhl from TU Delft. For helping me plan my thesis, always supportive advise and small chats about rowing during our two-weekly meetings. You helped me finishing my thesis in time, during corona and a small stress moment in the last two weeks. I also would like to thank Marleen Keijzer and Geurt Jongbloed, for taking place in my graduation committee.

Secondly, I would like to thank my daily supervisor Jeannette Leegwater from the NFI. Our weekly meetings were always a good mix between discussing research ideas, my progress and chatting about bouldering or something else. You helped me writing a clear thesis about the research I conducted. I am grateful for the opportunity to have carried out this research at the NFI. I had a great time with all my colleagues. They helped me with interesting suggestions for my thesis, but we also planned fun activities as bouldering and running breaks. Thank you Wauter and David, for your guidance in the forensic speaker recognition research.

A special thanks goes to my family. My parents, Rob & Noor, Bibi & Thei, for always supporting me during my entire time as a student and in every decision I made. For celebrating great times and offering help when I needed it. To my siblings Amy & Coen and my twin sister Guusje, for just always being there for me.

Lastly, I would like to thank all the people who made my student life as exciting as it was. My rowing teammates, with whom I spend almost all my time at the rowing association with, during my bachelor years. Thank you for all the sporty adventures and tea drinking sessions. My board members, with whom I took a gap year and who turned into close friends. Thank you for supporting me in good, but also stressful times. My flatmates, who were always caring and helpful when I needed a cheer-up. My study friends from the bachelor, but also new study friends from the master. You guys made studying till late hours that much more fun. Chris, for the sweet unconditional support during my thesis. And just to everyone I met on the way, thank you!

Nelleke Scheijen
Delft, June 2020

Contents

1 Introduction

Assume a person, suspected by the police of a criminal act. The police places a tap on their mobile phone to investigate the matter. On one of the tapped recordings, the suspect speaks about a criminal act. The tapped recordings can be used as evidence by the police. After this event the suspect is arrested. However, the person claims the phone was stolen and thus the person on the tap is not them. The police then tasks a forensic scientist to look at the value of evidence to research if the (unknown) speaker can be identified by information deduced from the telephone conversation. Can the telephone conversation be used as evidence by comparing the suspect's manner of speaking and the conversation on the phone? In forensic science, this type of research is called speaker recognition. Two approaches are possible for speaker recognition, the first one is to determine the speaker of a disputed fragment of speech by comparing the disputed fragment with speech from the suspect. The second approach is to compare two fragments of speech from unknown speaker(s), to determine if they originate from one speaker or two different speakers [34, 35]. This approach could be used to research whether two tapped recordings point to one or two suspects, even if the person is still unknown.

Speaker recognition is an important subject and a constantly developing field in forensic science. At the Netherlands Forensic Institute (NFI), where this thesis is conducted, the speech department currently uses two methods to identify the speaker of a speech fragment. These methods are the automatic voice recognition tools and the judgement of an expert [59]. Currently, a drawback of the automatic voice recognition system is its dependency on the recording situation. For different background noises, e.g. a telephone tap and a car tap, the system does not provide consistent results [28]. The judgement of an expert is performed by an individual and is a human judgement. This is a valuable analysis, but also subjective as the measurement metric is a human [59]. This research attempts to address automatic speaker recognition from a new perspective, by analysing the transcription from a fragment of speech. A new method based on text analysis could contribute to the already existing methods named before. Text analysis for authorship analysis on written text has already been a research topic for over 50 years and has promising results for applications like plagiarism detection or authorship analysis of novels [3, 4, 32]. It has been shown that authors unknowingly use specific words or sentence structures distinctively in all types of written text [15, 21]. This raises the question if a transcription from speech also contains a certain amount of information about the speaker, contained in the choice of words while having a conversation.

In order to determine the value of evidence of a speech fragment, a forensic scientist has to quantify this value. For this a universal framework to evaluate evidence is needed. The value is determined by comparing the evidence conditional on two competing hypotheses. The hypotheses are mutually exclusive and are commonly denoted as the prosecutor's hypothesis H_p and the defence hypothesis H_d. The ratio of the probabilities of the evidence conditional on these hypotheses results in the likelihood ratio quantifying the value of evidence. The choice of hypotheses is an important part of forensic evidence evaluation. To use speaker recognition based on text analysis in a forensic context, the result has to be stated in the form of a likelihood ratio, rather than just as a correct classification of the speaker. In a legal case a judge is interested in a quantitative judgement about the evidence: to what extent does the evidence support one of the hypotheses with respect to the other hypothesis? The framework to quantify the evidence is further outlined in Chapter 2. We research a problem where one is interested in the source of a trace, the so-called identification of source problem [34, 35]. In our case the forensic scientist is interested in the speaker of the fragment of speech, i.e. the source of the speech fragment.

At the moment of writing this thesis, to the best of the author's knowledge, no publications are available on authorship analysis on the transcriptions of spontaneous spoken text. Therefore, this thesis explores a new application. Text analysis can be assumed independent from current automatic speaker recognition software, which is mainly based on phonetics and speech signal processing. It thus would contribute significantly to the overall evidential value of speaker recognition. Especially for comparing fragments of speech with different background noise, text analysis of the transcription can be a suitable option to explore.

To summarise, the scope of this research is to quantify the evidential value in the transcription of spoken text in a forensic context. The theoretical statistical forensic framework is explained, taking into account the limitations present in a real forensic case as in the example stated above. The text analysis methods for authorship analysis are implemented and tested for automatic speaker recognition. The forensic framework and the text analysis methods are combined to cover the complete process and validation from the transcription of a fragment of speech to a value of evidence in terms of a likelihood ratio. The research is conducted on a data set of transcriptions of speech fragments, provided by the NFI.

1.1. Thesis outline

To cover this research scope, the following structure is used throughout this thesis. First, the statistical frameworks for the identification of source question are outlined in Chapter 2, based on work from [34, 35]. Two common used frameworks are explained, the common source and the specific source framework [35]. The competing hypotheses are specified, which is an important part in quantifying the value of evidence. Using the hypotheses outlined, the likelihood ratio framework is explained in Chapter 3. The value of evidence is presented, and the framework is explained for a direct and a score-based speaker recognition method. The chapter concludes with outlining how evidence can be combined. To build the complete likelihood ratio system, a data set with transcriptions of speech is needed. The used data set, provided by the NFI, is introduced in Chapter 4. The data is explored, after which the strengths and weaknesses of the data set are outlined. In the chapters that follow, the complete theoretical framework from input data, as transcription of speech fragments, to validated likelihood ratios is explained. Figure 1.1 shows the process from start to end.

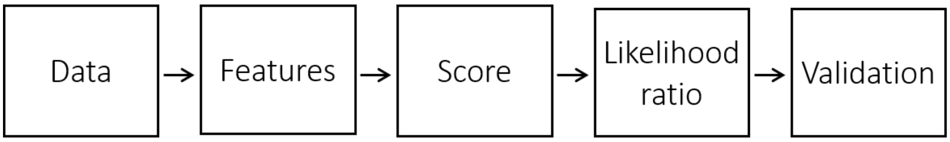


Figure 1.1: Overview of the structure of this thesis.

Chapter 5 till 8 are also structured according to the transitions between these steps. Chapter 5 explains the different features and frameworks used in text analysis for authorship analysis. It shows the large variation in used features for different research of authorship analysis. Furthermore it specifies the feature set of this research for extracting a numerical feature vector from a transcript. Chapter 6 describes the algorithms and methods to derive the classification scores from the feature vectors, based on the competing hypotheses. To obtain the classification scores, three methods are applied. As a baseline, a method based on a distance metric is used, where the score is the distance between the feature vectors. To improve upon this baseline, several machine learning algorithms are applied to calculate a score from the feature vectors [38, 50]. These include support vector machines and XGBoost, a decision tree based algorithm. As a third method a score likelihood ratio is calculated under several assumptions based on the feature vector. It is used as score instead of a direct likelihood ratio. After obtaining the classification scores, in Chapter 7 the calibration method to transform the scores to likelihood ratios is introduced. To conclude the theoretical method, a validation framework is defined in Chapter 8, presenting different performance characteristics and the corresponding performance metrics. The complete theoretical method is applied on the data set provided by the NFI. The results are presented in Chapter 9. The results are presented separately for the respective methods, i.e. the distance method, the machine learning techniques and the score likelihood ratio approach. Finally, the conclusion and recommendations from this thesis are presented in Chapter 10.

2 Identification of source framework

In this chapter a literature study is presented, mainly based on [34–36]. The purpose of this literature study is to present the framework and theory which is needed to understand the remainder of this thesis.

As outlined in the introduction, a forensic scientist is tasked to quantify the value of evidence. The value of evidence is often defined as a ratio between two competing hypotheses, which is called the likelihood ratio, first introduced by [13]. Typically, a hypothesis of the prosecution, denoted H_p, and a hypothesis of the defence, denoted H_d, are stated. The ratio between the probabilities of the evidence given the hypotheses is called the likelihood ratio. The exact formulation of the hypotheses depends on the type of identification of source model. Two identification of source approaches will be considered, the common source model and the specific source model, as specified by [34, 35]. The specific source model tries to answer the question whether a speech fragment originates from a specific known source, where the common source problem is focused on determining if two texts originate from the same unknown source. The two frameworks look quite similar and can be interpreted in the same context, but the method to approach both problems could lead to different results [34, 35]. Both models give rise to different pairs of competing hypotheses, these will be discussed below.

In forensic science, scientists are interested in the way the evidence is generated, instead of only classical hypotheses testing of the parameters [34, 35]. Thus, the hypotheses include the choice for a sampling model. In classical testing, the sampling model is specified beforehand, in forensic science the sampling model differs for the specific source and common source model. Therefore, to do hypothesis testing in forensic science, the following components are needed, as specified by [34, 35]. First, the sampling models corresponding to the competing hypotheses have to be stated. Secondly, the corresponding parametric distribution used in the sampling models has to be estimated. Theoretically a non-parametric distribution is also possible, but infeasible because of the high dimensional feature vector. The number of samples needed to fit a non-parametric distribution accurately, increases exponentially with number of dimensions [24]. It has to be noted here, that it has to be assumed the data follows a parametric distribution [11]. In many cases, as also in speaker recognition, picking a correct parametric distribution can be quite hard or even impossible due to the unknown distribution and high-dimensional data. This is further outlined in Section 3.2. Lastly, to use the parametric distribution, the parameters have to be estimated. In this section only the sampling models are presented, the parametric distribution with corresponding parameters is outlined in Chapter 6.

In the remainder of this chapter three sets of evidence are used. Evidence from a specific known source e_s, evidence from a unknown source e_u and evidence from a large set of known sources often referred to as the background population of alternative sources e_a, following the notation of [34, 35]. This background set should be relevant for the hypotheses that are tested. With relevant we mean the sources should be comparable with respect to the background information a forensic scientist has about the sources under investigation. Which evidence sets are used per framework will be further outlined below. It is important to note that the choice of background material can have a large impact on the results, it has to represent the total alternative population of speakers.

The common source and specific source sampling models are briefly discussed, to explain the frameworks for the application in this thesis. The models are introduced and detailed outlined in [34, 35].

2.1. Common source

The classical common source problem starts with two traces. Suppose two traces of unknown origin are found at two different crime scenes. It is interesting whether the crimes can be linked together. One is interested if the traces come from the same unknown source. The two different traces lead to two pieces of evidence with unknown source: e_{u_1} and e_{u_2}. One is not researching who is the unknown source, but only if both traces originate from the same unknown source [34, 35]. In terms of speaker recognition, assume having two different speech fragments from unknown speakers (e_{u_1},e_{u_2}). The question of interest is, whether both fragments originate from the same speaker. The evidence set is completed with material from the background population (e_a), the background population of alternative sources, and defined as $e = \{e_{u_1}, e_{u_2}, e_a\}$. For the common source problem, the hypotheses in the context of speaker recognition are stated as [34, 35]:

$\mathbf{H_p}$**:** The two speech fragments (e_{u_1}, e_{u_2}) from unknown speakers originate from the same unknown speaker.

$\mathbf{H_d}$**:** The two speech fragments (e_{u_1}, e_{u_2}) from unknown speakers originate from two different unknown speakers.

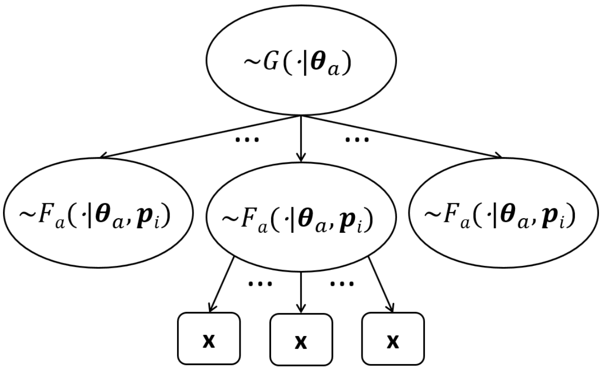
To evaluate the transcription in the statistical framework, first a numerical feature vector has to be deduced from the transcription. This can be achieved by using relevant features, which will be described in Chapter 5. Assume we have l features, define n as the number of different speakers in the background population and m the number of samples per speaker. The number of samples m can vary because some speakers will have a large speech fragment, which can be divided, resulting in several speech fragments. Using the background population data set containing a relevant subset of known speakers, samples of fragments of speech can be extracted. Assuming the background population has a sufficient amount of data, per speaker several samples m, with l features can be extracted. Assuming the feature vectors obtained from the data follow a certain parametric distribution, two distributions can be specified. First the between-source distribution as

$$G(\cdot|\boldsymbol{\theta}_a), \tag{2.1}$$

where $\boldsymbol{\theta}_a$ is the l-dimensional parameter that indexes the l-dimensional between-source distribution. define $\mathbf{P}_i$ as the sources in the background population of alternative sources, the sources are sampled according to the between-source distribution. Secondly, the within-source distribution given a specific source $\mathbf{P}_i = \mathbf{p}_i$ from the background population, is defined as

$$F_i\left(\cdot|\boldsymbol{\theta}_a, \mathbf{p}_i\right), \tag{2.2}$$

where $\boldsymbol{\theta}_a$ is the l-dimensional parameter that indexes the l-dimensional within-source distribution. Figure 2.1 shows the hierarchical structure of the within-source and in between-source distribution, the obtained samples are denoted as x. In the speaker recognition context, these distributions are referred to as the between-speaker and within-speaker distributions.



$\sim G(\cdot|\boldsymbol{\theta}_a)$
$\sim F_a(\cdot|\boldsymbol{\theta}_a, \boldsymbol{p}_i)$
$\sim F_a(\cdot|\boldsymbol{\theta}_a, \boldsymbol{p}_i)$
$\sim F_a(\cdot|\boldsymbol{\theta}_a, \boldsymbol{p}_i)$
x
x
x

Figure 2.1: Schematic view of the between-source distribution and the within-source distribution, based on [31] (Fig 2, p. 48).

Define m_{u_1} and m_{u_2} as the number of samples obtained from the unknown speaker(s). The hypotheses for the common source framework give rise to two different competing sampling models, the prosecution sampling model M_p and the defence sampling model M_d [34, 35]:

M$_\mathbf{a}$: The background evidence from all alternative known speakers e_a is generated by sampling n speakers from the background population and sampling m samples from every speaker.

M$_\mathbf{p}$: The evidence from the two speech fragments from the unknown speakers, e_{u_1} and e_{u_2}, is generated by first randomly sampling one speaker from the background population and then sampling $m_{u_1} + m_{u_2}$ samples from the same speaker.

M$_\mathbf{d}$: The evidence from the two speech fragments from the unknown speakers, e_{u_1} and e_{u_2}, is generated by first randomly sampling two sources from the background population and then sampling m_{u_1} samples from one source and sampling m_{u_2} samples from the other source.

The schematic sampling model is illustrated in Figure 2.2, a sample is denoted as $x_{\mathrm{p,j}}$, which corresponds to the j^{th} sample of speaker p. Note that the two hypotheses agree on the background sampling model M_a. The blue and orange box, top left and top right respectively, represent the competing sampling models M_p and M_d about the evidence [36]. The prosecution argues the evidence from the two unknown speech fragments, e_{u_1} and e_{u_2}, is generated by one speaker, but the defence argues the evidence is generated by two different speakers from the background population.

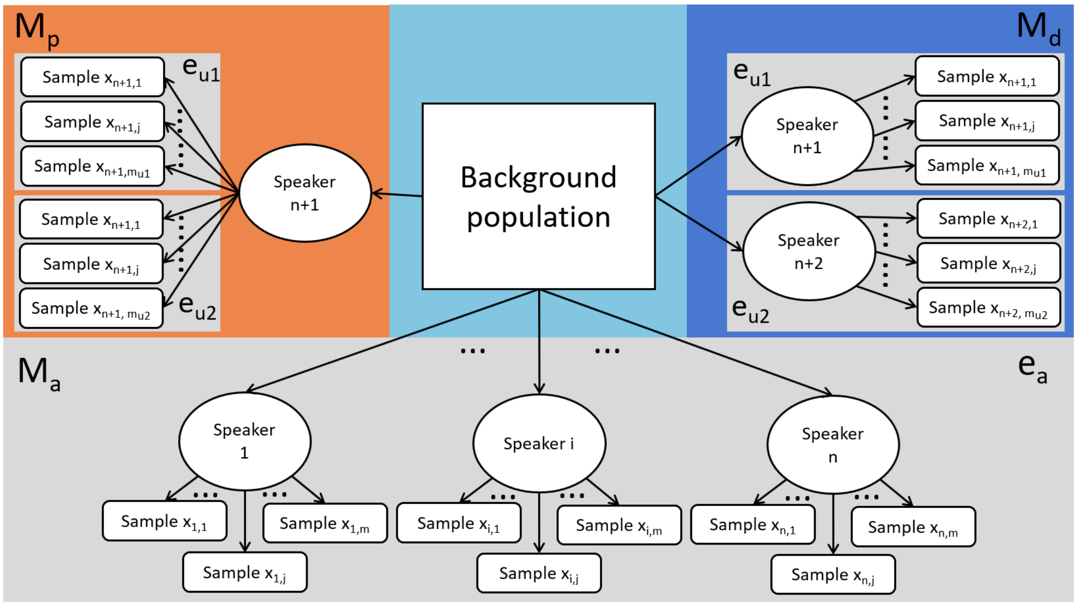

$\mathrm{M_p}$
Sample $\mathrm{x_{n+1,1}}$
Sample $\mathrm{x_{n+1,j}}$
Sample $\mathrm{x_{n+1,mu1}}$
$\mathrm{e_{u1}}$
Sample $\mathrm{x_{n+1,1}}$
Sample $\mathrm{x_{n+1,j}}$
Sample $\mathrm{x_{n+1,\ mu2}}$
$\mathrm{e_{u2}}$
Speaker n+1
Background population
$\mathrm{M_d}$
$\mathrm{e_{u1}}$
Speaker n+1
Sample $\mathrm{x_{n+1,1}}$
Sample $\mathrm{x_{n+1,j}}$
Sample $\mathrm{x_{n+1,\ mu1}}$
Speaker n+2
Sample $\mathrm{x_{n+2,1}}$
Sample $\mathrm{x_{n+2,j}}$
Sample $\mathrm{x_{n+2,\ mu2}}$
$\mathrm{e_{u2}}$
$\mathrm{M_a}$
$\mathrm{e_a}$
Speaker 1
Speaker i
Speaker n
Sample $\mathrm{x_{1,1}}$
Sample $\mathrm{x_{1,m}}$
Sample $\mathrm{x_{1,j}}$
Sample $\mathrm{x_{i,1}}$
Sample $\mathrm{x_{i,m}}$
Sample $\mathrm{x_{i,j}}$
Sample $\mathrm{x_{n,1}}$
Sample $\mathrm{x_{n,m}}$
Sample $\mathrm{x_{n,j}}$

Figure 2.2: Overview of the sampling model for the common source model. The orange and blue box represent the two competing sampling models M_p and M_d, top left and top right respectively. The schematic overview is based on [36] (Fig. 1, p. 4).

2.2. Specific source

The classical specific source problem is stated the following way: a trace is found at a crime scene and the question is whether a specific suspected source can be linked to this trace [34, 35]. In other research this model is applied to the question of source problem for glass fragments, fingermarks and DNA [37]. From the trace, several measurements are taken and this results in the unknown source evidence e_u. From the suspected source measurements are taken and this results in the specific source evidence e_s. In terms of speaker verification, assume having one speech fragment from an unknown speaker (e_u) and some speech fragments from a known specific speaker (e_s). The question of interest is, whether the unknown speech fragment originates from the known specific speaker. To complete the evidence set for the specific source model, material from the background population is needed (e_a). The complete set of evidence is defined as $e = \{e_u, e_s, e_a\}$. The competing hypotheses for the specific source problem, applied to speaker verification, are specified the following way [34, 35]:

H$_\mathbf{p}$: The speech fragment from an unknown speaker (e_u) and speech fragment from a known specific speaker (e_s) originate from the same known specific speaker.

$\mathbf{H_d}$: The speech fragment from an unknown speaker (e_u) does not originate from the known specific speaker, but from an alternative speaker.

Again the speech fragments are transformed to numerical feature vectors with l features. Define n as the number of different speakers in the background population and m the number of samples per speaker. Assume the specific speaker data gives rise to m_s l-dimensional samples. The unknown speech fragment results in m_u l-dimensional feature vectors. For the specific source model, an additional sampling model is added, the specific source sampling model M_s, followed by the background population sampling model and the two competing sampling models M_p and M_d [34, 35]:

$\mathbf{M_s}$: The evidence e_s from the known specific speaker is generated by sampling m_s samples from the specific speaker.

$\mathbf{M_a}$: The background evidence e_a from all alternative known speakers is generated by sampling n speakers from the background population and sampling m samples from every speaker.

$\mathbf{M_p}$: The evidence e_u from the speech fragment from unknown speaker is generated by sampling m_u samples from the specific speaker.

$\mathbf{M_d}$: The evidence e_u from the speech fragment from unknown speaker is generated by sampling one source from the background population and sampling m_u samples from the source.

Figure 2.3 shows the schematic overview of the specific source sampling model. The orange and blue box represent again the two competing sampling models [36]. Note that for the specific source model two data sets are needed, the specific speaker data set and the background population data set.

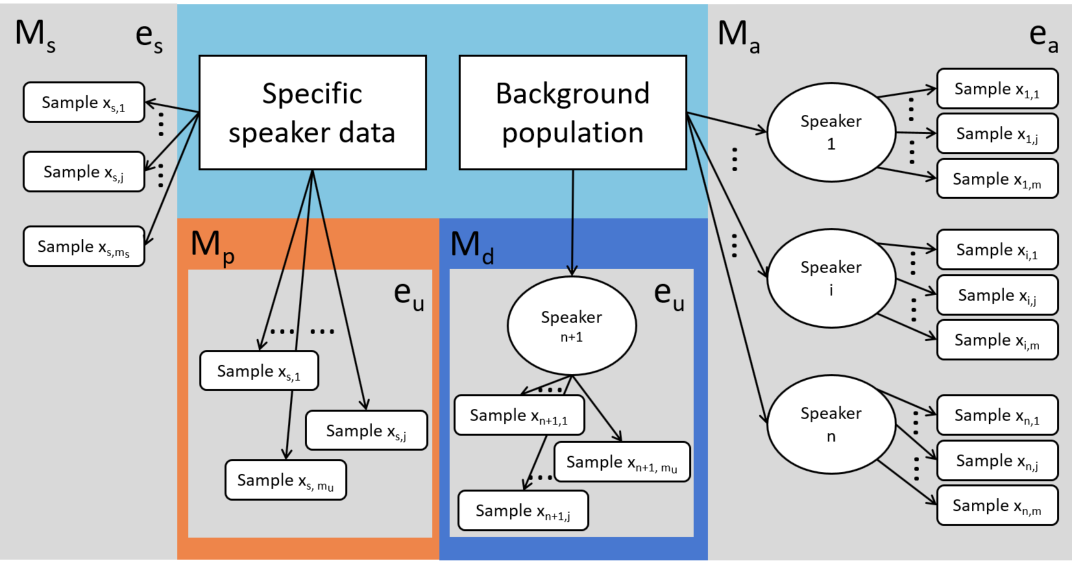


Figure 2.3: Overview of the sampling models for the specific source model. The orange and blue box represent the two competing sampling models M_p and M_d, bottom middle left and and bottom middle right respectively. The schematic overview is based on [36] (Fig. 1, p. 5).

2.3. Comparison frameworks for the speaker recognition case study

It is an ongoing discussion which of the two frameworks should be used in forensic science. In this section, the frameworks are compared taking into account the following: the question of interest in the legal case, the necessary data from the background population and specific source, and the suitability for the forensic case study in this thesis.

Often in the case of speaker recognition in court, a specific known suspect is investigated. The judge is interested whether this person is indeed the person who speaks on the disputed fragment of speech. Therefore it could be argued the specific source framework should be used. The set of competing hypotheses matches better with the question of interest. It can be argued the common source model is better suitable for an ongoing investigation, for example whether two speech fragments from different crimes can be linked together.

However, for specific source modelling, sufficient data from the specific source has to be available to train a model for the specific speaker according to the sampling model. The difference between the common source model and specific source model is sometimes referred to as the difference between using one or two background populations data sets, respectively. For the common source model one background population data set is needed. For the specific source model a background population data set and a specific source data set is needed [35]. For some applications in forensic science, for example glass, it is possible to take more samples from the disputed glass fragment, from which the specific source model can be constructed. For speaker recognition, the data of the specific speaker in question can be limited and it can be hard to collect more data. The suspect in an investigation, the specific speaker, is probable to not talk in a natural way during the interrogation or can be not cooperative. Often there is not enough transcribed speech available to model a specific source model from the suspect speaker. In some cases only one speech fragment from the specific speaker and one disputed fragment of speech is available. This suggests the common source model is more applicable to use in a real forensic case, where data is limited [34]. Common source modelling tends to be more conservative, because it contains an extra uncertainty.

The speech department at the NFI is interested what evidential value can be calculated in the case where the disputed speech fragment(s) and the speech fragments from a suspect or unknown person are compared. Often the disputed speech fragment(s) is used as one sample, as this is the limiting factor in the research. The background population can be divided according to the size of the disputed fragment. This sets the parameters $m_{u_1}, m_{u_2}, m_u = 1$. The parameter m_s, the number of specific speaker samples, is in most speech comparison cases small. This is important in the remainder of this thesis. The question is whether an evidential value can be obtained with small amounts of data from the specific source and how much background sources are needed.

For both frameworks several advantages and disadvantages have been mentioned. In the application this thesis is used for, the critical factor is the lack of a sufficient large data set from the specific speaker in a real forensic case. Therefore first the common source method is explored, to look whether results can be obtained from using only a background population data set. The specific source model is implemented in one method, the score likelihood ratio, to research whether it is possible to model with a small specific speaker data set.

3
Likelihood ratio framework

This chapter contains an introduction to the likelihood ratio framework. First, the definition of the value of evidence is introduced. After this, the score-based approach is outlined and the advantages and disadvantages are stated. The chapter concludes with the verbal equivalent of the calculated likelihood ratios and theory about the combination of evidence.

3.1. Value of evidence

Forensic scientists are tasked to quantify the value of evidence. Given two hypotheses as defined in the previous section, the question is to what extent the evidence supports one hypothesis over the other hypothesis. It is important to quantify the value of evidence objectively – a clear, universal framework is needed. The likelihood ratio framework to quantify the value of evidence is internationally accepted by forensic scientists and was first introduced in 1970 [13, 60]. The likelihood ratio is based on applying the Bayes theorem. The Bayes theorem describes how to update the probability of an event given new information.

In almost every legal case, scientific and non-scientific evidence is present [24]. Examples of non-scientific evidence are the testimony of a witness, a motive or an alibi. Scientific evidence includes, for example, DNA, patterns in traces and also the fragments of speech used in this thesis. Scientific evidence is based on empirical data. In a legal process, first the legal experts (e.g. a judge) have to determine the probability of hypothesis H_p against hypothesis H_d given the non-scientific evidence and background information I, this is specified as the prior. The prior is defined as

$$\frac{\mathbb{P}\left(H_p|I\right)}{\mathbb{P}\left(H_d|I\right)}, \tag{3.1}$$

and is also called the prior belief or prior odds. When scientific evidence is present, one is interested which hypothesis is more likely given all evidence, including all scientific evidence e. The value of interest is the ratio

$$\frac{\mathbb{P}(H_p|e,I)}{\mathbb{P}(H_d|e,I)}, \tag{3.2}$$

this is called the posterior odds. It is an update of the prior odds with information provided by the scientific evidence e. It gives legal experts a numerical value to support a decision between the two competing hypotheses. It is not possible to calculate the posterior odds immediately. To calculate the posterior odds, the prior odds have to be updated. First, the theorem of Bayes is introduced [45]. Let A and B be events where $\mathbb{P}(A) > 0$ and $\mathbb{P}(B) > 0$, then the conditional probability of A given B can be expressed through

$$\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(A)}{\mathbb{P}(B)}. \tag{3.3}$$

The posterior odds in Equation (3.2) can be rewritten using the Bayes theorem and the definition of conditional probabilities as

$$\begin{aligned}\frac{\mathbb{P}\left(H_p|e,I\right)}{\mathbb{P}\left(H_d|e,I\right)} &= \frac{\mathbb{P}(e,I|H_p)\mathbb{P}(H_p)}{\mathbb{P}(e,I)}\cdot\frac{\mathbb{P}(e,I)}{\mathbb{P}(e,I|H_d)\mathbb{P}(H_d)}\\ &= \frac{\mathbb{P}(e|H_p,I)}{\mathbb{P}(e|H_d,I)}\cdot\frac{\mathbb{P}(I|H_p)}{\mathbb{P}(I|H_d)}\cdot\frac{\mathbb{P}(H_p)}{\mathbb{P}(H_d)}.\end{aligned} \tag{3.4}$$

Often the non-scientific background information I is omitted, because of readability. Equation (3.4) leads to the standard form of the likelihood ratio framework

$$\underbrace{\frac{\mathbb{P}\left(H_p|e\right)}{\mathbb{P}\left(H_d|e\right)}}_{\text{posterior odds}} = \underbrace{\frac{\mathbb{P}\left(e|H_p\right)}{\mathbb{P}\left(e|H_d\right)}}_{\text{likelihood ratio}} \cdot \underbrace{\frac{\mathbb{P}\left(H_p\right)}{\mathbb{P}\left(H_d\right)}}_{\text{prior odds}} . \tag{3.5}$$

The likelihood ratio is used as the value of evidence. Equation (3.5) is called the odds form of the theorem of Bayes. The prior belief is updated by the likelihood ratio (LR), to find the posterior odds [23]. A $LR > 1$ favours the prosecutor's hypothesis H_p, an $LR < 1$ favours the defence hypothesis H_d. An advantage of the likelihood ratio framework is the division between the task of the legal experts and the forensic experts. The forensic scientist provides an objective likelihood ratio, while the legal expert can determine an appropriate prior belief, before seeing the scientific evidence. The legal expert can afterwards update this belief to the posterior odds with the likelihood ratio provided by the forensic scientist.

To calculate the ratio between the two probabilities of the evidence given the hypotheses, the two likelihood functions for the total evidence e conditional on the hypotheses need to be determined. The likelihood ratio is given by

$$\mathrm{LR}(\boldsymbol{\theta}, e) = \frac{f(e|H_p,\boldsymbol{\theta})}{f(e|H_d,\boldsymbol{\theta})}. \tag{3.6}$$

Here, $\boldsymbol{\theta}$ is the multi-dimensional parameter indexing the likelihood function. The true parameter $\boldsymbol{\theta}$ is unknown and an estimate of the true parameter has to be substituted. To determine an estimate for this parameter, two approaches are possible, the frequentist and Bayesian approach. The parameter can be estimated using the background material, the estimated parameter can be plugged into the equation, this approach is the frequentist approach. The Bayesian approach proposes an appropriate prior for parameter $\boldsymbol{\theta}$, this prior needs to be chosen carefully, as it effects the inference process. In this thesis only the frequentist approach is researched, as this approach is analytical tractable and no priors have to be specified. The likelihood functions are defined per identification of source framework, the common source model and the specific source model. A complete and detailed version of the derivation of the likelihood functions can be found in [34].

For the common source model, the unknown source evidence consists of two speech fragments of unknown speakers, $e = \{e_{u_1}, e_{u_2}\}$ [34], applying this to Equation (3.6) gives

$$\mathrm{LR}(\boldsymbol{\theta}_a, e_{u_1}, e_{u_2}) = \frac{f(e_{u_1}, e_{u_2}|H_p,\boldsymbol{\theta}_a)}{f(e_{u_1}, e_{u_2}|H_d,\boldsymbol{\theta}_a)}. \tag{3.7}$$

Here, $\boldsymbol{\theta}_a$ is the estimated l-dimensional parameter using the background population of alternative sources. The numerator is conditioned on hypothesis H_p. The sampling model specified in Chapter 2 shows that e_{u_1} and e_{u_2} are generated by drawing a random speaker $\mathbf{p}$ from the background population and drawing two samples from this speaker. The two samples are independent conditionally on the drawn speaker. Assume $\boldsymbol{y}_{u_1}$ and $\boldsymbol{y}_{u_2}$ are the resulting feature vectors from e_{u_1} and e_{u_2}, respectively. To draw a random speaker, the between-source and within-source distribution from Equations (2.1) and (2.2) are used. f_a is the probability density function of the within-source distribution and g is the probability density function of the between-source distribution. This results in

$$\begin{aligned} f(e_{u_1}, e_{u2}|H_p,\boldsymbol{\theta}_a) = f_a(\boldsymbol{y}_{u_1}, \boldsymbol{y}_{u_2}|H_p,\boldsymbol{\theta}_a) &= \int f_a\left(\boldsymbol{y}_{u_1}, \boldsymbol{y}_{u_2}|\mathbf{p},\boldsymbol{\theta}_a, H_p\right) g\left(\mathbf{p}|\boldsymbol{\theta}_a\right) d\mathbf{p} \\ &= \int f_a\left(\boldsymbol{y}_{u_1}|\mathbf{p},\boldsymbol{\theta}_a\right) f_a\left(\boldsymbol{y}_{u_2}|\mathbf{p},\boldsymbol{\theta}_a\right) g\left(\mathbf{p}|\boldsymbol{\theta}_a\right) d\mathbf{p}. \end{aligned} \tag{3.8}$$

The denominator is conditioned on hypothesis H_d. The sampling model specified in Chapter 2 shows that the evidence is generated by randomly drawing two speakers $\mathbf{p}_1$ and $\mathbf{p}_2$ from the background population and drawing one sample from each speaker. The evidence is from two different randomly drawn speakers, thus evidence e_{u_1} and e_{u_2} are unconditionally independent. Then the denominator in Equation (3.7) can be written as

$$\begin{aligned} f(e_{u_1}, e_{u_2}|H_d,\boldsymbol{\theta}_a) &= f(e_{u_1}|\boldsymbol{\theta}_a) f(e_{u_2}|\boldsymbol{\theta}_a) = f_a(\boldsymbol{y}_{u_1}|\boldsymbol{\theta}_a) f_a(\boldsymbol{y}_{u_2}|\boldsymbol{\theta}_a) \\ &= \int f_a\left(\boldsymbol{y}_{u_1}|\mathbf{p}_1,\boldsymbol{\theta}_a\right) g\left(\mathbf{p}_1|\boldsymbol{\theta}_a\right) d\mathbf{p}_1 \cdot \int f_a\left(\boldsymbol{y}_{u_2}|\mathbf{p}_2,\boldsymbol{\theta}_a\right) g\left(\mathbf{p}_2|\boldsymbol{\theta}_a\right) d\mathbf{p}_2. \end{aligned} \tag{3.9}$$

For the specific source model, the unknown source evidence consists of one fragment of speech of unknown speaker $e = \{e_u\}$ [34]. The numerator is conditioned on hypothesis H_p, the sampling model specified in Chapter 2 shows that e_u is generated by the specific known speaker. Therefore, no between-source distribution is present here. Let $\boldsymbol{y}_u$ be the resulting feature vector from e_u. The numerator can be rewritten as

$$f(e|H_p,\boldsymbol{\theta}) = f(e_u|H_p,\boldsymbol{\theta}_s) = f_s(\boldsymbol{y}_u|\boldsymbol{\theta}_s). \tag{3.10}$$

Here, $\boldsymbol{\theta}_s$ is the estimated parameter using the specific speaker data set and f_s the probability density function of the specific speaker. The denominator is conditioned on hypothesis H_d. The sampling model from Chapter 2 shows that e_u is generated by drawing a random speaker p from the background population and drawing one sample from this speaker. In the same manner as the common source denominator, this results in

$$\begin{aligned} f(e|H_d,\boldsymbol{\theta}) = f(e_u|H_d,\boldsymbol{\theta}_a) &= f_a(\boldsymbol{y}_u|\boldsymbol{\theta}_a) \\ &= \int f_a\left(\boldsymbol{y}_u|\mathbf{p},\boldsymbol{\theta}_a\right) g\left(\mathbf{p}|\boldsymbol{\theta}_a\right) d\mathbf{p}. \end{aligned} \tag{3.11}$$

Here, $\boldsymbol{\theta}_a$ is again the estimated parameter using the background population of alternative sources.

To calculate the specified likelihood functions, the corresponding parametric distribution and the indexing parameter for f_a and g have to be estimated, this is further outlined in Chapter 6. However, as already mentioned in the introduction, specifying an appropriate parametric distribution and fitting it correctly for a high-dimensional space is challenging. Therefore, in the next section the score-based approach is outlined.

3.2. Score-based approach

In the previous section the value of evidence is expressed as a likelihood ratio. The direct method to calculate a likelihood ratio uses Equation (3.6) directly with the specified likelihood functions. This is called a feature-based approach, it assumes a distribution of the original features. However, to calculate the feature-based likelihood functions, several problems arise. First, due to the high dimensional feature space in text analysis (further outlined in Chapter 5), the number of samples has to be substantially large to fit a parametric distribution accurately. Due to limited available data this is not always feasible. Also, high-dimensional models require a large computational power. Second, by directly calculating the likelihood ratio using the high dimensional feature space, maybe unsupported assumptions are made regarding the underlying process that generated the evidence [19]. No literature is available which proposes distributions for the used features based on words.

To avoid the difficulties arising from a high dimensional feature-based approach, the score-based approach is introduced. Instead of having a multi-dimensional feature vector per sample, a one-dimensional score is calculated between two samples, under the given competing hypotheses [6, 19, 36]. This score is calculated by using a scoring method. As a baseline a distance based method is used, where the score is the distance between two feature vectors. To improve upon this baseline, several machine learning algorithms are applied to calculate a score from the feature vectors. As a third method a feature-based likelihood ratio is calculated under several assumptions and used as a score. The used score methods are further explained in Chapter 6. All scoring methods have the same kind of input and output. Assume two samples with l-dimensional feature vectors x_i and x_j, this is the input of the method. The output is a one-dimensional score labelled as same-source or different-source score. The same-source scores are obtained by comparing same-source pairs of feature vectors and the different-source scores are obtained comparing different-source pairs of feature vectors. The likelihood ratio is calculated dividing the probability density function of the same-source score distribution by the probability density function of the different-source score distribution. The needed data sets for the score-based approach agree with the sampling models specified in Chapter 2. The input is respectively the fragment(s) of speech from an unknown speaker, the background population of alternative speakers data set and in case of specific source modelling, a specific known speaker data set. From the background population (and the specific speaker data set) the same-source and different-source scores are calculated. The probability distribution of both of sets of scores is estimated using a non-parametric kernel density estimate, this is further outlined in Chapter 7. The score likelihood ratio is calculated by the ratio between the two distribution of the scores, given the competing hypotheses [14]. Figure 3.1 shows the schematic figure for obtaining the likelihood ratio with the score-based approach [31].

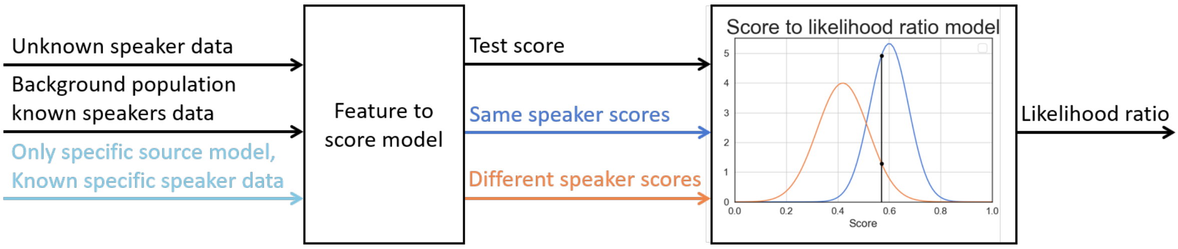

Unknown speaker data
Background population known speakers data
Only specific source model, Known specific speaker data
Feature to score model
Test score
Same speaker scores
Different speaker scores
Score to likelihood ratio model
Score
Likelihood ratio

Figure 3.1: Schematic overview of the score-based method, based on [31] (Fig 1, p. 48)

The common source model uses the background population to calculate the same-source and different-source scores. In case of speaker recognition, the same-source scores are calculated from sample pairs from one speaker and the different-source scores are calculated from sample pairs from two different speakers. The score distributions do not change when using a different disputed sample of speech, since the distributions are only based on the background population. Figure 3.2 shows the schematic overview for obtaining the scores for the common source model.

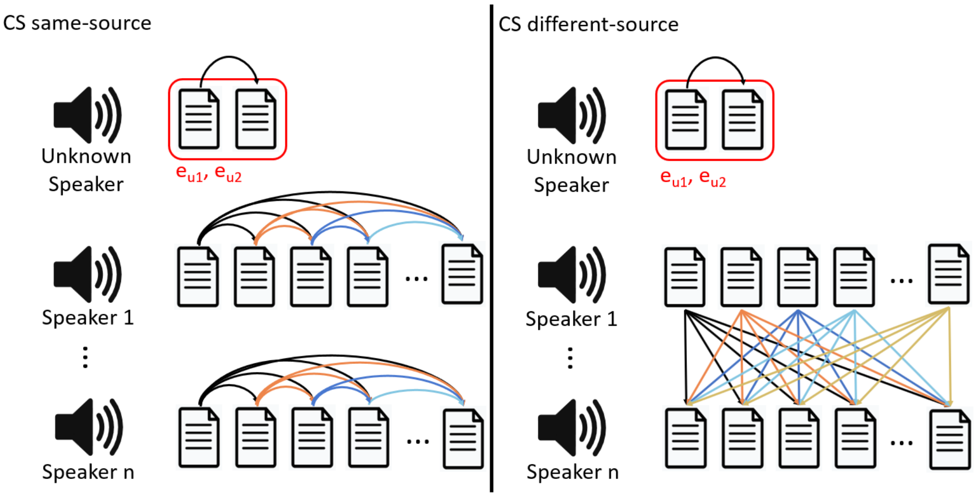


Figure 3.2: Schematic overview of obtaining the same-source and different-source scores for speaker recognition using the common source approach.

The common source score-based likelihood ratio is defined as

$$\mathrm{LR}(e_{u_1}, e_{u_2}, e_a) = \frac{f(s(e_{u_1}, e_{u_2})|H_p, e_a)}{f(s(e_{u_1}, e_{u_2})|H_d, e_a)}, \qquad (3.12)$$

with f the density functions estimated by the kernel density estimate and s the used scoring method.

Important to note is the number of different-speaker pairs and same-speaker pairs, which can be obtained from a given data set. To form the different-speaker pairs, the samples of a speaker can be paired with samples from all other speakers in the set. To form the same-speaker pairs, the samples from a speaker can only be paired with samples obtained from the same speaker. The effect on the number of pairs that can be retrieved is explained with an example. Assume we have a data set which contains $n = 3$ speakers with each $m_i = 5$ samples, where m_i is the number of samples of speaker i. To form same-speaker pairs, within each speaker i, $\sum_{j=1}^{m_i}(m_i - j) = 4+3+2+1 = 10$ combinations are possible. This results in $3 \times 10 = 30$ same-speaker pairs. For the different-speaker pairs, all samples from a speaker can be paired with samples from other speakers. This results per speaker i in $m_i \times \left(\sum_{j\neq i} m_j\right) = 5 \times (5+5) = 50$ different-speaker pairs. Because of the double counting per pair, the total amount is $\frac{3\times 50}{2} = 75$ different-speaker pairs. In a real data set the number of samples obtained per speaker is not always equal, but the effect is the same. The example shows that the number of same-speaker pairs can be a limiting factor to estimate the score distribution under the given hypotheses. The number of pairs obtained to calculate the scores will be outlined in the results chapter.

The specific source model constructs the same-source and different-source scores from two data sets, the background population and the specific source data set. In case of speaker recognition, the same-source scores are constructed by using combinations of samples from the specific known speaker. The different-source scores are calculated with sample pairs from two different speakers in the background population of alternative sources. The same-score distribution is dependent on the specific known speaker specified in the hypothesis. Figure 3.3 illustrates the generation of scores for the specific source model.

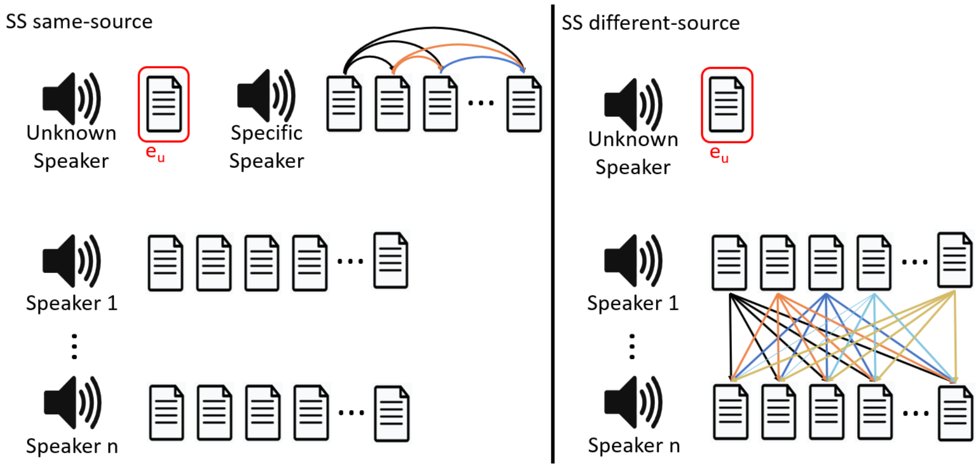


Figure 3.3: Schematic overview of obtaining the same-source and different-source scores for speaker recognition with a specific source model.

The score-based likelihood ratio for the specific source model can be represented as

$$\mathrm{LR}(e_u, e_a, e_s) = \frac{f(s(e_u)|H_p, e_a, e_s)}{f(s(e_u)|H_d, e_a)}, \tag{3.13}$$

with f the density functions estimated by the kernel density estimate and s the used scoring method. This is called an asymmetric score-based approach, since the numerator and denominator are conditioned on different evidence. This method is chosen, as the specific speaker evidence is still very limited in a real forensic case.

An important topic in score-based likelihood ratio systems is taking into account similarity and typicality. If we take two feature vectors, similarity is defined as how similar the two feature vectors are. Typicality is defined as how likely it is to observe the features in the general population. Taking into account only typicality or similarity with the score calculation could lead to other inference results. Because as input only a one-dimensional score is calculated from the multi-dimensional feature vector, a large amount of information is lost. The score should contain the largest possible amount of information about the original feature vectors. The literature shows the results of some similarity-only scores give varying results [31, 52], but also promising results are presented in, for example, fingermarks [26]. The results are dependent on the specific case and the available data. Therefore, in this thesis different scoring methods are explored, further outlined in Chapter 6. For all score-based approaches, an extensive validation framework is important.

3.3. Verbal likelihood ratios

It is important to mention that likelihood ratios are used in a numerical context, but also in an equivalent verbal context. In 2014, the NFI stated a generic framework for likelihood ratios, including a table with the verbal equivalent per range of likelihood ratios, this is shown in Table 3.1.

Table 3.1: Table of the verbal expressions of numerical likelihood ratio values [33].

Values of likelihood ratio	Verbal equivalent
1-2	As probable, no assistance
2-10	Slightly more probable
10-100	More probable
100-10.000	Much more probable
10.000-1.000.000	Far more probable
>1.000.000	Exceedingly more probable

Especially when a likelihood ratio has to be reported in court, it is useful to have a verbal equivalent. The numerical statement of an forensic expert in a speaker recognition case would be:

> *'The information found by comparing the two speech fragments is 160 times more probable if they come from the same speaker than if they come from different speakers.'*

Which can be translated, according to Table 3.1, to the verbal counterpart:

> *'The information found by comparing the two speech fragments is much more probable if they come from the same speaker than if they come from different speakers.'*

3.4. Combination of forensic evidence

Analysing the transcribed speech fragment for speaker recognition is assumed to be independent from current automatic voice recognition. As an additional mean of evidence, text analysis can contribute significantly to the overall evidential information about the speaker. To specify this possibility more precisely, in this chapter the theory and constraints of combining evidence are further outlined.

First, the concept 'Hierarchy of propositions' is introduced. The concept is introduced by [9] and is divided in three levels, 1. Source, 2. Activity and 3. Offence. A small example with corresponding hypotheses for a DNA research is given in Table 3.2.

Table 3.2: Example of the hierarchy of propositions in a DNA research context [63].

Level	Hypotheses
3. Offence	H_p: Suspect A assaulted person B H_d: An unknown person assaulted person B
2. Activity	H_p: Suspect A was present during the assault H_d: Suspect A was not present during the assault
1. Source	H_p: The hair from suspect A was found at the crime scene H_d: The hair from an unknown person was found at the crime scene

Level 1 contains the least complex propositions, the identification of source propositions. The analysis of the first level is based on empirical data. Level 2 consists of propositions about activities. Not only data is needed to evaluate these propositions, but also the probability of transferring the evidence and persistence. Level 3 is a proposition about an actual offence, this is up to legal experts to decide [63]. A legal expert is most interested in answering the higher level questions. Forensic scientist are generally working on the activity or source level. For some types of evidence, as is the case with speaker recognition, the source and activity level are overlapping. For speaker recognition, the evidence is the speech fragment, which includes the source and the activity of speaking. For example, a fragment of speech where a person talks about an illegal activity is the disputed fragment. The source and activity level are overlapping, as the source of fragments speaks about the activity.

Different pieces of evidence can contribute to the total strength of the evidence. The different likelihood ratio values can be stated separately, but to guarantee the likelihood ratio values are combined in the most optimal way, the evidence can also be combined before used at court. To combine forensic evidence, one needs to understand the probabilistic dependency structure of the problem. Bayesian networks can be used to represent the dependency structure [63]. A Bayesian network is a directed graph that illustrates the dependencies between random variables. The variables are the nodes, the dependencies are illustrated by directed edges.

For speaker recognition three parts of evidence can be combined. First, the information obtained from the automatic speaker recognition e_{asr}, based on phonetics and speech signal processing. Second, the information from the text analysis e_{ta}, only based on the transcription of the speech fragment. Third, the evidence resulting from the judgement of an expert e_{je}. For the first two pieces of evidence e_{asr} and e_{ta}, a numerical likelihood ratio can be computed. The two pieces of evidence are based on different parts of the speech fragment, transcription and phonetics, and thus can be assumed conditionally independent. This assumption must be taken cautiously and has to be validated precisely in further research. This means the prior belief regarding the hypotheses can be updated in the same way as introduced in Section 3.1. An addition to the likelihood ratio defined in Equation (3.6) results in

$$\mathrm{LR}(e_{asr}, e_{ta}) = \frac{f(e_{asr}|H_p)}{f(e_{asr}|H_d)} \cdot \frac{f(e_{ta}|H_p)}{f(e_{ta}|H_d)}. \tag{3.14}$$

The dependencies are shown in a Bayesian network on the left side of Figure 3.4. The hypotheses node, with no incoming edges, is called the parent node. The evidence nodes, with incoming edges, are called child nodes. The two edges pointing to the evidence show the conditional probability dependency.

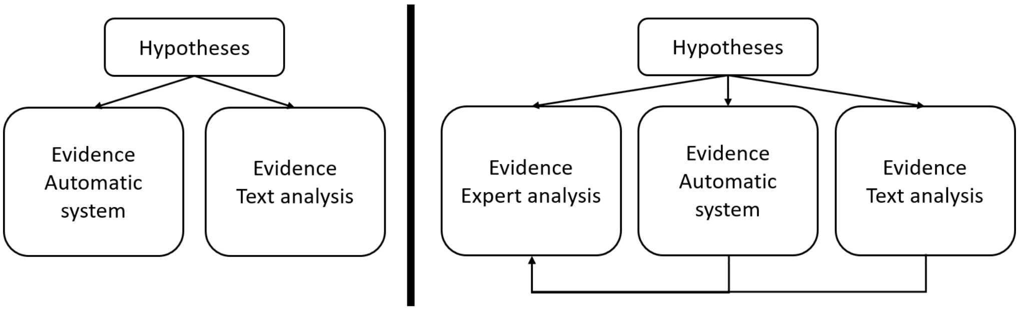


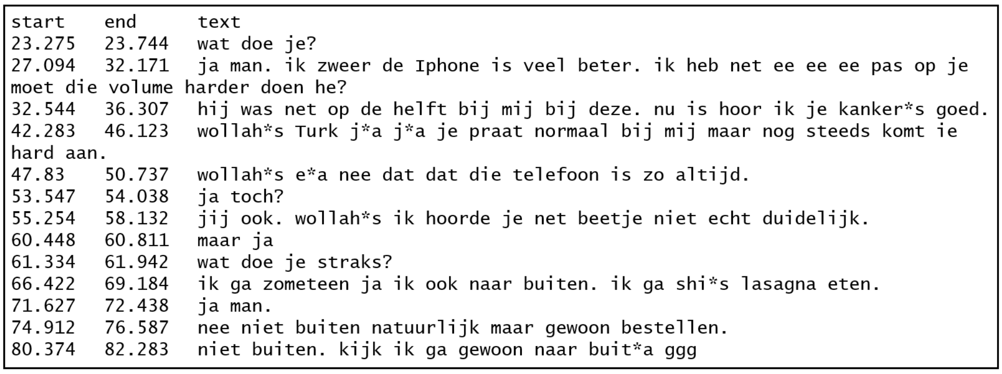
Figure 3.4: Two Bayesian networks for combining evidence in a speaker recognition case. The left side shows the simple Bayesian network for the evidence from the automatic system and from the text analysis, which are assumed to be independent. The right side shows the more complex Bayesian network including the judgement of a speech expert, which cannot be assumed independent from the other two pieces of evidence.

The evidence resulting from the judgement of a legal expert e_{je} is harder to combine with the other two pieces of evidence e_{asr} and e_{ta}. The analysis of the speech expert is based both on phonetics and the transcription, thus the two pieces of evidence e_{asr}, e_{ta} and the judgement of expert e_{je} are conditionally dependent. Furthermore, the judgement is often stated in a verbal context as outlined in Table 3.1 and not in a numerical value. The right side of Figure 3.4 shows the the dependency structure in a Bayesian network. Probably a more complete evidential value can be determined, but only if the dependency between is the pieces of evidences is clear and defined. This is outside the scope of this thesis and will not be discussed further.

4

Data set NFI

The speech department at the NFI recently acquired a data set specifically designed for research in forensic science, the Forensically Realistic Inter-Device Audio Database (FRIDA). This data set is constructed of conversations of 250 different people, each having 16 conversations of approximately 5 minutes. In total this resulted in 333 hours of speech [56]. The speakers are all from a similar target group; male, dutch speaking (50% native, 50% immigrant background), of lower education level and 80% younger than 35 years of age [54]. The data set is designed for automatic speaker comparison (based on phonetics) research and to serve as a background population in forensic case studies. Half of the conversations of all speakers have been transcribed and thus this data set is also very suitable for the research in this thesis. In this research only the transcriptions of the data set are used, not the audio files or further information about recording devices. Each person is recorded during conversations with another person from the group, with 4 recording sessions per day. Half of the conversations per person were held inside and half outside, divided in noisy areas and quiet areas. All transcripts contain the timestamp followed by the spoken text. The conversations are orthographic transcribed by three native Dutch speakers. Orthographic means according to the standard spelling of a language, in this case Dutch. Per conversation, the fragment is transcribed by one and checked by the other two, to avoid personal influences in transcribing. A small example of the transcribed data can be found in Figure 4.1.



```
start    end      text
23.275   23.744   wat doe je?
27.094   32.171   ja man. ik zweer de Iphone is veel beter. ik heb net ee ee ee pas op je
moet die volume harder doen he?
32.544   36.307   hij was net op de helft bij mij bij deze. nu is hoor ik je kanker*s goed.
42.283   46.123   wollah*s Turk j*a j*a je praat normaal bij mij maar nog steeds komt ie
hard aan.
47.83    50.737   wollah*s e*a nee dat dat die telefoon is zo altijd.
53.547   54.038   ja toch?
55.254   58.132   jij ook. wollah*s ik hoorde je net beetje niet echt duidelijk.
60.448   60.811   maar ja
61.334   61.942   wat doe je straks?
66.422   69.184   ik ga zometeen ja ik ook naar buiten. ik ga shi*s lasagna eten.
71.627   72.438   ja man.
74.912   76.587   nee niet buiten natuurlijk maar gewoon bestellen.
80.374   82.283   niet buiten. kijk ik ga gewoon naar buit*a ggg
```

Figure 4.1: An example of a part of a transcript.

An interesting feature for this research is the notation of words in informal language. The transcription makes special notations for words from a different language, new non-existing words, street words, not completed words and distortion of words, with *v, *n, *s, *a and *u respectively. An example, also included in Figure 4.1, is the word ‘wollah’ which is transcribed as ‘wollah*s’. This ensures the data set is readable and resistant against different spellings, this is important for text analysis.

The strength of this data set is the size, with 250 different speakers and 8 transcribed fragments per speaker, the homogeneous target group and the available transcription. However, some remarks are important to note, to keep in mind in the remainder of this thesis. For speaker verification, it is important to have a data set that contains conversations about different topics. The goal is to classify a transcription by the speaker, not by topic. The speakers are asked to talk normal and not focus on specific topics. However, some speakers talk about the same topic in all eight conversations. If this is the case, it is difficult to determine which features are effective non-topic instead of topic based. In a real forensic case, topics could completely differ between the two fragments of speech from unknown speakers, thus it is necessary to use non-topic features. It is noted that in some conversations the person did not know what to talk about, but had to finish up the required 5 minutes. This results in sentences which would probably not have been used in a spontaneous conversation. Consequently this could lead to less informative samples for a speaker. Furthermore, because the data set is used as training and test set to validate the method, it can be assumed that the background population is certainly relevant. For a real forensic case, this background population has to be specified according to the knowledge the forensic scientist has about the speaker from the forensic case. Thus the agreement of the background population with the disputed data in an actual forensic case could be different, which could give other results.

The FRIDA data set showed an inconsistent number of transcribed recordings per speaker. For some speakers, a smaller number than 8 transcriptions of recording sessions were available. To obtain a stable amount of data per speaker, the speakers with less than 8 transcribed speech fragments are not used in the test case. After the data selection, the data set contains 217 speakers.

Besides the FRIDA data set, another data set containing transcribed Dutch speech is the ‘Corpus Gesproken Nederlands’ (CGN) data set. The CGN data set is acquired from 1998 to 2004 with the goal to have a database of contemporary Dutch as spoken by adult speakers in the Netherlands and Flanders [1]. The set contains transcribed fragments from various speech fragments, for example telephone conversations, interviews, speeches etc. The speech fragments are transcribed according to the same guidelines as the FRIDA data set. A subset of the CGN data set, containing spontaneous telephone conversations, is used to determine a set of frequent words independent from the FRIDA data set.

5 Text analysis

The first step in the process of analysing the transcriptions, is the transformation from raw transcriptions to features vectors. A feature is a measurable characteristic from a transcription. A feature should help to distinguish between different speakers of speech fragments. All different features from a speech fragment combined form a feature vector. The numerical feature vector is used to train a model or measure similarity between different transcriptions.

Author analysis, based on written text, has already been applied to different data sets, for example email, literary books and presidential written speeches. From the authorship analysis research, the promising, often used effective features are selected. Besides using features from authorship analysis research, it is also interesting to explore if transcribed spoken text has data specific features. An example of a data specific feature is a title or greeting for email data. A feature set could work well for classifying literary books, but this set could fail for online messages. In the case of transcribed speech, it is interesting to look at features which are normally not used in written text. In the upcoming part of this section, different types of features are discussed, including their applicability in the speaker recognition case. At the end, the final selected features are presented.

While establishing the selected features, it is important to keep the goal of the research in mind. For speaker recognition, the features have to be discriminating for different speakers, not for different topics in a conversation. For example, a speaker will speak about completely different topics during a police conversation, than during a phone conversation with a friend. The ideal features do not discriminate for topics or content, but only on variables which are unconsciously used in speech. The features discriminate between style and not topic.

Lexical features

Lexical features are based on tokens. A text is a sequence of tokens divided by spaces, thus words, numbers and special symbols are all separate tokens. In the analysed transcription from a speech fragment no punctuation is present, every word or number can be seen as a token. For authorship analysis research, several studies use features based on frequencies of separate tokens or token n-grams. A token n-gram is a combination of n different tokens in a sequence. An example is given in Figure 5.1.

```
Hij heeft dat geld

1-grams: [hij], [heeft], [dat], [geld]
2-grams: [hij heeft], [heeft dat], [dat geld]
3-grams: [hij heeft dat], [heeft dat geld]
```

Figure 5.1: Example of dividing a sentence in token n-grams, for $n \in \{1,2,3\}$.

In most authorship analysis research, only the frequencies of the $F_{\#}$ most frequent words (token 1-grams) in a relevant background population data set are used [5, 48, 62]. These words do not contain information about the topic, and are also called function words. Function words are words that primarily have a grammatical function within a sentence and no substantive meaning. Examples of Dutch function words are ‘want’, ‘naar’ and ‘voor’, translated in English to ‘as’, ‘to’ and ‘for’. It is interesting to note that for topic-based text classifying, the most frequent words are often not used, as they do not contribute to topic classifying. The parameter $F_{\#}$, the number of frequent words used as features, varies a lot in different studies [50].

Character features

Character features are based on characters, every separate letter or punctuation is a character. A feature can be one character or a sequence of characters. The most useful character-based features in literature are letter frequency, the total number of characters per sentence and character n-grams [4]. A character n-gram is a combination of different letters, spaces and punctuation marks in a sequence. Figure 5.2 shows an example of token n-grams.

```
Hij heeft dat geld

1-grams: [h], [i], [j], [h], [e], [e], …
2-grams: [hi], [ij], [j_], [_h], [he], …
3-grams: [Hij], [ij_], [j_h], [_he], [hee], …
```

Figure 5.2: Example of dividing a sentence in character n-grams, for $n \in \{1,2,3\}$.

The parameter n determines the size of the combination. A large n could lead to using thematic information, because complete sets of words will be included. Small n will lead to more stylistic factors. As for the token n-grams, only the k most frequent ones should be taken into account, to guarantee only non-substantive information. A disadvantage of character n-grams is the large amount of features which are created, which results in large dimensional feature vectors. This requires more computational power for calculations.

Syntactic features

Syntactic features are based on syntactic elements, these are language rules. Examples are function words usage, punctuation usage and part of speech (POS) tagging. POS tagging means annotating the grammar within a sentence. It is a promising feature, but also noise and errors could arise because the POS-tagging is done by an natural language processing (NLP) algorithm which is not flawless. These algorithms are designed for a computer to ultimately analyse and understand human language. Function words are already introduced in the lexical features. Instead of using the most frequent words, also a standard set of function words based on grammar rules can be used.

Semantic features

Semantic features show the basic substantive meaning of a lexical element in the text. For example, the word ‘man’ applies to the semantic features ‘human +’ and ‘male +’. The word ‘woman’ applies to the semantic features ‘human +’ and ‘male –’. The ‘+’ indicates the the presence of property and the ‘–’ the absence of the property. Different words can be in the same semantic family. Determining the semantic features heavily relies on complex NLP algorithms. These algorithms are prone to errors and noise. Including semantic features could lead to more noise instead of more information [50].

Structural features

Structural features are based on how text is structured, for example headings, title pages and greetings. They are not based on specific content. The structure of a letter or report could be discriminating for an author. For speaker recognition based on transcriptions, structural features are not useful as in spoken text no structural forms are present.

Idiosyncratic features

Idiosyncratic features are based on the use of non-language elements, for example cursing, non-existing words and spelling mistakes. For transcribed spontaneous speech, these features could be interesting. Different non-language elements during conversations could be discriminating for different speakers.

Speech specific

As stated above, it is interesting to look at speech specific features which are only present in the transcription of spontaneous speech, for example cursing, saying ‘-uh’ less or more often and street language. In this thesis the specific moments of using street language, pronunciation variants and non-Dutch words are precisely transcribed. The speech-specific features are easy to implement and explore as frequencies of speech parts.

5.1. Features FRIDA data set

Summarising the features section, the used features in this thesis are presented. We have chosen to only use frequencies of the $F_{\#}$ most frequent words and the speech specific features. The $F_{\#}$ most frequent words show promising results in authorship analysis research and often have a low number of features in comparison with character n-grams. Most other, more complex, features only contribute to the already discriminating behaviour of the frequent words [15]. Table 5.1 shows the overview with the types of features. In this research the objective is showing effectiveness for spoken text. Some features might be less informative than others. Different algorithms will be used with different combinations of feature sets, to find a suitable feature set with parameters. A general remark is that all highlighted features are frequency features. The feature vector is a simple count of how often a specific feature is present in the data. From this text analysis an $F_{\#}$-dimensional feature vector per text subset is created. All feature vectors combined are the input of the methods in the next chapter.

Table 5.1: Table of the possible features for author analysis on spoken text.

Feature	Type
$F_{\#}$ most frequent words with $F_{\#} \in (50, \ldots, 600)$	Lexical
Counts of using -uh	Speech specific
Counts of street language words	Speech specific
Counts of pronunciation variants	Speech specific
Counts of non-existing words	Speech specific
Counts of not completed words	Speech specific
Counts of non-Dutch words	Speech specific

For the data used in this thesis, the $F_{\#}$ most frequent words are used as features together with the special tokens from the speech transcriptions. The enumeration below shows the first 25 most frequent words from the FRIDA data set. First the word is stated, then the absolute occurrence in the whole set and last the percentage this word occurs of all words in the transcriptions. The total number of words and special speech tokens in the transcription is 961698.

1. ja: 49919x - 5.19%
2. ik: 32697x - 3.40%
3. uh: 32273x - 3.36%
4. je: 29893x - 3.11%
5. *a: 26474x - 2.75%
6. is: 17026x - 1.77%
7. dat: 16568x - 1.72%
8. maar: 14816x - 1.54%
9. die: 14059x - 1.46%
10. en: 14047x - 1.46%
11. niet: 13142x - 1.37%
12. *s: 10986x - 1.14%
13. dan: 10873x - 1.13%
14. een: 10849x - 1.13%
15. wel: 9905x - 1.03%
16. man: 9842x - 1.02%
17. nee: 9432x - 0.98%
18. het: 9398x - 0.98%
19. ook: 8553x - 0.89%
20. gewoon: 8487x - 0.88%
21. de: 8399x - 0.87%
22. wat: 7862x - 0.82%
23. *v: 7739x - 0.80%
24. t: 7554x - 0.79%
25. weet: 7470x - 0.78%

The complete list with the first 200 most frequent words in the FRIDA data set can be found in Appendix A.1. An additional list with 200 most frequent words from the CGN data set can be found in Appendix A.2. What is interesting to see is that the speech token for not completed words *a is very high on place 5. The use of street words and words in another language, transcribed with *s and *v, is also relatively high on places 12 and 23, respectively. The number of frequent words $F_{\#}$ is an important parameter to test during the performance studies later in this thesis. It is interesting to investigate what the optimal number of frequent words $F_{\#}$ is.

To start the process of obtaining feature vectors from the raw data, first the number of frequent words $F_{\#}$ and sample length N is chosen. Obtaining the most frequent words from the complete data set leads to a set of words similar to the one introduced above. First, all data is concatenated per speaker. The available data per speaker is divided in samples of sample length N, so every sample consists of N tokens of the original data. After dividing the data in samples, per sample all words are deleted, except for the high frequent words. An example with $F_{\#} = 25$, as showed above, is shown in Figure 5.3.

```
Ja man hij heeft dat geld thuis,
ik weet het anders niet
Frequent words:
[ja] [man] [dat] [ik] [weet] [het] [niet]
```

Figure 5.3: Example of removing all non-frequent words, with $F_{\#} = 25$.

Per sample of N tokens, a feature vector based on the frequencies of the frequent words in the sample is obtained. The feature vectors are used as input for the scoring methods in Chapter 6. Algorithm 1 shows the overall method of obtaining features from the raw data set.

Algorithm 1 Feature extracting method

```
Specify parameters: $N, F_{\#}$
Read data set, tokenise all words
Count occurrence of all words, select the $F_{\#}$ most frequent words as features
Concatenate all words per speaker
Divide, per speaker, the words in samples of size $N$
Remove in all samples all words except the $F_{\#}$ most frequent words
```

5.2. Frameworks in text analysis

In almost all literature about authorship analysis, two main choices between different methods to start the text analysis are made. The choices are between intrinsic and extrinsic methods and the profile-based and the instance-based methods. These methods are introduced below.

Intrinsic methods, also called one-class problems, are author analysis problems that literally only use one class. A class is defined as a specific speaker, the method tries to decide if the text originates from this speaker or not. In a forensic context with competing hypotheses, this is not suitable or desirable, as it is necessary to evaluate the evidence given the competing hypotheses. Extrinsic methods, also called two-class problems, use two classes. A disputed text belongs to one class or the other class. To model the classes, sufficient data from both classes has to be present, this corresponds to the forensic context.

Profile-based methods start with combining all available text from an author [27]. From all text available from the author, a profile is made based on selected features. For all possible authors in the data set, an author model is determined. These models are compared with the model from the unknown text sample [41]. For the forensic context, where the text per author and thus the author profile can be limited, this is not applicable. Instance-based methods start also with combining the available text samples. Instead of creating one author profile, the pile is randomly divided in multiple smaller samples. The instances are used to develop a model to classify new authors. The method is based on the different samples, i.e. different instances, instead of one author model. This method works well if a sufficient amount and length of text of the author is available [41]. For speaker recognition, this approach is useful because all smaller texts can be combined, from which afterwards samples from a specific length can be formed. The instance-based approach is used in this research.

6 Features to scores

The first step of analysing the transcription was obtaining feature vectors from the raw data. The second step is obtaining scores given hypotheses H_d and H_p, as described in Section 3.2. The feature vectors specified in Chapter 5 are the input for the scoring algorithms, a set of $F_{\#}$-dimensional feature vectors corresponding to frequencies of $F_{\#}$ features. The scoring method is an important part of the likelihood ratio method, as this determines the discriminating power between the two hypotheses. Three different scoring methods with varying characteristics are proposed. This is the main distinction in the likelihood ratio approaches investigated in this research. First, the distance method is introduced as a baseline. The transformation from feature vectors to scores is based on a distance measure between two feature vectors. This is the least complex method. Secondly, to improve upon this baseline, several supervised machine learning algorithms are presented. A supervised learning model is trained using a set of labelled feature vectors as examples. The model maps the input feature vectors to an output score. The model calculates a probability score corresponding to the feature vector. As third method a score likelihood ratio is presented. This is a naive feature-based likelihood ratio calculated under several assumptions based on the feature vector. Gaussian distributions are assumed for the within-speaker and between-speaker distributions in the data. The resulting calculated likelihood ratios are used as scores instead of as direct likelihood ratios. The distance method and the machine learning method are only applied to the common source model, because of the insufficient amount of data to model the specific source model. The likelihood ratio score method is implemented for the specific source and the common source model.

6.1. Distance method

The distance method is based on calculating the similarity between two feature vectors by calculating a distance between two feature vectors. The distance between two feature vectors is defined by a distance metric or a statistic. The method is only common source modelled, which implies that the scores calculated are between a pair of feature vectors originating from the same speaker and pairs of feature vectors from two different speakers, all from the background population. Figure 6.1 shows the schematic steps from feature vectors to scores for the distance method.

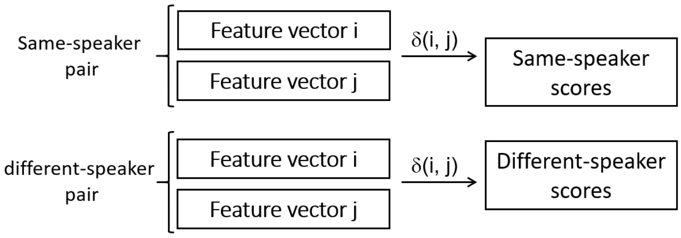


Figure 6.1: Schematic overview of the distance method from input feature vectors to output scores.

The distance method is based on (dis)similarity only, typicality is not used as the score is only a comparison between two features vectors. This means an important discriminating factor is lost. The similarity from a high dimensional space is compressed in a one-dimensional score, which also implies loss of information. The distance method is a simple way to obtain scores, instead of having a more ‘black box’ approach with machine learning. The distance method can be used as a baseline, more complex techniques are expected to have a better performance.

The most important step is choosing an effective distance measure δ. This measure should contain as much information as possible about the original feature vectors. The method is named distance measure as it does not need to possess the mathematical metric properties. A distance metric is a function which obeys the identity, symmetry and triangle inequality axioms. Although the first axioms are desirable, the third one is not needed. Some statistics, also called similarity measures, do not obey these axioms, but show promising behaviour for defining a similarity measure between two feature vectors. Both similarity measures and distance metrics are explored for the distance method. In the literature, many different distance measures have been proposed and tested for authorship analysis. Examples of used distance measures for author analysis on written text are the Manhattan distance by [15] and the Kullback—Leibler divergence by [48]. In the literature about different identification of source questions (drugs, shoe soles, etc.), the distance method is also applied with different distance metrics, for example the Kullback–Leibler divergence for fingermarks [19]. Below, the best performing distances are presented. For every measure outlined below, $\mathbf{x}$ and $\mathbf{y}$ are the feature vectors where we want to determine the distance between and x_i and y_i a single feature i from these feature vectors, with $i \in (1,\dots,l)$. Below several often used metrics are introduced, the performance of all the distance measures is assessed in the complete likelihood ratio model.

An often used metric for text analysis is the cosine similarity, defined as

$$\delta_{cos}(\mathbf{x},\mathbf{y}) = \frac{\mathbf{x}\cdot\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|} = \frac{\sum_{i=1}^{l} x_i y_i}{\sqrt{\sum_{i=1}^{l} x_i^2}\sqrt{\sum_{i=1}^{l} y_i^2}}. \tag{6.1}$$

The cosine similarity is used distance measure for the authorship detection research in [21].

The Minkowski distance is a generalised distance metric. The Minkowski distance of order p is defined as

$$\delta_{min}(\mathbf{x},\mathbf{y}) = \left(\sum_{i=1}^{l} |x_i - y_i|^p\right)^{1/p}, \tag{6.2}$$

with integer $p \geq 1$. Three well-known distance metrics are the Manhattan, Euclidean and Chebyshev distance, for $p = 1$, $p = 2$ and $p = 3$, respectively. The Manhattan distance is used in the author verification research in [15], for written text in different languages and genres. The Manhattan distance is also used within other applications for score-based likelihood ratios, for example comparing the source of drug tablets [6].

The Bray–Curtis distance is often used in biology and defined as

$$\delta_{bc}(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{l} |x_i - y_i|}{\sum_{i=1}^{l} |x_i + y_i|}. \tag{6.3}$$

The metric is commonly used in ecology, quantifying the similarity between two areas based on counts of species on the two sites [46]. Instead of counting species per site, in text analysis, frequent words per sample are counted.

The Jensen–Shannon distance is based on the Kullback–Leibler divergence and is defined as

$$\delta_{SJ}(\mathbf{x},\mathbf{y}) = \sqrt{\frac{D(\mathbf{x}\|\mathbf{m}) + D(\mathbf{y}\|\mathbf{m})}{2}}, \tag{6.4}$$

with D the Kullback–Leibler divergence and $\mathbf{m}$ the element-wise mean of the two feature vectors $\mathbf{x}$ and $\mathbf{y}$. The Kullback–Leibler divergence is defined as

$$D(\mathbf{x}\|\mathbf{y}) = \sum_{i=1}^{l}\left(\begin{cases} x_i \log(x_i/y_i) & x_i > 0, y_i > 0 \\ 0 & x_i = 0, y_i \geq 0 \\ \infty & \text{otherwise} \end{cases}\right). \tag{6.5}$$

The Jensen–Shannon distance is the square root from the Jensen–Shannon divergence. This metric is described by [12], together with the proof of the metric properties. The advantage of the Jensen–Shannon distance over the Kullback–Leibler divergence is that it is a metric and thus symmetric. The calculation of scores between feature vectors should be symmetric, as the feature vectors are exchangeable.

6.2. Machine learning score algorithms

To improve upon the baseline of the proposed distance method, three machine learning algorithms are outlined. All three algorithms introduced are supervised learning methods, this type of model is trained using a set of labelled feature vectors as examples. Labelled means the feature vectors are tagged with a label, indicating to which class a feature corresponds. For common source speaker recognition, we try to distinguish between two classes, feature vector pairs originating from the same speaker and feature vector pairs originating from two different speakers. These classes correspond to the hypotheses stated for the common source model. Through training, the model learns how the pairs of feature vectors are optimally divided per class. After training the model, the algorithm tries to classify new pairs in the correct class. To train a model accurately, a sufficient amount of training data is needed. We are interested in the probability that a pair corresponds to a certain class, in addition to the classification decision. The algorithms assign a probability score to a pair of feature vectors, which indicates the probability that the pair is in a class. The scores obtained under the given hypotheses are used to calculate the likelihood ratio.

An important topic in using machine learning for forensic purposes is the explainability of the method. A forensic scientist needs to be able to explain how the algorithm contributed to the value of evidence. Following the approach presented by [39], who researched explainability of artificial intelligence for biometrics, four principles are essential for an explainable algorithm. The four principles are explanation, interpretability, explanation accuracy and knowledge limits. This leads to a system which supports the decision with accompanying evidence, which is understandable for the user of the algorithm. The evidence is acceptable for the user and the system only presents decisions for the conditions it was designed for. For the three algorithms introduced below, the general concept is explained including the (dis)advantages presented in the literature. All supervised machine learning algorithms for the common source approach have the same schematic process showed in Figure 6.2, starting with the feature vectors introduced in Chapter 5.

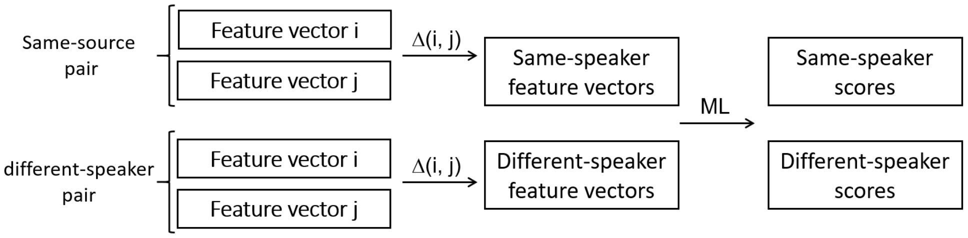


Figure 6.2: Schematic overview of a machine learning method from input feature vectors to output scores.

An important step in the process showed in Figure 6.2, is obtaining a single feature vector for the same-speaker and different-speaker pairs. As a machine learning algorithm uses feature vectors with labels as input, a pair of feature vectors has to be merged to one feature vector with one label. Two approaches can be used. The first one is concatenating the two feature vectors and obtaining a new feature vector with twice the number of features. The second approach is using a distance measure element-wise, to obtain a feature vector of differences with the same number of features. Following the approach in [38], in this thesis we have chosen to merge the two feature vectors to one feature vector by using an element-wise measure $\boldsymbol{\Delta}(\mathbf{x},\mathbf{y})$, to obtain a new feature vector from vectors $\mathbf{x}$ and $\mathbf{y}$. An advantage is that the dimensions of the feature vectors stay equal, instead of being doubled. If the vectors would be concatenated, the machine learning algorithm has to be very powerful to match the different coupled features. The distance measure is chosen, based on the best performing measure from the distance method. The distance measure is adapted, to result in an l-dimensional feature vector, instead of a one-dimensional score. For the Jensen—Shannon distance from Equation (6.4), this results in

$$\boldsymbol{\Delta}_{\boldsymbol{SJ}}(\mathbf{x},\mathbf{y}) = \sqrt{\frac{\boldsymbol{D}_{\boldsymbol{\Delta}}(\mathbf{x}\|\mathbf{m}) + \boldsymbol{D}_{\boldsymbol{\Delta}}(\mathbf{y}\|\mathbf{m})}{2}}, \tag{6.6}$$

with D_{Δ} the Kullback–Leibler divergence element-wise and $\mathbf{m}$ the element-wise mean of the two feature vectors $\mathbf{x}$ and $\mathbf{y}$. The element-wise Kullback–Leibler divergence is given by

$$\boldsymbol{D}_{\boldsymbol{\Delta}}(\mathbf{x}\|\mathbf{y}) = \left[\begin{cases} x_i \log(x_i/y_i) & x_i > 0, y_i > 0 \\ 0 & x_i = 0, y_i \geq 0 \\ \infty & \text{otherwise} \end{cases}\right]. \tag{6.7}$$

The obtained single feature vectors for each same-speaker or different-speaker pair are the input for the machine learning algorithms.

Additional characteristics of machine learning algorithms to take into account include:

- Within machine learning an algorithm can be used for multi-class or binary problems. This means the algorithm has to learn to classify between multiple classes or between two classes. In this research, we are interested in the distinction between two classes, corresponding to the prosecutor's hypothesis H_p and the defence hypothesis H_d. Thus the described algorithms are always used for a binary problem.

- An important property of machine learning algorithms is the bias-variance trade-off. Bias is defined as wrongly making simplifying assumptions in the learning algorithm. High bias in an algorithm can cause missing the important relations and not being sufficiently flexible to model the problem, also known as underfitting. The variance is defined as the sensitivity to small fluctuations in the input. A model with high variance models the noise in the problem, instead of the important relations, also called overfitting [18].

- A critical note is the sensitivity of machine learning algorithms to class imbalance. Class imbalance means that the distribution of classes in the training set is not equally divided. As explained in Section 3.2, the number of same-speaker pairs and different-speaker pairs can vary largely. The user of the algorithm has to be aware when this is the case for a certain training set. Training the algorithm on data with a large class imbalance can lead to poor predictions. This can be regulated using different parameters for the machine learning algorithms and balancing the data set.

Naive Bayes

Naive Bayes is a machine learning algorithm, included in the probabilistic classifier family. It is used in several authorship analysis studies with promising results [4, 20]. The output of the algorithm is a probability of the feature vector being in a class, denoted as c.

Assume input features $\mathbf{x}_i$ with corresponding class labels c_i. The possible classes are the same-speaker and different-speaker class, c_{ss} and c_{ds}, respectively. For a feature vector $\mathbf{x}_i = (f_1, \ldots, f_l)$, the probability of class c given the feature vector $\mathbb{P}(c|\mathbf{x}_i)$ is calculated. This probability can be calculated using the Bayes formula (defined in Equation (3.3)), written as

$$\mathbb{P}(c|\mathbf{x}_i) = \frac{\mathbb{P}(\mathbf{x}_i|c) \cdot \mathbb{P}(c)}{\mathbb{P}(\mathbf{x}_i)}. \tag{6.8}$$

To calculate this equation explicitly, the 'naive' part of the classifier is used, assuming all features in $\mathbf{x_i} = (f_1, \ldots, f_l)$ are independent and uniformly distributed, thus $\mathbb{P}(\mathbf{x_i}|c) = \prod_{i=1}^{l} \mathbb{P}(f_i|c)$. In many realistic cases, this assumption is too strict, but naive Bayes still shows promising results. Noting that the denominator is a constant value, Equation (6.8) can be written as

$$\mathbb{P}(c|\mathbf{x}_i) \propto \mathbb{P}(c) \cdot \prod_{i=1}^{l} \mathbb{P}(f_i|c). \tag{6.9}$$

The labelled training data is used to estimate the probabilities $\mathbb{P}(c)$ and $\mathbb{P}(f_i|c)$. $\mathbb{P}(c)$ is the prior probability, the relative frequency of class c in the training set. The probability $\mathbb{P}(f_i|c)$ is calculated by assuming a parametric distribution. The main distinction between different naive Bayes algorithms is the assumption used to calculate this probability. An often used naive Bayes algorithm for text classification is multinomial naive Bayes [61]. $\mathbb{P}(f_i|c)$ is calculated using a multinomial distribution, the probability that the feature f_i appears in class c is calculated as

$$\mathbb{P}(f_i|c) = \frac{N_{f_i} + \alpha}{N_f + \alpha l}, \tag{6.10}$$

where N_{f_i} is the count of feature f_i in a sample of class c and N_f the count of all features in class c. The smoothing parameter α is added to avoid dividing by zero.

The largest disadvantage of the naive Bayes method is the assumption of independent and uniform distributed features. This assumption is too strict for the features used in this research. An advantage is the simplicity, the algorithm is clearly explainable and has fast computation times. For all calculations the multinomial naive Bayes implementation in the *Scikit-learn* Python library is used. The default and custom settings are outlined in Appendix B.2.

Support vector machines

Support vector machines (SVM) are a powerful classification and regression machine learning algorithm and have been used in various authorship analysis studies [5, 21, 51]. The basic idea behind SVM is determining an optimal hyperplane that divides the data points in classes. For a binary classification problem, the data points on one side of the hyperplane are classified in one class and the data points on the other side of the hyperplane in the other class. First the basic method is outlined for a linearly separable data set, after that the method is extended to usage for non-linearly separable data set.

The separating hyperplane in l-dimensional space is defined as

$$\mathbf{w}^T \mathbf{x}_h + b = 0, \tag{6.11}$$

where $\mathbf{x}_h$ is the set of points that defines the hyperplane, $\mathbf{w}$ the weights and b the bias. Assume a linearly separable training set of l-dimensional feature vectors $\mathbf{x}_i$ and class labels $c \in \{0,1\}$. Since it is assumed that the set is linearly separable, for all feature vectors it holds that

$$\begin{array}{ll} \mathbf{w}^T \mathbf{x}_i + b \geq 0 & \text{for } c_i = 1, \\ \mathbf{w}^T \mathbf{x}_i + b < 0 & \text{for } c_i = 0. \end{array} \tag{6.12}$$

The SVM algorithm tries to find the most optimal hyperplane with weights $\mathbf{w}$ and bias b, to distinguish the two classes. The most optimal hyperplane is defined as the hyperplane with the largest margin between the data points and the hyperplane itself. The margin is the separation between the hyperplane and the nearest data point. The nearest data points can be seen as most difficult to classify and are called the support vectors. The problem then reduces to an optimisation problem to determine the weights and bias to maximise the margin, given the constraint in Equation (6.12). Figure 6.3 shows the choice of an optimal margin for a two-dimensional space and a linearly separable data set.

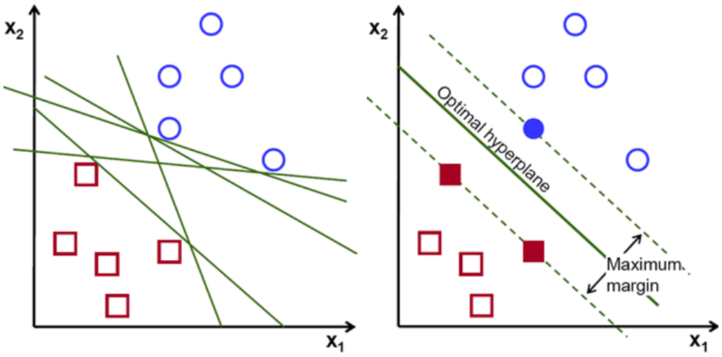


Figure 6.3: The process of determining the optimal hyperplane dividing data points in two classes, for a set linearly dividable data points [22].

The example in Figure 6.3 shows the linear separation for a two-dimensional feature space. For most data one can expect it is not possible to linearly separate the data. To solve the classification problem for a non-linearly separable data set, the data points are mapped to a higher dimensional space, where a hyperplane can be chosen that separates the data. A difficulty is determining the mapping function. The kernel trick can solve this problem. To calculate the optimal hyperplane in a higher dimension, only the dot products from the kernel function have to be calculated. With the kernel trick, the map function does not need to be specified. This is graphically shown in Figure 6.4, where ϕ is the kernel function.

To use the SVM algorithm to produce likelihood ratios, the output of the classifier should be two scores. Instead of only obtaining the predicted class label, we are interested in obtaining $\mathbb{P}(c|\mathbf{x}_i)$. To obtain scores from the SVM algorithm, Platt scaling is used [40]. Platt scaling transforms the output of a classifier into a probability distribution. The method is based on fitting a logistic regression function on the output of the classifier.

An advantage of SVM is the optimality criterion for constructing the hyperplane, thus SVM is less prone to overfitting on the training data. A disadvantage is that the calculation of the dot products requires large computational power. Hence, the algorithm tends to get very slow for a large number of samples. For all calculations the SVM implementation in the *Scikit-learn* Python library is used. The default parameters are used, outlined in Appendix B.2.

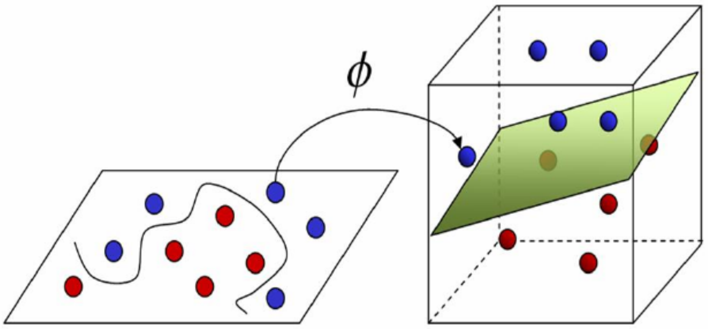


Figure 6.4: Graphical explanation of the kernel trick for support vector machines [30].

Random Forest and XGBoost

Random Forest and XGBoost are both in the class of ensemble methods, a method that combines multiple weak classifiers to one stronger classifier. In this thesis, decision trees are used as weak classifiers. Before introducing the algorithms, first the concept of a basic decision tree is explained.

A decision tree is a supervised learning method. It is a structure build of nodes and branches. Every node represents a decision, a split in the data points. The branches arising from a node represent the outcome. If a branch does not end in a next node, this is called a leaf, corresponding to a classification. The first node of the tree is called the root. The complete route from the root to a leaf is the process of classifying a feature vector. An example of a decision tree with features corresponding to speaker recognition is shown in Figure 6.5.

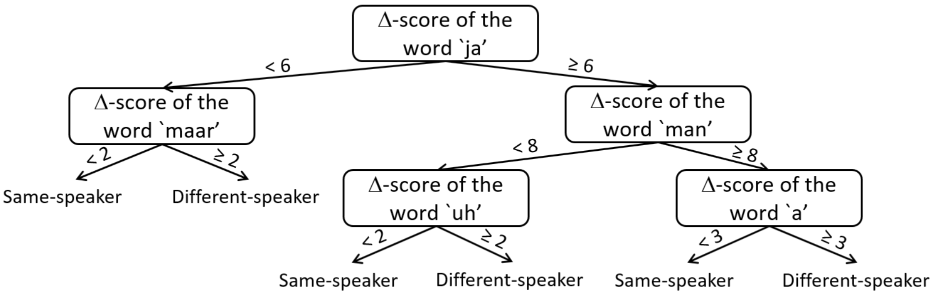


Figure 6.5: A small example of a decision tree based on the distances obtained using the element-wise metric Δ.

Decision trees are usually constructed top-down, root to leaf, using the labelled data set. In each node a feature is chosen, by evaluating this feature the data is divided. The selection of this feature is determined by minimising a cost metric. The feature which minimises the cost metric is selected. Almost all metrics quantify the homogeneity of class labels in the two resulting subsets of data, after splitting in the node. A common used method is the CART algorithm, which uses the Gini index as metric. The Gini index is defined as

$$\text{Gini index} = 1 - \sum_{i=1}^{2} p_i^2 = 1 - \left(p_{ss}^2 + p_{ds}^2\right), \tag{6.13}$$

where p_i is the chance on a sample with label i, and thus p_{ss} and p_{ds} are the probabilities on a sample with a same-speaker or different-speaker label [47]. The Gini index quantifies the homogeneity in a data set, it displays the amount of mixture of class labels. A perfect homogeneous data set gives a Gini index of 0, all data points in the subset have the same class label. A drawback of using only one decision tree as model is the high variance in the model. To solve this problem, ensemble methods use several decision trees to reduce variance.

The XGBoost algorithm and the random forest method use both the same ensemble model, using decision trees as weak classifiers. The difference between the two methods arises in the training approach. First, the XGBoost algorithm will be explained briefly, after that, the difference with the random forest method is outlined. The complete derivation of the XGBoost algorithm can be found in [8].

The mathematical structure of a tree ensemble, consisting of Z decision trees, is defined as

$$\hat{c}_i = \sum_{z=1}^{Z} f_z\left(\mathbf{x_i}\right), \quad f_z \in \mathscr{F}, \tag{6.14}$$

where $\hat{c}_i$ is the binary prediction for class c_i based on feature vector $\mathbf{x_i}$, Z is the number of trees in the ensemble, and f_z a tree in the set of decision trees $\mathscr{F}$ constructed with the CART algorithm. To define the performance of the ensemble of decision trees, an object loss function is defined. The object loss function L to be minimised is

$$L(\phi) = \sum_i l\left(\hat{c}_i, c_i\right) + \sum_z \Omega\left(f_z\right), \tag{6.15}$$

where l is the loss function and Ω the regularisation term [8]. The loss function indicates how well the model performs for the training data. This can be a standard loss function, for example the mean squared error. The regularisation term accounts for encouraging simple models, measuring the complexity of the model. Using a simple model supports stability in the predictions [8]. XGBoost trains in an additive way, this means a decision tree is added greedily. Greedy means the next tree f_t which is added, is selected by minimising the object loss function over all possible trees. Building the ensemble is an iterative process, in every iteration a new decision tree is added to the ensemble. Let $\hat{c}_i^{(t)}$ be the prediction for c_i in the t^{th} iteration. Every iteration t, adding the next tree f_t is done by minimising

$$L^{(t)} = \sum_{i=1}^{n} l\left(c_i, \hat{c}_i^{(t-1)} + f_t\left(\mathbf{x}_i\right)\right) + \Omega\left(f_t\right). \tag{6.16}$$

The remaining question is how to find the best tree to minimise the loss function. Because of the extensive amount of possible trees, it is not possible to enumerate all trees and choose the best one. Instead, one level of a tree is optimised per step, this process is further outlined in [8]. The algorithm stops iterating if adding a tree does not result in a lower object loss function.

On the contrary, a random forest classifier does not train in an additive way. In order to combine the decision trees to a random forest, a method called bagging is introduced. In this method, Z decision trees constructed with the CART algorithm are trained on a newly sampled subset of the training set. To make sure the trees are uncorrelated, for every tree only a subset of the possible features is taken into account. Averaging over different subsets of the data and different subset of features, the variance in the model is decreased. For a random forest, every decision tree is trained independently of the other trees. This is the difference in training between a random forest and XGBoost. For XGBoost, the next added tree is dependent on the already existing ensemble of trees.

The main advantage of an ensemble classifier using trees, is no preprocessing of the data is needed. The classifier also works well for non-scaled data or dependent data. A disadvantage is that the algorithm is still prone to overfitting. The XGBoost method is chosen over the random forest algorithm. The random forests algorithm needs a large amount of trees to predict accurately, whereas XGBoost uses new trees to help correct errors made by previous trees. For all calculations the XGBoost implementation in the *XGBoost* Python library is used. The default parameters are used, outlined in Appendix B.2.

6.3. Feature-based likelihood ratio as score

The third scoring method is a naive feature-based likelihood ratio used as score. The method is abbreviated to score LR, not to be confused with the definition of a likelihood ratio. This method is implemented because it takes into account similarity and typicality. No distance measure is used to concatenate the features, thus all information contained in the original features is used. Although the feature-based approach is an advantage, a disadvantage is that a large set of data is needed to correctly calculate the high-dimensional feature-based likelihood ratio. The score LR is a likelihood ratio directly calculated using

$$\mathrm{LR}(\boldsymbol{\theta}, e) = \frac{f(e|H_p, \boldsymbol{\theta})}{f(e|H_d, \boldsymbol{\theta})}, \tag{6.17}$$

where θ is the l-dimensional parameter that indexes the distribution and the likelihood functions f as specified in Section 3.1. However, instead of using the result directly as a likelihood ratio, it is used as score, given the two competing hypotheses. This approach is used, since to calculate the feature-based likelihood ratio, several assumptions are needed. This suggests the directly calculated likelihood ratios are probably ill-calibrated. The assumptions are specified later in this section. The scores are calibrated to obtain a likelihood ratio and hence this method is again a score-based approach. Two distributions are needed: the within-speaker and between-speaker distribution. Because no underlying distribution is known, a two-level normal-normal model is proposed. This model is often used in forensic sciences, for example for modelling substances in drugs tablets or composition analysis of glass fragments [2, 11]. First the two-level normal-normal model is explained generally, afterwards the implementation for speaker recognition is outlined.

6.3.1. Two-level normal-normal model

A two-level model is a statistical model which is appropriate to use when data has a hierarchical structure with more than one level. In our case the speech fragments are nested in different speakers. The distribution of speech fragments from the same speaker is defined as the within-speaker distribution and the distribution of speech fragments from two different speakers is defined as the between-speaker distribution. The two-level normal-normal model assumes the within-speaker and between-speaker distributions follow multivariate normal distributions. To make this more precise, assuming a normal distribution for the within- and between-speaker distribution, results in the hierarchy shown in Figure 6.6.

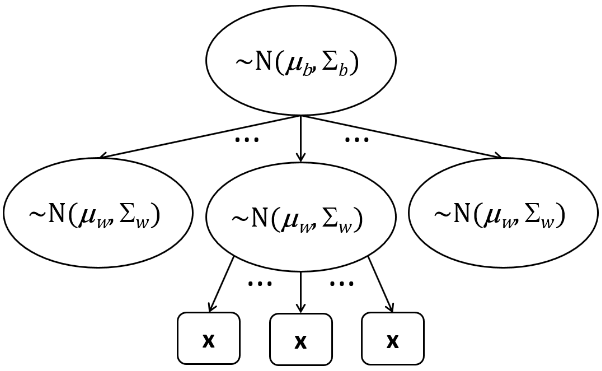


Figure 6.6: Structure of the two-level normal-normal distribution assumed in the background population of alternative speakers, the feature samples are denoted as x, based on [31] (Fig 2, p. 48).

Here $\boldsymbol{\mu}_b$ is the l-dimensional vector with the overall mean values per feature and $\boldsymbol{\Sigma}_b$ is the covariance matrix from the between-speaker distribution. $\boldsymbol{\mu}_w$ is the l-dimensional vector with mean values per feature per speaker, the within-speaker mean, and $\boldsymbol{\Sigma}_w$ the covariance matrix from the within-speaker distribution. The within-speaker mean is inherited from the overall mean, thus the parameters $\boldsymbol{\mu}_b$, $\boldsymbol{\Sigma}_b$ and $\boldsymbol{\Sigma}_w$ have to be estimated. With $\mathbf{y}_{i,j}$ defined as a sample j of speaker i and $\mathbf{P}_i$ the speakers in the background population, the two-level normal-normal model is defined as

$$\begin{aligned} &\mathbf{P}_i \sim N(\boldsymbol{\mu}_b, \boldsymbol{\Sigma}_b) \\ &\mathbf{y}_{i,j}|\mathbf{P}_i = \mathbf{p}_i \sim N(\mathbf{p}_i, \boldsymbol{\Sigma}_w), \end{aligned} \tag{6.18}$$

where $\mathbf{p}_i$ is the realisation of $\mathbf{P}_i$. For the common source model all data is assumed to follow a two-level model. For the specific source model, instead of sampling from an unknown speaker, the sample is obtained from a known specific speaker. This indicates the model is a one-level model, with $\mathbf{y}_{s,j}$ defined as sample j of the specific speaker s, represented as

$$\mathbf{y}_{s,j} \sim N(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s), \tag{6.19}$$

where $\boldsymbol{\mu}_s$ is the specific speaker mean vector and $\boldsymbol{\Sigma}_s$ the covariance matrix of the specific speaker distribution.

The background data set of alternative speakers is used to estimate the parameters indexing the distributions. It is challenging to estimate the high dimensional parameters $\boldsymbol{\mu}_b$, $\boldsymbol{\Sigma}_b$, $\boldsymbol{\Sigma}_w$, $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$ correctly. The following assumptions are used to implement the two-level normal-normal model for the speaker recognition case and estimate the parameters.

- The frequentist approach is used, the parameters $\boldsymbol{\mu}_b$, $\boldsymbol{\Sigma}_b$, $\boldsymbol{\Sigma}_w$, $\boldsymbol{\mu}_s$ and $\boldsymbol{\Sigma}_s$ are estimated and plugged into the equation.

- The overall mean $\boldsymbol{\mu}_b$ can be estimated using a weighted or unweighted mean [25]. In this thesis, only the weighted approach is used, because of ease of implementation. The overall mean is estimated by $\boldsymbol{\mu}_b = \frac{1}{\sum_i n_i} \sum_{i=1}^{m} \sum_{j=1}^{n_i} \mathbf{y}_{ij}$, where n_i is the number of samples per speaker i and m the total number of speakers. It is assumed no cross-variance is present and the between-speaker covariance matrix $\boldsymbol{\Sigma}_b$ is a diagonal matrix. This implies all features are independent from each other. The variance per feature is determined. To calculate the between-speaker statistics, the *Standardscaler* implementation in the *sklearn* library in Python is used.
- The within-speaker covariance matrix $\boldsymbol{\Sigma}_w$ is estimated by first estimating the sample variance per feature per speaker i, denoted as σ_i^2. The within-speaker variance is the mean value of all within-speaker variances $\frac{1}{m} \sum_i^m \sigma_i^2$, called the pooled within-speaker variance, following [31].
- For the specific source model, the mean vector $\boldsymbol{\mu}_s$ is estimated from the samples of the specific speaker. Because in the speaker recognition case often a small amount of data is available, the specific speaker covariance matrix $\boldsymbol{\Sigma}_s$ is estimated by the within-speaker distribution covariance matrix $\boldsymbol{\Sigma}_w$ of the background population of alternative speakers.
- The assumption of normality can be validated using QQ-plots or a goodness of fit test, this is outside the scope of this thesis.

All assumptions above suggest that a directly calculated likelihood ratio is probably ill-calibrated. Thus, the score-based approach is adopted, to investigate if the advantages of feature-based approach can still be used. The functions to calculate the likelihood ratio from Section 3.1 together with the specified Gaussian distribution, result in two equations for the likelihood ratio. One for the specific source model and one for the common source model. They are presented in the two sections below, together with the implementation for a score-based approach.

6.3.2. Common source score calculation

For the common source model, Equations (3.8) and (3.9) are used to calculate the likelihood ratio score. Using the specified normal distributions for the within-speaker and between-speaker distributions, the common source likelihood ratio is given as

$$\begin{aligned} LR_{CS}\left(\boldsymbol{y}_{u_1}, \boldsymbol{y}_{u_2}, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w, \boldsymbol{\mu}_b\right) &= \\ &|\boldsymbol{\Sigma}_b|^{1/2} \left|\boldsymbol{\Sigma}_b^{-1} + \boldsymbol{\Sigma}_w^{-1}\right| \left|\boldsymbol{\Sigma}_b^{-1} + 2\boldsymbol{\Sigma}_w^{-1}\right|^{-1/2} \exp\left[\frac{1}{2} \boldsymbol{\mu}_b^T \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b\right] \\ &\cdot \exp\left[\frac{1}{2}\left(\sum_{j=1}^{2} \mathbf{y}_{u_j}^T \boldsymbol{\Sigma}_w^{-1} + \boldsymbol{\mu}_b^T \boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1} + 2\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1} \sum_{j=1}^{2} \mathbf{y}_{u_j} + \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b\right)\right] \\ &\cdot \exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_1}^T \boldsymbol{\Sigma}_w^{-1} + \boldsymbol{\mu}_b^T \boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1} + \boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1} \mathbf{y}_{u_1} + \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b\right)\right] \\ &\cdot \exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_2}^T \boldsymbol{\Sigma}_w^{-1} + \boldsymbol{\mu}_b^T \Sigma_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1} + \boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1} \mathbf{y}_{u_2} + \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b\right)\right], \end{aligned} \tag{6.20}$$

where $(\boldsymbol{y_u} = (\boldsymbol{y}_{u_1}, \boldsymbol{y}_{u_2}))$ are the two feature vectors resulting from the unknown speech transcriptions, $\boldsymbol{\mu}_b$, $\boldsymbol{\Sigma}_b$ and $\boldsymbol{\Sigma_w}$ are the estimated between-speaker overall mean vector, the between-speaker covariance matrix and the within-speaker covariance matrix, respectively. The full derivation can be found in Appendix B.1.1

Instead of directly using the result as likelihood ratio, scores are calculated following the score-based approach. Define $\mathbf{y}_{SP_i}$ as a feature vector resulting from a transcription from speaker SP_i. The same-speaker scores s_{ss} and different-speaker scores s_{ds} are calculated according to

$$\begin{aligned} s_{\text{ss}} &= LR_{CS}\left(\boldsymbol{y}_{SP_i}, \boldsymbol{y}_{SP_j}, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w, \boldsymbol{\mu}_b\right)\Big|_{SP_i = SP_j}, \\ s_{\text{ds}} &= LR_{CS}\left(\boldsymbol{y}_{SP_i}, \boldsymbol{y}_{SP_j}, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w, \boldsymbol{\mu}_b\right)\Big|_{SP_i \neq SP_j}. \end{aligned} \tag{6.21}$$

6.3.3. Specific source score calculation

For the specific source model, Equations (3.10) and (3.11) are used to calculate the likelihood ratio score. Using the specified normal distribution for the within-speaker and between-speaker distribution and the estimated specific speaker statistics, the specific source likelihood ratio is given as

$$\begin{aligned}
&LR_{SS}\left(\boldsymbol{y_u}, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w, \boldsymbol{\mu}_b, \boldsymbol{\Sigma_s}, \boldsymbol{\mu_s}\right) = \\
&\quad |\boldsymbol{\Sigma_s}|^{-1/2} \, |\boldsymbol{\Sigma}_w|^{1/2} \, |\boldsymbol{\Sigma}_b|^{1/2} \left|\boldsymbol{\Sigma}_b^{-1} + \boldsymbol{\Sigma}_w^{-1}\right|^{1/2} \\
&\quad \cdot \exp\left[-\frac{1}{2}(\mathbf{y}_u - \mu_{\mathbf{s}})^T \boldsymbol{\Sigma_s^{-1}} (\mathbf{y}_u - \mu_{\mathbf{s}})\right] \\
&\quad \cdot \exp\left[\frac{1}{2}\left(\mathbf{y}_u^T \boldsymbol{\Sigma}_w^{-1} \mathbf{y}_u\right) + \frac{1}{2}\boldsymbol{\mu}_b^T \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b\right] \\
&\quad \cdot \exp\left[-\frac{1}{2}\left(\mathbf{y}_u^T \boldsymbol{\Sigma}_w^{-1} + \boldsymbol{\mu}_b^T \boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1} + \boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1} \mathbf{y}_u + \boldsymbol{\Sigma}_b^{-1} \boldsymbol{\mu}_b\right)\right],
\end{aligned} \tag{6.22}$$

where $\boldsymbol{\mu}_s$ is the specific speaker mean vector, $\boldsymbol{\Sigma}_s$ the covariance matrix of the specific speaker distribution and the remainder of the indexing parameters are equal to the ones defined in Equation (6.20). The extensive derivation can be found in Appendix B.1.2

The scores are calculated following the score-based approach. Define $\mathbf{y}_{SP_i}$ as a feature vector resulting from a transcription from speaker SP_i. The same-speaker scores s_{ss} and different-speaker scores s_{ds} are calculated using the labelled training set as

$$\begin{aligned}
s_{\text{ss}} &= LR_{SS}\left(\boldsymbol{y}_{SP_i}, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{SP_j}, \boldsymbol{\mu}_{SP_j}\right)\Big|_{SP_i = SP_j}, \\
s_{\text{ds}} &= LR_{SS}\left(\boldsymbol{y}_{SP_i}, \boldsymbol{\Sigma}_b, \boldsymbol{\Sigma}_w, \boldsymbol{\mu}_b, \boldsymbol{\Sigma}_{SP_j}, \boldsymbol{\mu}_{SP_j}\right)\Big|_{SP_i \neq SP_j},
\end{aligned} \tag{6.23}$$

where $\boldsymbol{\Sigma}_{SP_j}$ is the specific speaker SP_j covariance matrix and $\boldsymbol{\mu}_{SP_j}$ the specific speaker SP_j mean vector.

7

Calibration of scores to likelihood ratios

This chapter contains the theory for transforming the scores obtained from a scoring method to a likelihood ratio. First non-parametric kernel density estimation is introduced. The second part of this section describes the practical implementation to calculate the likelihood ratio using the kernel density estimator.

7.1. Kernel density estimation

Independent of which scoring method is chosen from Chapter 6, the scores cannot be used as likelihood ratios directly. The transformation from scores to likelihood ratios is called calibration. All the scores per class, H_p or H_d, give rise to a probability distribution. A first impression of the distribution is given by the histogram obtained from the scores. A smooth and continuous distribution can be estimated using two approaches. The first approach is non-parametric by using a kernel density estimator. The second approach is fitting a parametric model to the scores. For some applications, it has been shown that a specific parametric model is suitable, but for scores derived from frequency feature vectors it is not self-evident. Thus the non-parametric approach called kernel density estimation will be used.

Kernel density estimation is a non-parametric method to estimate the probability density function of a random variable. The method is non-parametric, since no underlying distribution is assumed for the variable. Kernel density estimation is in some sense similar to a normalised histogram. The area under the histogram is normalised to one, the height of the bins in the histogram show the probability density. Instead of having sub-intervals, for kernel density estimation a kernel function is placed on each data point. All the obtained scores are individual data points. The different kernels at all the data points are summed up and normalised to obtain the overall distribution of the sample. Figure 7.1 shows this process for an example where a Gaussian kernel is used.

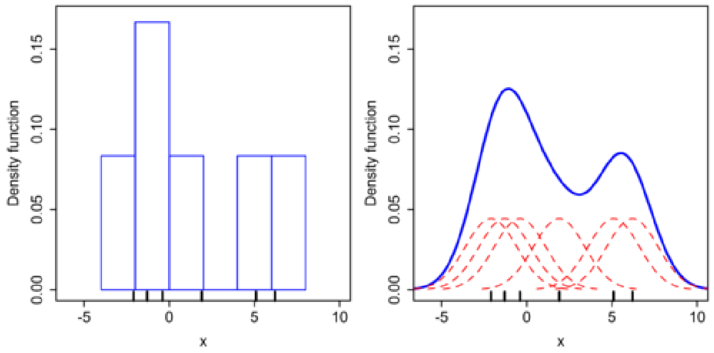


Figure 7.1: Example of a kernel density estimate. The left side of the figure shows a histogram where the area of the histogram is normalised to one. The right side of the figure shows the corresponding kernel density estimate in blue, the red curves are the single kernels [57].

To define this process more precisely, assume we have obtained the scores $(x_1,\ldots,x_n)$, which are generated by an unknown density function F. Then the estimator of F, denoted as $\widehat{F}_h$, is defined as

$$\widehat{F}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right), \tag{7.1}$$

with h the bandwidth of the kernel and K the kernel function [57]. A kernel function can be any function that integrates to one

$$\int K(x)dx = 1. \tag{7.2}$$

Often used kernels are a Gaussian function or a uniform function. In this thesis, we use a Gaussian kernel function with zero mean and standard deviation one, defined as

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right). \tag{7.3}$$

The standard Gaussian kernel is widely used for kernel density estimates.

The bandwidth of the kernel distribution determines the scale of the kernel function and has a strong effect on the estimated distribution. For a large bandwidth, the mass of the kernel function is widely spread around the data points. This could result in a density estimate which is over smoothed. For a small bandwidth, the mass is concentrated closely around the data points. This could lead to a result which is under smoothed. To choose the parameter h in an optimal way, different criteria are used to estimate an appropriate bandwidth. The most common criterion is choosing h in such a way that the mean integrated squared error (MISE) is minimised. The MISE is defined as

$$\text{MISE}(h) = \text{E}\left[\int \left(\hat{f}_h(x) - f(x)\right)^2 dx\right]. \tag{7.4}$$

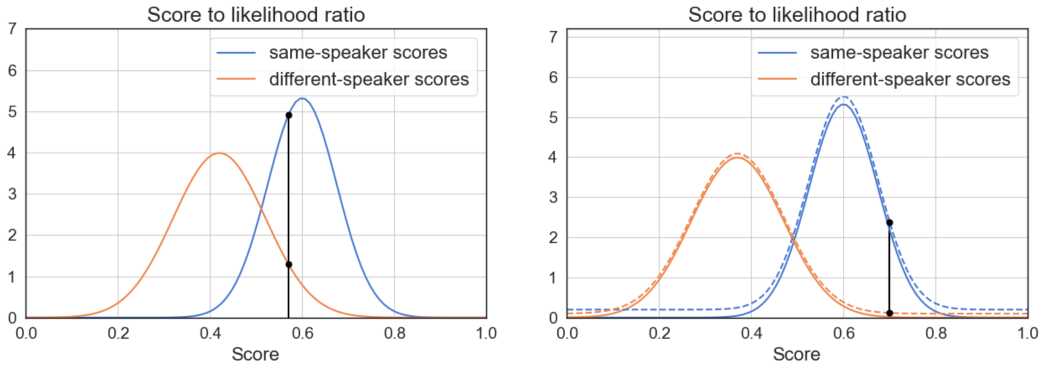
Here $\hat{f}_h$ is the estimated density function and f the unknown, real density function [57]. Because the unknown density function is used in the function, this criterion cannot be used directly. Hence, several empirical approaches are defined to practically estimate the parameter h. In this thesis Silverman's rule of thumb is used to determine the bandwidth, defined as

$$\hat{h} = \left(\frac{4\hat{\sigma}^5}{3n}\right)^{\frac{1}{5}}, \tag{7.5}$$

with $\hat{\sigma}$ the standard deviation of the sample and n the number of samples [17]. The rule of thumb performs well for estimating distributions which approach a Gaussian distribution, but can become inaccurate for distributions not similar to a Gaussian distribution. This is a point which has to be kept in mind, when calculating the results and inspecting the density estimation.

7.2. Score calibration to likelihood ratio

With kernel density estimation the distribution of scores under H_p and H_d is estimated. It is important to look at the goodness of the fit, a poor fit does not represent the underlying data. Furthermore, an inaccurate fit can lead to instability of the likelihood ratio method, as the density estimation does not follow the underlying data. Looking back at Equations (6.23) and (6.21), calculating the likelihood ratio for the disputed transcription follows directly and is illustrated in Figure 7.2a.



(a) Example of two score distributions and the vertical line of the score of a new disputed sample.

(b) Example of two score distributions (solid line) and the same distributions with the added base value B_{class} (dotted line).

Figure 7.2: Two score-based probability density distribution for the original estimation (a) and by adding a base value (b).

The probability density distributions in Figure 7.2a go to zero for values of scores that are not present in the score sample. For example, if the score obtained from the disputed transcription is 0.8, which is in the tail of the same-speaker distribution, the different-speaker distribution is zero. To calculate the likelihood ratio, using Equation (6.21) or (6.23), the denominator is the probability for the score given the defence hypothesis, the different-speaker distribution. However, this probability is zero, which results in a likelihood ratio of infinity. This over estimates the evidence, as a likelihood ratio of infinity can almost never reflect the evidence. To avoid this problem, the base value is defined, based on the number of samples used to fit the kernel density distribution, defined as

$$B_{class} = \frac{1}{S_{\#class}}, \tag{7.6}$$

where $S_{\#class}$ is the number of samples of a class in the training set. The base value thus increases for smaller number of samples $S_{\#class}$ used to fit the distribution, as a smaller number of samples leads to a less stable density estimate. The base value differs for the same-speaker and different-speaker score distribution, because the density estimation is based on different number of samples $S_{\#class}$. The base value is a small number which is simply added to the total estimated density distribution. This avoids likelihood ratios of infinity or zero, because of non-occurrence in the training data set. Figure 7.2b illustrates this approach. This is a basic approach to avoid over estimating the evidence because of a finite background data set. More methods exist to prevent over estimation of the evidence. An example is the 'ELUB' method, which stands for the 'Empirical Lower and Upper Bound' of a likelihood ratio. The method bounds the absolute resulting likelihood ratio by the information that is present in the system, dependent of the number of samples available [55]. The base value method used in this research, is chosen as it has a straightforward implementation to test the overall performance of a likelihood ratio system for different training and test data.

8 Likelihood ratio validation

Validation is an important part of the likelihood ratio method. In order for the likelihood ratio method to be eligible for use in court, the method has to be validated and the results need accompanying documentation. For validation of the results obtained in this thesis, the framework defined by [29] is used. The framework is based on three important concepts: performance characteristics, performance metrics and validation criteria.

Performance characteristics are quantities that are important for the validation of the likelihood ratio method. An important aspect of a likelihood ratio is the discriminating power between the two competing hypotheses. Discriminating power is a performance characteristic of the likelihood ratio system. *Performance metrics* quantify the performance characteristic. It is essential to have a suitable metric for the characteristic. A *validation criterion* is a threshold which is set on the characteristic metric. Whether the result is below or above the threshold, determines if a likelihood ratio system can be assumed to be valid.

The use of a standard validation process is not common practice, this is a critical note on the likelihood ratio validation process [43]. A guideline has been presented in [29], but not all likelihood ratio systems constructed are validated according to this guideline. Different approaches and metrics are used to present the validation and assessment of a likelihood ratio system.

In Table 8.1 the three most important performance characteristics are outlined: accuracy, discriminating power and calibration [42]. Accuracy means the agreement between the determined likelihood ratio and the ground truth. The ground truth is the label that corresponds to the sample, H_p or H_d. A likelihood ratio is more accurate if the value supports the correct hypothesis. Discriminating power is defined as the capability of the method to distinguish between two competing hypotheses. Calibration is a property of the likelihood ratios. A good calibration refers to a set of likelihood ratios that can be interpreted as the strength of the evidence for the given competing hypotheses [16]. More performance characteristics exist, the characteristics used in this thesis are selected as they show a good indication of the performance of a likelihood ratio system. The corresponding performance metrics and the corresponding graphical representation are stated in Table 8.1.

Table 8.1: Table with performance characteristics, performance metrics and graphical representations [29].

Performance characteristic	**Performance metric**	**Graphical representation**
Accuracy	C_{llr}, accuracy	ECE plot
Discriminating power	C_{llr}^{min}	ECE plot
Calibration	C_{llr}^{cal}	PAV plot, ECE plot, Tippett plot

Note that accuracy is defined as a performance characteristic, but also a performance metric. We have chosen to add accuracy to the performance metrics proposed in the validation framework in [29]. This choice is made as it shows an intuitive measure and can be used to compare the results with other authorship analysis papers, where only the classification accuracy is presented instead of forensic likelihood ratios. Furthermore, it is interesting to look at the interaction between the accuracy and the C_{llr}. To give a clear view, the accuracy is calculated using the same number of likelihood ratios per class H_p and H_d. The accuracy is defined as the number of likelihood ratios that support the correct hypothesis divided by the total number of calculated likelihood ratios.

The validation criteria are not explicitly stated. Since this thesis is an exploratory study to research the question whether value of evidence is present in spoken transcribed text, all results are averaged for several speakers and samples. In further research, it will be interesting to determine the validation criteria. The validation criteria can be tested in a forensic case study where the likelihood ratio method is developed for a specific test.

In the remainder of this chapter, the performance metrics and graphical representations are outlined. First the pool adjacent violators algorithm is introduced, after this the cost likelihood ratio, the ECE plot and the Tippett plot are presented.

8.1. Pool adjacent violators algorithm

The pool adjacent violators algorithm (PAV) is a calibration method which can used to calibrate scores to likelihood ratios [10]. Besides transforming scores to likelihood ratios, likelihood ratios themselves can also be transformed and calibrated in an optimal way. The PAV algorithm has been proven to give optimal calibration for binary scoring methods for a given data set [7]. As it is assumed optimal calibration for a given data set, the calibration is overfitted on this data set. Thus the calibration method is not used in a forensic case to determine the original likelihood ratios, only to determine the theoretical best calibrated likelihood ratios. For the research in this thesis, PAV calibrated likelihood ratios can be assumed to result in an optimally calibrated likelihood ratio system, given the underlying data. The PAV algorithm is an algorithm for isotonic regression. Isotonic regression fits a non-decreasing function to a given set of data points. PAV is an iterative algorithm that solves the isotonic regression problem in linear time. The algorithm is based on merging two data points if the monotonicity constraint is not satisfied. This process is iterated until a monotone function is obtained, a graphical example can be found in Figure 8.1.

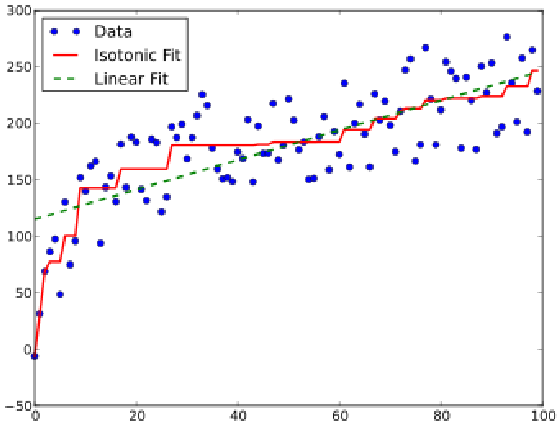
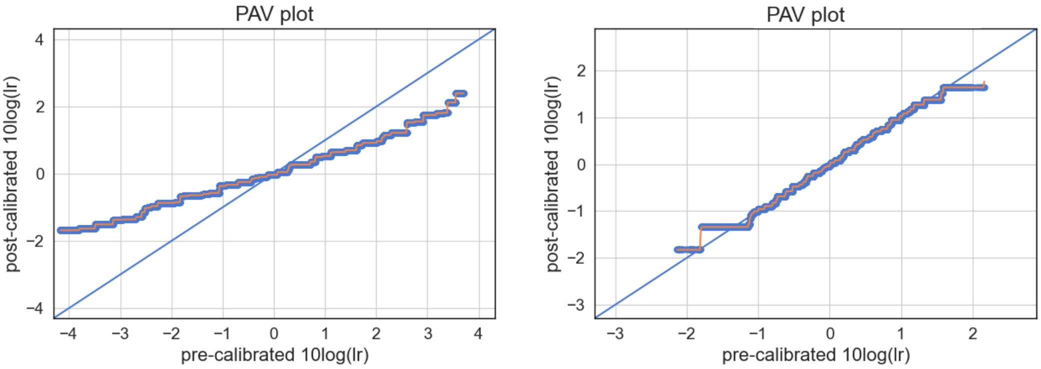


Figure 8.1: An example of a isotonic regression, the problem the PAV algorithm solves in linear time [58].

The difference between the original likelihood ratios and the PAV likelihood ratios can be visualised by plotting the values against each other. Figure 8.2 shows two PAV plots with on the x-axis the original calculated log-likelihood ratios, against the calibrated PAV log-likelihood ratios on the y-axis. Figure 8.2a shows a PAV plot which indicates bad calibration. The PAV calibrated likelihood ratios are consistently weaker than the original likelihood ratios. It shows the evidence is overestimated. An original likelihood ratio of 10^3 is optimal calibrated to a significantly lower likelihood ratio of 10^2. Both point to the prosecutor's hypothesis H_p, but the original likelihood ratio is overestimated. Figure 8.2b shows a PAV plot which indicates good calibration. The original log-likelihood ratios are almost equal to the PAV transformed log-likelihood ratios. This indicates no over- or underestimation of the value of evidence.



(a) PAV plot indicating a poor calibration.

(b) PAV plot indicating a good calibration.

Figure 8.2: Two PAV plots, showing the difference between indicating a non-conservative calibration and a good calibration.

8.2. Log-likelihood-ratio cost

The log-likelihood-ratio cost, after this denoted as C_{llr}, is based on a strictly proper scoring rule [29]. This means the measure shows better performance if the likelihood ratio points to the correct hypothesis. It is a performance metric for the performance characteristic accuracy. It is defined as cost, thus a lower value indicates a better performance. For the likelihood ratio framework, before a decision can be based on the posterior odds, also a prior $\frac{\mathbb{P}(H_p)}{\mathbb{P}(H_d)}$ has to be stated by the legal experts (Equation (3.5)). The C_{llr} is defined as the cost of a decision, for an uniform prior odds of 1. It is argued that information is lost by compressing the cost of a decision to one value, by averaging over all priors. Nonetheless, it gives a clear measure to compare likelihood ratio systems, since the priors are often not known. In [16], the C_{llr} is defined as

$$\begin{aligned} C_{llr} = \frac{1}{2 \cdot N_p} & \sum_{x:LR(H_p=True)} \log_2\left(1+\frac{1}{LR(x)}\right) \\ + \frac{1}{2 \cdot N_d} & \sum_{x:LR(H_d=True)} \log_2\left(1+LR(x)\right), \end{aligned} \tag{8.1}$$

where the likelihood ratio is abbreviated to LR. N_p and N_d are the number of likelihood ratio values with a label that corresponds to accepting H_p or H_d. The indices under the sum represent the condition on the label of the likelihood ratios values, H_p or H_d. The equation shows the cost increases if a likelihood ratio points in the wrong direction. To give an intuitive feeling for the C_{llr} value, two hypothetical examples are given. Assume the likelihood ratios do not have any discriminating power, all likelihood ratios are equal to 1. Using Equation (8.1) this results in $C_{llr} = 1$. This shows that for $C_{llr} = 1$ no information is added and a $C_{llr} > 1$ is worse than adding no information. Assume the likelihood ratios shows large discriminating power, with likelihood ratios for H_p of $1{,}000{,}000$ and for H_d of $\frac{1}{1{,}000{,}000}$, assume equal samples per class. Using Equation (8.1) this results in $C_{llr} = 1.4 \cdot 10^{-6}$. This shows an C_{llr} approaching 0 shows a perfect separation between the two competing hypotheses with strong likelihood ratios.

The C_{llr} can be divided in different parts, most commonly used are the C_{llr}^{min} and C_{llr}^{cal}. They give rise to the information loss and costs in terms of discriminating loss and calibration loss and are further outlined in Section 8.3. The relation to the C_{llr} is given by

$$C_{llr} = C_{llr}^{min} + C_{llr}^{cal}. \tag{8.2}$$

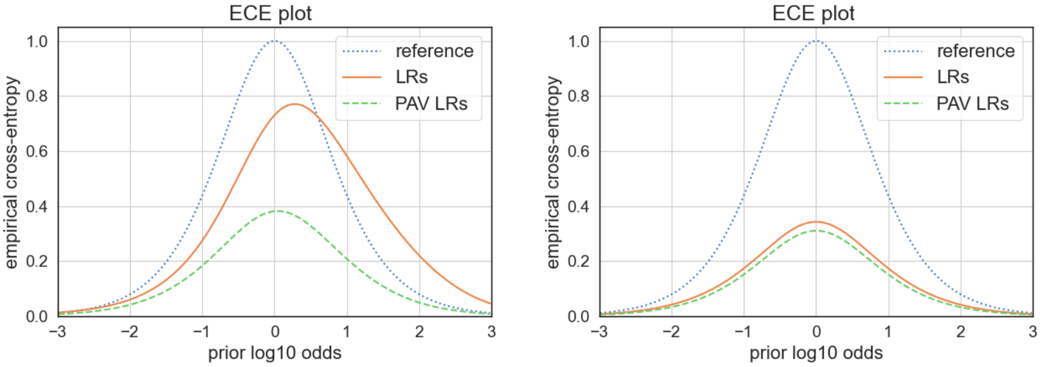
8.3. Empirical cross entropy plot

The ECE plot denoted in Table 8.1, stands for the empirical cross entropy (ECE) plot. Similar to the C_{llr}, the ECE decreases if the likelihood ratios point to the correct decision. The C_{llr} value can be seen as the summary of the ECE plot, the C_{llr} is equal to the ECE at uniform prior 1 ($\mathbb{P}(H_p) = 0.5$, $\mathbb{P}(H_d) = 0.5$). ECE is defined as the information needed for a specific prior to support the correct hypothesis for a set of likelihood ratios. In [44], the ECE is defined as

$$ECE = +\frac{\mathbb{P}\left(H_p\right)}{N_p}\sum_{x:LR(H_p=True)}\log\left(1+\frac{1}{LR(x)\cdot\frac{\mathbb{P}(H_p)}{\mathbb{P}(H_d)}}\right) + \frac{\mathbb{P}\left(H_d\right)}{N_d}\sum_{x:LR(H_d=True)}\log\left(1+LR(x)\cdot\frac{\mathbb{P}\left(H_p\right)}{\mathbb{P}\left(H_d\right)}\right), \tag{8.3}$$

where $\frac{\mathbb{P}(H_p)}{\mathbb{P}(H_d)}$ is the prior probability. To generate an ECE plot, the ECE is plotted as a function of the prior [44]. The ECE values are computed for a range of prior odds, centred around the uniform prior odds 1, thus a log prior odds of zero.

Figure 8.3 shows an example of two ECE plots. The x-axis corresponds to the log prior odds. The y-axis shows the ECE values. The blue, dotted line refers to the baseline, where the likelihood ratio values are all 1. No information is contained in these likelihood ratios. As earlier introduced in the small example, indeed for the uniform log prior odds of zero, $ECE = C_{llr} = 1$. The calculated likelihood ratios are presented by the solid, orange line. The green, striped line is labelled as 'PAV LRs', these are the resulting likelihood ratios after optimal calibration using the PAV algorithm. The calculated likelihood ratios of the constructed system should have a better performance than the reference baseline for all prior odds. Figure 8.3a shows a very badly calibrated likelihood ratio system, for a prior larger than 0.8 the likelihood ratio method does not contribute effective information. The calculated likelihood ratios differ substantially from the PAV calibrated likelihood ratios. Figure 8.3b shows a better calibrated system, where the empirical likelihood ratios agree better with the PAV likelihood ratios. For all priors, the calculated likelihood ratios perform better than the baseline.



(a) ECE plot indicating a poor calibration.

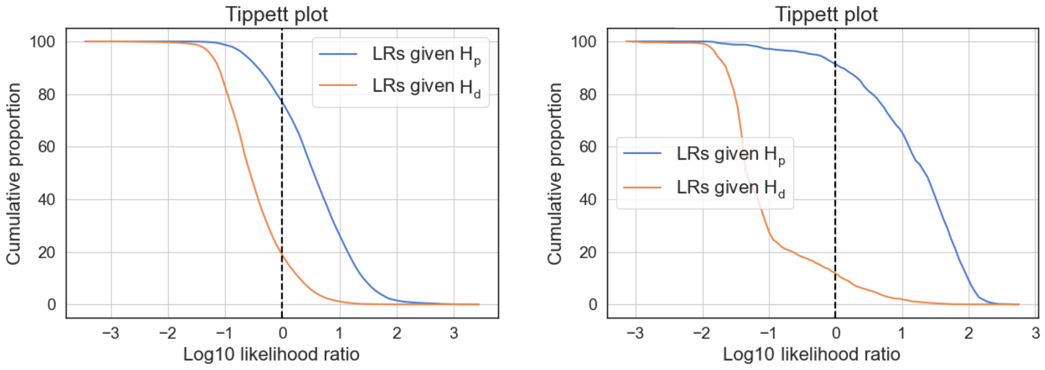
(b) ECE plot indicating a good calibration.

Figure 8.3: Two ECE plots, showing the difference between indicating a poor calibration and a good calibration.

The C_{llr}, C_{llr}^{min} and C_{llr}^{cal} can be inferred from the ECE plot. All three values are defined at uniform prior $\log_1 0(\text{odds}) = 0$ (note the log scale). The C_{llr} is the ECE value of the calculated likelihood ratio values. The C_{llr}^{min} is defined as the minimum C_{llr} possible, based on the the information contained in the data and the performance of the likelihood ratio system for an optimal calibration. The ECE value of the likelihood ratio values after the PAV transform is the C_{llr}^{min}. The C_{llr}^{cal} quantifies the calibration loss. It is defined as the difference between the calculated likelihood ratio values and the likelihood ratio values after the PAV transform.

8.4. Tippett plot

A Tippett plot shows the inverse cumulative density of the log-likelihood ratio values. It is named after the research on pairs of paint flakes by Tippett [53]. Figure 8.4 shows two Tippett plots. A Tippett plot of a likelihood ratio system consists of two lines, each line shows the descend of the proportion of likelihood ratios labelled as class H_p or H_d. It gives a measure for the performance of the likelihood ratio method in terms of calibration and strength of the obtained likelihood ratios. The broader the area between the lines supporting hypotheses H_p and H_d, the more information is obtained from the evidence [16]. It also shows the misleading evidence, visible at the intersection between the inverse density plot lines and the black, vertical line at log(LR) = 0. For likelihood ratio values given H_p, all log-likelihood ratio values should be larger than 0 to point to the correct hypothesis. This means the decrease before the vertical line at log(LR) = 0 is caused by likelihood ratios which support the wrong hypothesis. Figure 8.4a shows a Tippett plot from a poor performing likelihood ratio system and a substantial amount of misleading likelihood ratios. Figure 8.4b shows a Tippett plot of a better likelihood ratio system. It shows a better calibration and less misleading likelihood ratios.



(a) Tippett plot showing a small discriminating power.

(b) Tippett plot showing a larger discriminating power.

Figure 8.4: Two Tippett plots, showing the difference between small and large discriminating power.

9 Results

In the previous sections the process to obtain likelihood ratios and the validation framework were outlined. Several different scoring methods were proposed. The goal of this section is to explore the performance of the different scoring methods. The results section is structured in the following way. The main division is between the three different types of scoring methods, the distance method, the machine learning method and the LR score method. To avoid long computation times, the results are deliberately acquired in the order of increasing complexity, as described in Chapter 6. Table 9.1 shows the three different methods and varying parameters which are presented in the results chapter. In the results chapters only the most interesting results are presented, in Appendix C additional tables are presented.

Table 9.1: Table of the different parameter settings in the result section

Method	Main results	Variations
Distance	Jensen–Shannon Common source	1. $N \in [250,\dots,1500]$ 2. $F_{\#S} \in [50,\dots,600]$ 3. $S_{\#} \in [10,\dots,5000]$
Machine learning	SVM Common source	1. $N \in [250,\dots,1500]$ 2. $F_{\#} \in [50,\dots,600]$ 3. $S_{\#} \in [100,\dots,7000]$
	XGBoost Common source	
LR score	Two-level normal-normal model Common source	1. $N \in [250,\dots,750]$ 2. $F_{\#} \in [20,\dots,75]$ 4. $S_{\#s} \in [2,\dots,10]$
	Two-level normal-normal model Specific source	

Per method the main performance metrics, the mean C_{llr} and the accuracy A, are presented for different parameter settings. The validation representations are presented for the best performing parameter settings. Note that the number of samples $S_{\#}$ for the the score-based approach is defined as the number of same-speaker and different-speaker pairs. From both classes $S_{\#}$ pairs are used, if this number of pairs is possible to obtain from the data set, which results in two times $S_{\#}$ samples. In the results chapter four new parameters are introduced. The number of speakers in the training set is denoted as A_{train} and the number of speakers in the test set is denoted as A_{test}. The two sets are disjoint. The model is trained using the speech fragments from speakers in the training set and tested using the speech fragments from speakers in the test set. To show the general performance for all different speakers, the likelihood ratio model is repeated multiple (R) times to produce stable results. For the machine learning and LR score results section, results from different model settings are presented, the model is given as M. Summarising, the parameters used are defined as

N = Length of a sample
$S_{\#}$ = Number of samples
$S_{\#s}$ = Specific speaker samples
$F_{\#}$ = Number of frequent words
M = Model setting

A_{train} = Speakers in the training set
A_{test} = Speakers in the test set
R = Number of times algorithm is repeated
δ / Δ = Distance measure
.

9.1. Performance distance method

As introduced earlier, the distance method is the least complex of the explored methods. The results of the distance method are used as a baseline for other more complex methods. First, the results for different sample lengths N and different number of frequent words $F_{\#}$ are presented. After that, the effect of the number of samples $S_{\#}$ in the training set is explored. Lastly, the results are combined and the most interesting results are outlined. The validation graphics are presented to extensively explore the performance of the likelihood ratio model. Algorithm 2 is used to calculate all results. Additional results are shown in Appendix C.1.

Algorithm 2 Distance testing method

```
Specify parameters: $\delta, A_{train}, A_{test}, N, F_{\#s}, S_{\#}, R$
Read data set
for $iter = 1$ to $R$ do
    Random draw $A_{test}$ speakers as test set $X_{test}$
    Random draw $A_{train}$ other speakers as training set $X_{train}$
    Extract feature vectors per speaker, based on $N$ and $F_{\#}$
    Pair all feature vectors to same-speaker and different-speaker pairs within $X_{train}$ and $X_{test}$
    Calculate 1-dimensional scores $S_{dist}$ from all pairs with $\delta$
    Random draw $S_{\#}$ scores per class from $X_{train}$
    Calibrate the scores from the training set using KDE
    Calculate corresponding likelihood ratio values and $C_{llr}$ statistics
    Using the KDE densities, calculate corresponding likelihood ratio values and $C_{llr}$ for the test scores
Average over $R$ rounds
Acquire results regarding the performance and validation
```

The algorithm shows the available data set is divided in two parts, a set to train the model and a set to determine the performance of the method. It is important the two sets are disjoint and no information about the test set is leaked in the training set. The performance is determined for R different training and test set splits, to show a stable overall performance. For this section, if not specified otherwise, the parameters specified in Table 9.2 are used. These parameters are used as a default setting, as they show clear results with relatively small spread in results, however these parameters do not give the best results.

Table 9.2: Table with standard parameter values for the distance method

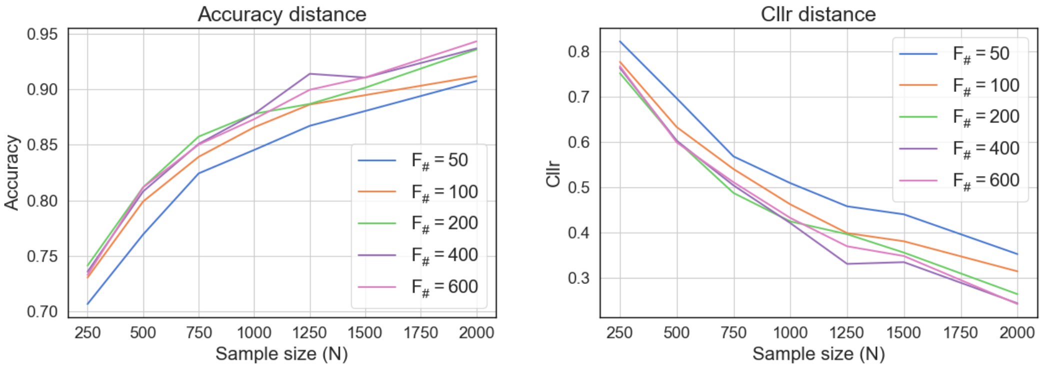
R	δ	$A_{\#train}$	$A_{\#test}$	$S_{\#}$	$F_{\#}$	N
100	Shannon-Jensen (Eq (6.4))	190	10	5000	200	750

As can be seen in Table 9.2, only results with the Jensen—Shannon distance as distance metric δ are presented in this section. This metric showed the best performance and thus is used to explore the final results for the distance method. Table 9.3 shows the mean accuracy A and C_{llr} for different metrics to support this decision. The table shows the Jensen–Shannon distance has the best performance for the given parameters. Table C.1 in Appendix C.1 shows the same results for a different parameter setting.

Table 9.3: Results in terms of mean C_{llr} and accuracy A for the distance method for different distance metrics, with $N = 500$, $F_\# = 200$ and $S_\# = 5000$.

Distance metric δ	C_{llr}	Accuracy
Jensen-Shannon	**0.61**	**0.81**
Bray-Curtis	0.62	0.81
Manhattan	0.69	0.77
Euclidean	0.71	0.77
Cosine	0.71	0.77

First, the effect of varying number of frequents words $F_\#$ and sample length N on the performance is shown in Figure 9.1. For different numbers of frequent words $F_\#$, the mean C_{llr} and accuracy A are plotted against sample length N. It shows the clear trend that for a higher sample length, the performance in terms of mean accuracy A and mean C_{llr} improves. The accuracy A increases and the C_{llr} decreases. The figures show that for smaller number of frequent words in the range between $50 \leq F_\# \leq 100$, the performance is lower than for higher number of frequent words in the range $200 \leq F_\# \leq 600$. For the range $200 \leq F_\# \leq 600$, the performance does not differ largely for varying number of frequent words.



(a) Mean Accuracy A.

(b) Mean C_{llr}.

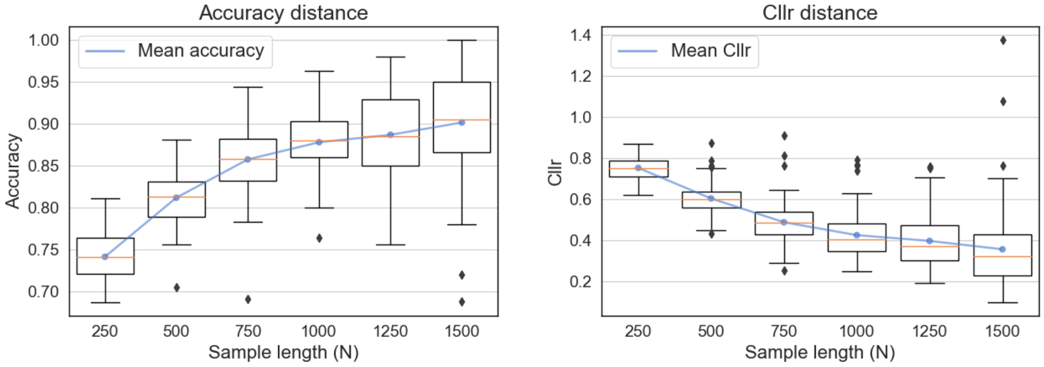
Figure 9.1: Plots of mean accuracy A and C_{llr} per sample length N for different values of number of frequent words $F_\#$ and $S_\# = 5000$.

To give a more extensive view of the two parameters N and $F_\#$, the boxplots of the sample obtained by repeating the algorithm are presented. The boxplots show the degree of spread in R repetitions. First the number of frequent words is fixed on $F_\# = 200$. This number has been chosen as this shows a good performance in the first exploratory plot. An important trade-off in these results is the number of samples $S_\#$ and the sample length N. A larger sample length leads to a smaller number of samples which can be obtained with $A_{train} = 190$ and $A_{test} = 10$. Table 9.4 shows the decrease in the number of different-speaker and same-speaker pairs for varying sample lengths. It shows the limiting factor is the number of same-speaker pairs. The number of different-speaker and same-speaker pairs used to train the method is determined in the default parameters as $S_\#$, however the numbers in the table show this amount of samples is not obtained for all sample lengths. We have chosen not to keep the number of samples static for the test of the influence of the sample length, as this is a natural effect of a limiting data set, which cannot be omitted. The parameter $S_\#$ is defined as a cap for the maximum number of used sample pairs. When the parameter $S_\#$ is varied to test the sensitivity of the method for this parameter, the sample length N is chosen such that both classes have at least $S_\#$ pairs.

Table 9.4: The data statistics corresponding to $A_{train} = 190$ and $A_{test} = 10$

Sample length	Samples from data	Same-speaker pairs	Different-speaker pairs
250	3276	28733	5338343
500	1591	6415	1259341
750	1029	2513	526788
1000	745	1234	276165
1250	573	672	163399
1500	472	416	110883
1750	386	246	74420

Figure 9.2 shows again that the mean accuracy A increases and the C_{llr} decreases for the a larger sample length. However, the graph also shows an increasing spread for an increasing sample length N. Looking at the accuracy at sample length $N = 1500$, the best result is 1, but the worst result is 0.77. This large spread could have multiple reasons. Some speakers have more distinctive speech than others. Within an automatic test case where different speakers are used, some will give better results than other speakers. Therefore, a natural spread in the results is present, resulting from the available data quality and not from the method. The larger spread could also be attributed to the fewer number of samples available. As is shown in Table 9.4, for a large sample length the data set is not sufficient to keep using $S_{\#}$ number of same-speaker and $S_{\#}$ number of different-speaker pairs. To increase the sample length more and to test adequately, more data is needed. If we look at the mean C_{llr} value, it also shows a large deviation between 0.07 and 0.7 for sample length 1500. This shows again the method is working, but has a natural spread due to the different speakers in the data set, but also due to the other reasons mentioned.

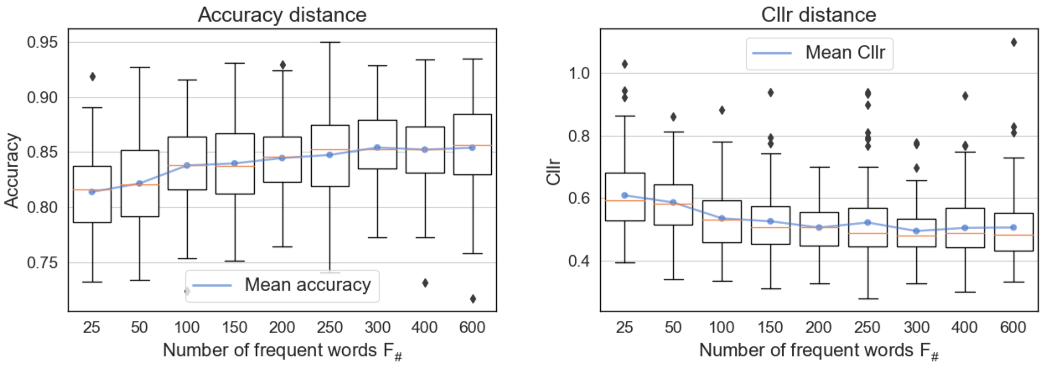


(a) C_{llr} per sample length N.

(b) Accuracy A per sample length N.

Figure 9.2: Boxplots of accuracy A and C_{llr} per sample length N for the distance method, including the mean values, with $F_{\#} = 200$ and $S_{\#} = 5000$.

To show the trend and degree of spread in the mean C_{llr} and accuracy A for different number of frequent words $F_{\#}$, in Figure 9.3 the mean C_{llr} and accuracy A are plotted against the number of frequent words $F_{\#}$ with corresponding boxplots. Note that the x-axis of the figure is not linear and thus the slope of the graph is out of proportion. The plot shows the method reaches a stable performance for $F_{\#} \geq 200$. The boxplots show that the spread is again relatively large, but no trend is shown relative to the number of frequent words $F_{\#}$. In determining the optimal number of frequent words $F_{\#}$, it is important to keep in mind that a model should not be more complex than needed. Thus, in our case it is best to take $F_{\#} = 200$ as optimal number of frequent words, as this shows almost an equal performance as a higher number of frequent words $F_{\#}$. It is also important to note that although the performance in terms of accuracy A and C_{llr} is lower for $F_{\#} = 50$, still a discriminating performance is obtained.

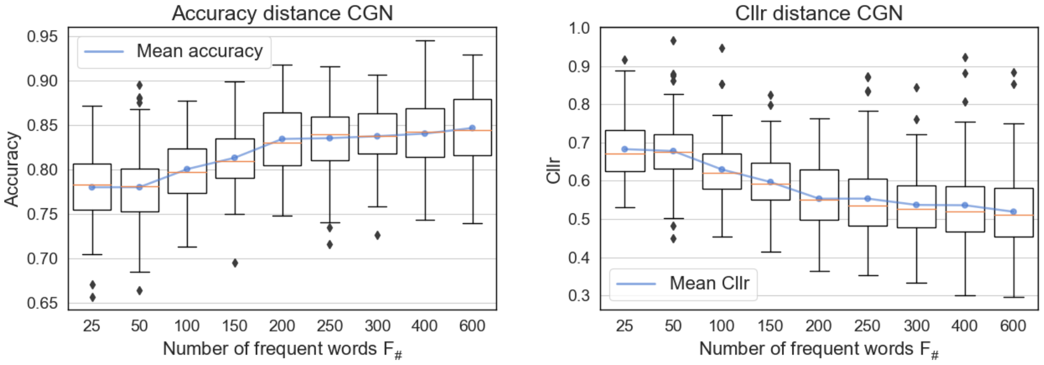


(a) C_{llr} per number of frequent words $F_{\#}$

(b) Accuracy A per number of frequent words $F_{\#}$

Figure 9.3: Boxplots of accuracy A and C_{llr} per number of frequent words $F_{\#}$ for the distance method, including the mean values, with $N = 750$ and $S_{\#} = 5000$.

To support the choice of $F_{\#} = 200$, a set of frequent words is obtained from the CGN data set, instead of the FRIDA data set. Figure 9.4 shows the trend and degree of spread in the mean C_{llr} and accuracy A for different number of frequent words $F_{\#}$ obtained from the CGN data set. The performance in terms of accuracy and C_{llr} is almost equal for the two different sets of frequent words. This shows the performance of the method is not dependent on a specific set of frequent words obtained from the same data set. This is an important result, as for forensic case studies, often also a different data set is used to obtain the frequent words. What stands out is the shift in the slope of the function before and after $F_{\#} = 200$, this supports the choice for $F_{\#} = 200$.

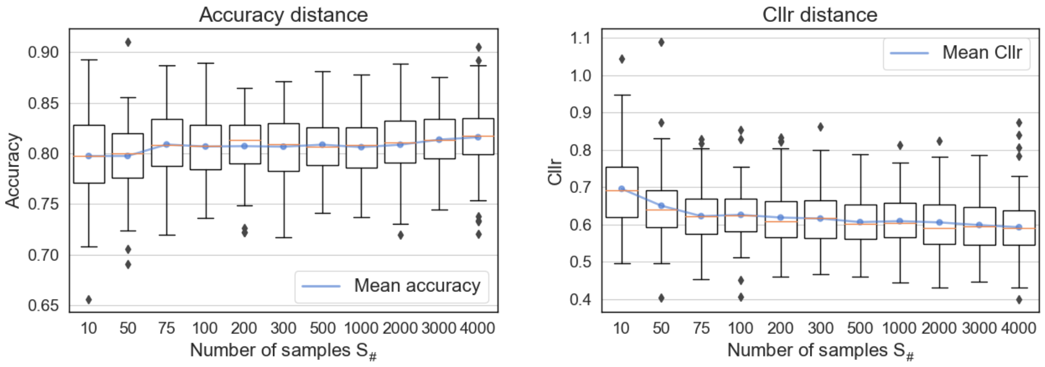


(a) C_{llr} per number of frequent words $F_{\#}$

(b) Accuracy A per number of frequent words $F_{\#}$

Figure 9.4: Boxplots of accuracy A and C_{llr} per number of frequent words $F_{\#}$ for the distance method with $N = 750$ and $S_{\#} = 5000$, including the mean values. The frequent words are obtained from the CGN data set

As third, the necessary number of samples $S_{\#}$ is tested, to test the required size of the background population data set. The number of speakers in the training set (A_{train}) is not adapted, only the number of same-speaker and different-speaker pairs used for the scoring method. Figure 9.5 shows the mean C_{llr} and accuracy A plotted against the sample length N with corresponding boxplots of the sample from repeating the algorithm. Here, the sample length $N = 500$ is chosen, for this value the number of same-speaker and different speaker pairs are sufficient to reach $S_{\#}$. Note that the x-axis of the figure is not linear and thus the slope of the graph is out of proportion.

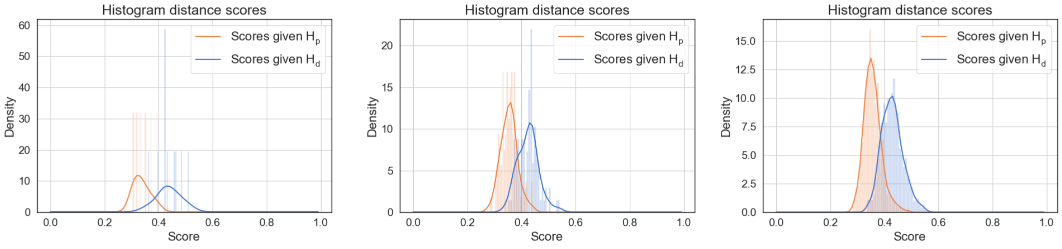


(a) Accuracy A per number of samples $S_{\#}$

(b) C_{llr} per number of samples $S_{\#}$

Figure 9.5: Boxplots of accuracy A and C_{llr} per number of samples to train $S_{\#}$ for the distance method with $N = 500$ and $F_{\#} = 200$, including the mean values.

The graph shows the number of samples $S_{\#}$ for the distance method can be surprisingly small to have an equal performance. 10 same-speaker samples and 10 different-speaker samples already give rise to a relatively good mean performance, however with a big spread in the results. For around 75 to 150 pairs, the result stabilises. The plot shows a small increase in performance for a larger number of samples, however not convincing. To explain this behaviour, we look closer at the score distributions. The good performance for a small number of samples is due to the smoothing behaviour of the kernel density estimator (KDE). For a small number of samples, a large bandwidth is used. Thus, with a small number of samples already a smooth, well estimated distribution is obtained. This is shown in Figure 9.6 for different number of samples.



(a) $S_{\#} = 10$

(b) $S_{\#} = 100$

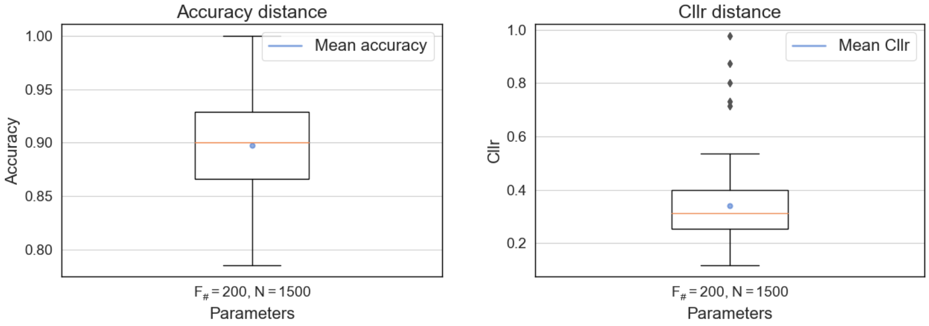
(c) $S_{\#} = 1000$

Figure 9.6: Histograms of the distance scores with corresponding KDE fit for different number of samples $S_{\#}$, with $N = 750$ and $F_{\#} = 200$.

Further important to note is that the training set in this case study fits extremely well with the test set, as they originate from the homogeneous acquired data set FRIDA. For a less suiting training set, which is almost always the case for a real forensic case, the distance method probably needs more samples to have a stable performance.

Concluding the section on the distance method, the best settings deduced from the different tested parameters are repeated to validate the system with the validation techniques outlined in Chapter 8. The parameters used are $F_{\#} = 200$, $N = 1500$ and $S_{\#} = 5000$. With these parameters, around 472 same-source samples can be obtained, sufficient for the distance method to perform well. The number of different-speaker samples is equal to $S_{\#}$. Figure 9.7 shows the two boxplots and the mean accuracy A and C_{llr} are respectively 0.90 and 0.38. The boxplots show a wide spread, this is due to the reasons specified above. This shows the performance for a real forensic case depends on the data which is available.

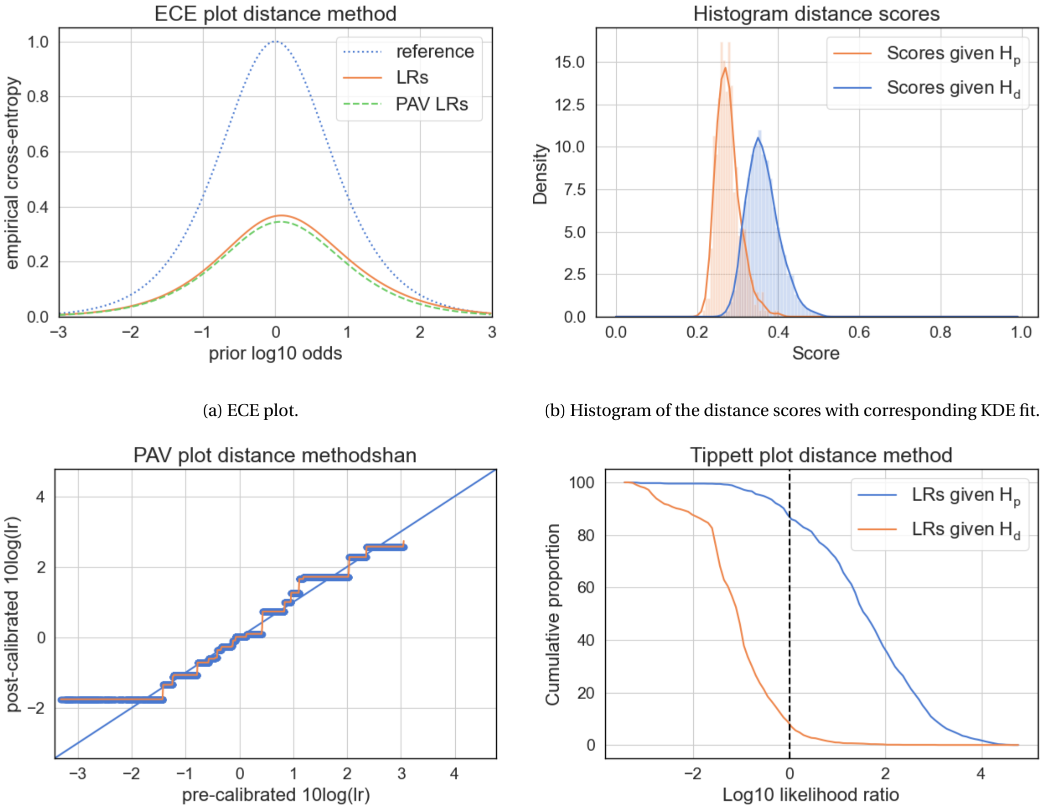
Figure 9.8 shows the validation plots. The left upper corner, Figure 9.8a, shows the ECE plot of the distance method. It shows good behaviour for all priors and a small calibration loss. The right upper corner, Figure 9.8b, shows the histogram of scores and the KDE fit of these scores. It shows that indeed the number of same-source scores is large enough to support a smooth fit. The histogram shows that the scores indeed give rise to two different distributions, but a large overlapping area. In the range where the two distributions are overlapping, the likelihood ratio values can point in the wrong direction. The lower left corner, Figure 9.8c, shows the PAV plot of the calibrated likelihood ratio values. The obtained likelihood ratio values match with the PAV calibrated likelihood ratio values. The lower right corner, Figure 9.8d, shows the Tippett plot of the likelihood ratio values. It shows the resulting likelihood ratio values range from 0.01 to 100. Concluding, the accuracy A and C_{llr} values show that the distance method can indeed work for speaker recognition, however a large sample length $N > 750$ is needed.



(a) Accuracy A.

(b) C_{llr}.

Figure 9.7: Boxplots of accuracy A and C_{llr}, including the mean values, for parameters $F_{\#} = 20$, $N = 1500$ and $S_{\#} = 5000$.



(a) ECE plot.

(b) Histogram of the distance scores with corresponding KDE fit.

(c) PAV plot.

(d) Tippett plot of the likelihood ratio values.

Figure 9.8: Validation plots of the distance method with the best working parameters $N = 1500$, $F_{\#} = 200$.

9.2. Performance machine learning methods

The second scoring method evaluated is the one using various machine learning algorithms. The algorithms add an extra layer of complexity to the system and thus have longer computation times. In Section 6, three methods are outlined, naive Bayes, support vector machines (SVM) and XGBoost. In this result section only the SVM and XGBoost results are presented, because they both outperform naive Bayes. This is surprising, as naive Bayes is often used in the literature for authorship analysis. Algorithm 3 is implemented in Python and used to calculate all results. Additional results are shown in Appendix C.2.

Algorithm 3 Machine learning testing method

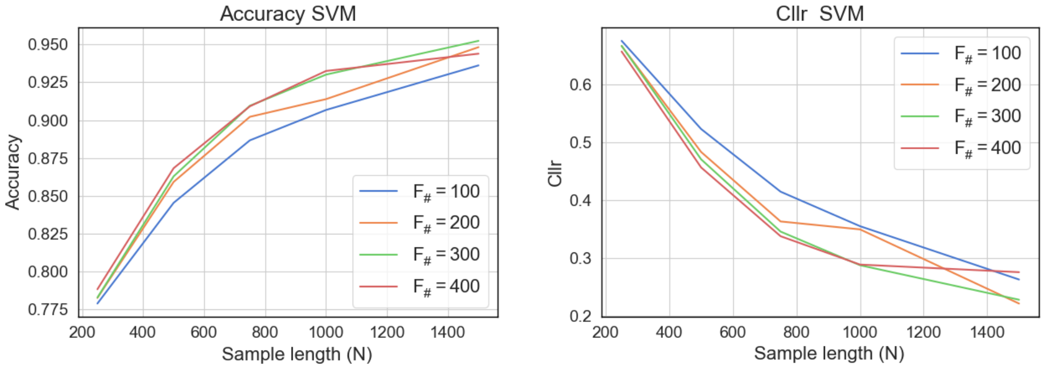
```
Specify parameters: $\Delta, A_{train}, A_{test}, N, F_{\#}, S_{\#}, R, M \in \{SVM, XGB\}$
Read data set
for $iter = 1$ to $R$ do
    Random draw $A_{test}$ speakers as test set $X_{test}$
    Random draw $A_{train}$ other speakers as training set $X_{train}$
    Extract feature vectors per speaker, based on $N$ and $F_{\#s}$
    Pair all feature vectors to same-speaker and different-speaker pairs within $X_{train}$ and $X_{test}$
    Calculate $F_{\#s}$-dimensional score vectors from all pairs with $\Delta$ for the training and test set
    Random draw $S_{\#}$ score vectors per class as training data $X_{train}$
    Train model $M$ with the training data $X_{train}$ and calculate the 1-dimensional scores $S_{ML}$
    Calibrate the calculated scores from the training set using KDE
    Using the KDE densities, calculate corresponding likelihood ratio values and $C_{llr}$ for the test scores
Average over $R$ rounds
Acquire results regarding the performance and validation
```

Table 9.5 shows the default settings which are used throughout the machine learning result section, if not specified otherwise.

Table 9.5: Table with standard parameter values for the machine learning methods

R	Δ	A_{train}	A_{test}	$S_{\#}$	$F_{\#}$	N
100	Jensen-Shannon (Eq (6.6))	190	10	5000	200	750

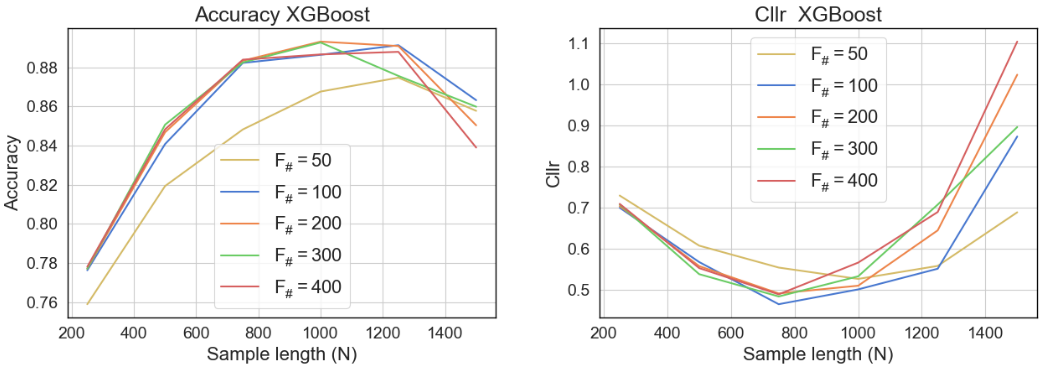
First the effect of the number of frequent words $F_{\#}$ and sample length $S_{\#}$ is explored. For different numbers of frequent words $F_{\#}$, the mean C_{llr} and accuracy A are plotted against sample length N. Figure 9.9 shows the results for the SVM method and Figure 9.9 for the XGBoost algorithm. Note the number of samples $S_{\#}$ used to train the method is an imbalanced data set for higher sample lengths. The number of different-speaker samples remains $S_{\#} = 5000$, while the number of same-speaker samples decreases as presented in Table 9.4.



(a) Mean accuracy A.

(b) Mean C_{llr}.

Figure 9.9: Plots of mean accuracy A and C_{llr} per sample length N for different values of the number of frequent words $F_{\#}$ and $S_{\#} = 5000$, with scoring method SVM.



(a) Mean Accuracy A.

(b) Mean C_{llr}.

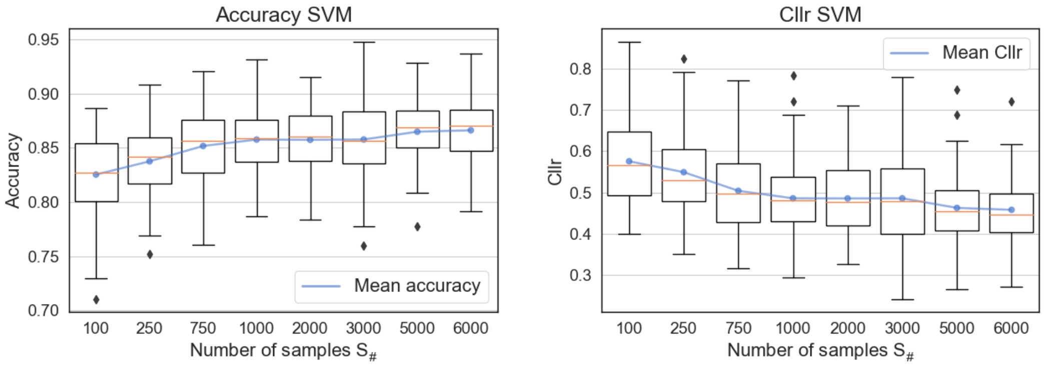
Figure 9.10: Plots of mean accuracy A and C_{llr} per sample length N for different values of the number of frequent words $F_\#$ and $S_\# = 5000$, with scoring method XGBoost.

Figure 9.9, obtained with the SVM method, shows again the clear link between sample length N and the performance of the system. The mean accuracy A increases and C_{llr} decreases for higher sample length, as expected from the results of the distance method. The best performance is again for a large sample length $N = 1500$ with $A = 0.94$ and $C_{llr} = 0.29$. This is an increase in peak performance in comparison with the distance method.

For the results of the XGBoost method in Figure 9.10 the performance also increases until $N = 750$, but drastically drops for larger sample lengths in terms of the C_{llr}. Also the increase in accuracy A stagnates. This result already shows a result related to the number of samples $S_\#$. Machine learning algorithms are trained on labelled data and need a sufficient amount of input data to train accurately. The number of samples $S_\#$ is too small to train accurately in this case. With a high sample length N, the data set is too small to keep the same number of samples to train. The figures show that this amount of labelled data is not sufficient to train the XGBoost algorithm accurately, as the performance drops for larger sample length N. It could be argued the model is overfitted, the bias-variance trade-off tends to a high variance. The parameters are kept equal for varying parameters, the method performs well for smaller samples sizes N and thus higher number of samples $S_\#$. If the variance has to be decreased, more bias is introduced, which leads to worse performance. If we look closer at Figure 9.9 obtained with the SVM method, the same trend start to show. For higher sample lengths N, the performance still increases, but not as fast as for smaller sample lengths N. Note the decreasing performance can also be caused by the class imbalance present in the data set for high sample lengths.

It is also interesting to look at smaller sample lengths instead of the peak performance at $N = 1500$. For example, both machine learning methods have a better performance at $N = 250$ and $F_\# = 200$ than the distance method. Here, the mean accuracy A and the C_{llr} (parameters $F_\# = 200$, $N = 250$) is respectively (0.73/0.76), (0.78/0.70) and (0.79/0.65) for the distance method, the XGBoost algorithm and the SVM method, respectively. SVM has the best performance in terms of accuracy A and the C_{llr}. The computation time of XGBoost is shorter than the SVM method, but longer than the distance method. Thus it is interesting to look at the further performance of the SVM and XGBoost method.

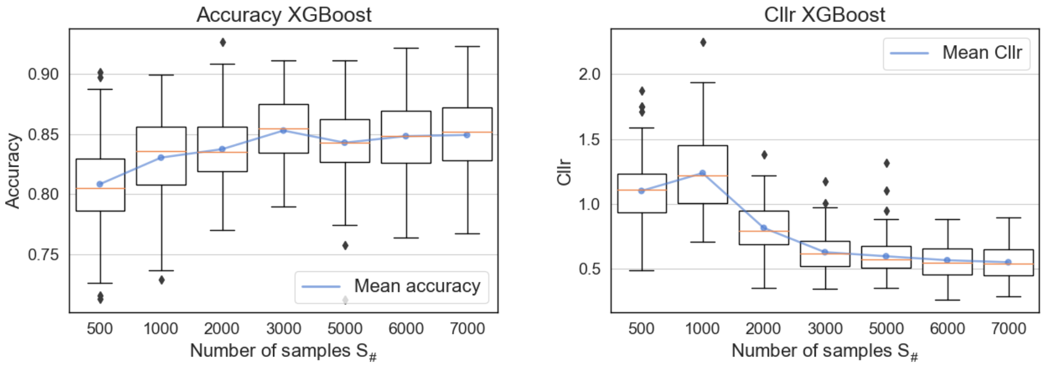
The number of samples $S_\#$ has a large impact on the performance of both algorithms. The precise effect is studied, before other parameters are tested. The next plots show the behaviour per number of samples $S_\#$ used to train the model. To test this parameter, a small sample length $N = 500$ is selected. Figure 9.12 shows the results for the XGBoost algorithm and Figure 9.11 shows the results for the SVM method. Note that the x-axis of the figure is not linear and the difference in axis spread for the XGBoost and SVM method. As expected from previous results, for XGBoost a relatively high number of samples is needed before the performance stabilises. From around $S_\# = 3000$ per class, the mean performance and the amount of spread stabilises. However, still small improvements are made by using more samples. The SVM method performs well for a small number of samples $S_\#$, for around $S_\# = 1000$ per class the performance stabilises. It has to be noted the number of samples needed is largely dependent on the hyper parameters chosen for the algorithms, specified in Appendix B.2. For other values the performance and needed number of samples $S_\#$ can change, but the results are still an indication of the order of magnitude.



(a) C_{llr} per number of samples $S_{\#}$.

(b) Accuracy A per number of samples $S_{\#}$.

Figure 9.11: Boxplots of accuracy A and C_{llr} per number of samples $S_{\#}$ for the machine learning method with SVM with $N = 500$ and $F_{\#} = 200$, including the mean values.



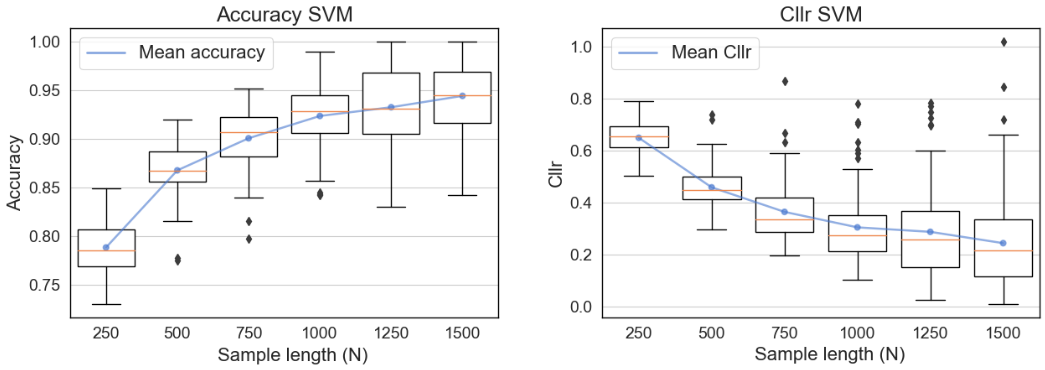
(a) C_{llr} per number of samples $S_{\#}$.

(b) Accuracy A per number of samples $S_{\#}$.

Figure 9.12: Boxplots of accuracy A and C_{llr} per number of samples $S_{\#}$ for the machine learning method with XGBoost with $N = 500$ and $F_{\#} = 200$, including the mean values.

The results show that XGBoost could have a promising behaviour, but the data set is too small to test for higher values of sample length N with a sufficient amount of samples $S_{\#}$. Therefore, the remainder of the results for the machine learning section only contain the SVM results.

To research the effect of sample length N and the spread in the results more accurately, the boxplots resulting from the repeated algorithm are presented in Figure 9.13. The performance increases for higher sample length N. Similar to the case for the distance method, the spread of the C_{llr} increases. This is due to the natural spread in the data and the decrease of the number of samples $S_{\#}$ available to train the algorithm.

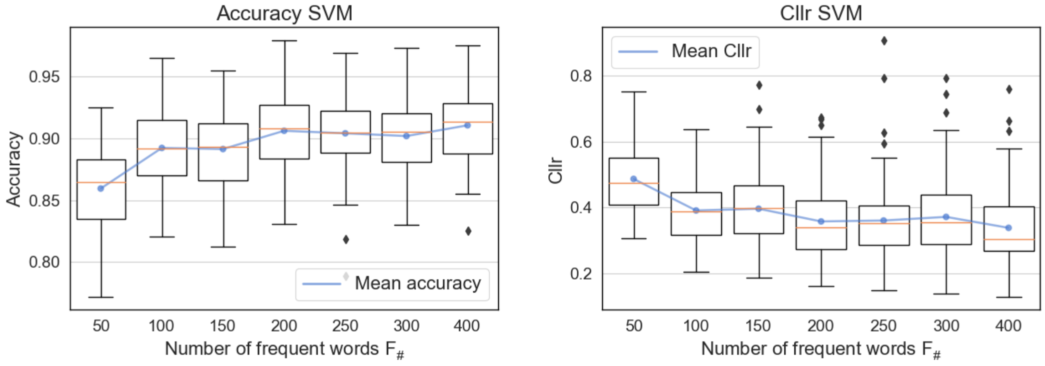


(a) C_{llr} per sample length N.

(b) Accuracy A per sample length N.

Figure 9.13: Boxplots of accuracy A and C_{llr} per sample length N for the machine learning method with SVM, including the mean values, with $F_{\#} = 200$ and $S_{\#} = 5000$.

Finally, the effect of the number of frequent words $F_{\#}$ is studied. The results are obtained for sample length $N = 750$, to ensure a consistent number of samples $S_{\#}$. Figure 9.14 shows the boxplots and mean line for the SVM method. The mean accuracy A and C_{llr} stabilises around $F_{\#} = 200$. Since using more features increases the possibility of overfitting and a simple model supports more stable predictions, $F_{\#} = 200$ is the optimal choice. Again it is noted although for smaller values of the number of frequent words $F_{\#}$ the performance decreases, it still shows a discriminating ability.



(a) C_{llr} per number of frequent words $F_{\#}$.

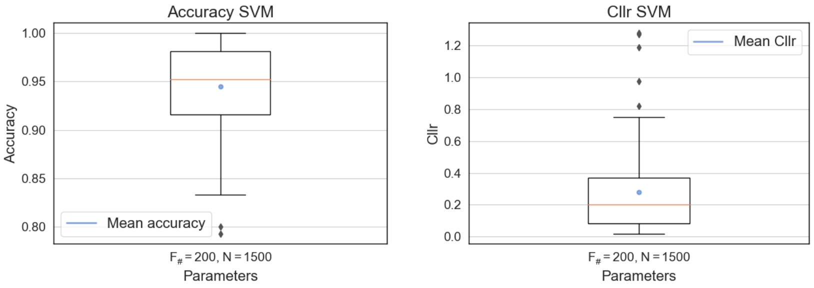
(b) Accuracy A per number of frequent words $F_{\#}$.

Figure 9.14: Boxplots of accuracy A and C_{llr} per number of frequent words $F_{\#}$ for the machine learning method with the SVM method, including the mean values, with $N = 750$ and $S_{\#} = 5000$.

Concluding the section about the results of the machine learning method, the best settings deduced from the sensitivity analysis are repeated to validate the system with the validation techniques outlined in Chapter 8. For the current data set used, SVM shows the most promising behaviour. The optimal parameters used for the SVM method are $F_{\#} = 200$, $N = 1500$ and all samples (maximal $S_{\#}$) that are possible to obtain from the total background population. Figure 9.15 shows the two boxplots, the mean C_{llr} and the accuracy A are 0.28 and 0.94, respectively. The boxplots show a wide spread, this is due to the same reasons as specified above. This shows that the performance for a real forensic case depends on the available data and the speakers present in the data set.

Figure 9.16 shows the validation plots. The left upper corner, Figure 9.16a, shows the ECE plot. The right upper corner, Figure 9.16b, shows the histogram of scores and the KDE fit of the scores. The fit is inaccurate, because the bandwidth is quite large for the sudden peak in the distribution of scores. Also, the bandwidth is not well estimated, since the score distributions is not close to a Gaussian. It does show a large discriminating power between same-speaker and different-speaker feature pairs. The left lower corner, Figure 9.16c, shows the PAV plot of the calibrated likelihood ratio values. The plot shows that for lower likelihood ratio values

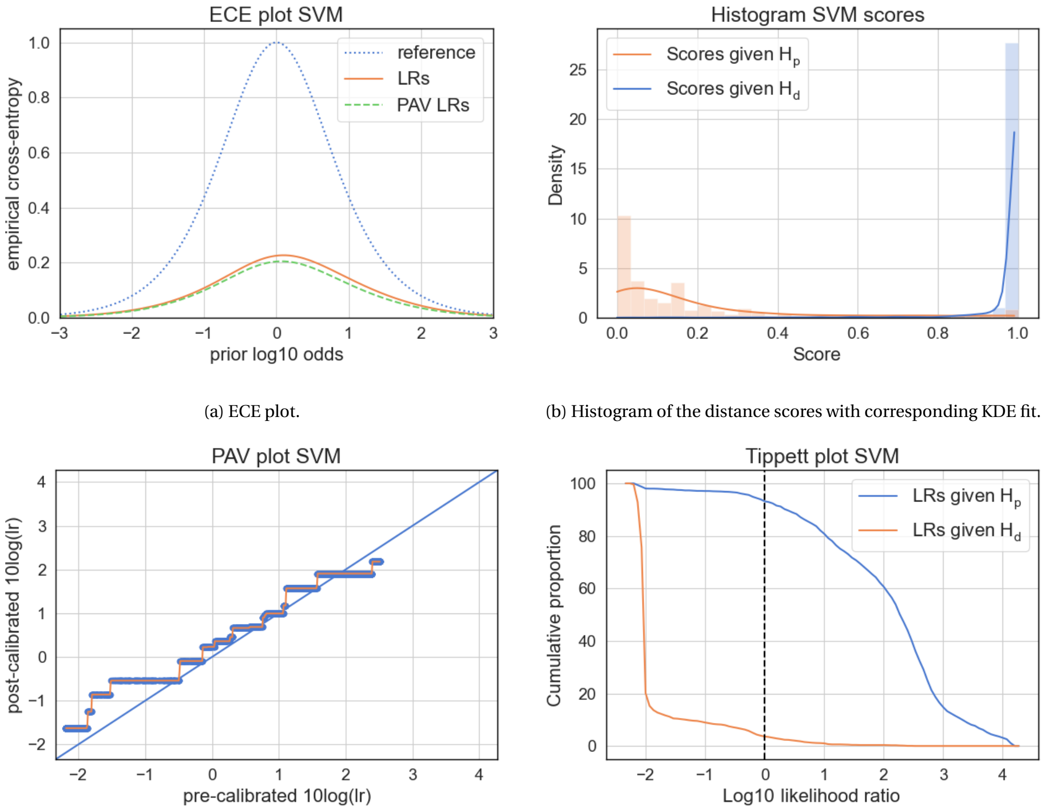
$10\log(\text{LR}) \le -1$, the likelihood ratios are overestimated in comparison with the optimal calibrated likelihood ratios. The right lower corner, Figure 9.16d, shows the Tippett plot of the likelihood ratio values. The values of the resulting likelihood ratios range from 0.05 to 100. This is slightly more conservative than the likelihood ratios obtained from the distance method. The C_{llr} values are, however, lower than for the distance method. Concluding, the accuracy A and C_{llr} values show that the SVM method performs better than the distance method for the best performing parameters. An interesting area for further research is for lower samples sizes N, where the machine learning algorithms both outperform the distance method. This area is interesting to study for a specific real forensic case, where it could lead to promising results.



(a) Accuracy A.

(b) C_{llr}.

Figure 9.15: Boxplots of accuracy A and C_{llr}, including the mean values, for parameters $F_\# = 200$, $N = 1500$ and scoring method SVM.



(a) ECE plot.

(b) Histogram of the distance scores with corresponding KDE fit.

(c) PAV plot.

(d) Tippett plot of the likelihood ratio values.

Figure 9.16: Validation plots of the SVM method with the best performing parameters $N = 1500$, $F_\# = 200$.

9.3. Performance LR score method

This chapter presents the LR score results. The chapter is divided in two sections, the common source model results and the specific source model results. First the common source model results are analysed. Afterwards, it is explored if specific source modelling with a small amount of specific speaker data leads to better performance. All results for the LR score method are obtained using Algorithm 4. Additional results are shown in Appendix C.3.

Algorithm 4 LR score

```
Specify parameters: $A_{train}, A_{test}, N, F_{\#}, S_{\#}, S_{\#s}, R, M \in \{\text{common,specific}\}$
Read data set
for $iter = 1$ to $R$ do
    Random draw $A_{test}$ speakers as test set $X_{test}$
    Random draw $A_{train}$ other speakers as training set $X_{train}$
    Extract feature vectors per speaker, based on $N$ and $F_{\#s}$
    Estimate the between-source and within-source distribution parameters $\boldsymbol{\mu}_b$, $\boldsymbol{\Sigma}_b$ and $\boldsymbol{\Sigma}_w$ from the training set
    if $M$ = common then
        Pair all feature vectors to same-speaker and different-speaker pairs according to Figure 3.2
        Calculate 1-dimensional scores $S_{LR}$ from all pairs in the training set $X_{train}$ and test set $X_{test}$ using Equation (6.21)
    if $M$ = specific then
        Random draw $S_{\#s}$ samples per speaker from the specific speaker samples
        Estimate the specific speaker distribution parameters $\boldsymbol{\mu}_s$ from the training set
        Calculate 1-dimensional scores $S_{LR}$ from all feature vectors in the training set $X_{train}$ and test set $X_{test}$ using Equation (6.23)
    Calibrate the calculated scores $S_{LR}$ using KDE
    Using the KDE densities, calculate corresponding likelihood ratio values and $C_{llr}$ for the test scores
Average over $R$ rounds
Acquire results regarding the performance and validation
```

Table 9.6 shows the default parameters which are used throughout the LR score results section, if not specified otherwise. Note that the number of speakers used in the training set A_{train} is lower than for the previous used methods. Some speakers in the data set have a relatively small set of speech fragments. For the specific source model a long fragment of speech is needed, as the data has to be divided in a training and test set for the specific speaker. Speakers with small speech fragments are not suitable for this method. As the $F_{\#}$-dimensional parameters indexing the distributions have to be estimated, a large amount of data is needed. Exploratory results showed that even when using all feature vectors obtained from A_{train}, the method can be unstable. The estimated parameters result in numerical instabilities, leading to a singular covariance matrix or under- or overflow errors. These errors occur when a calculated number is smaller than a computer can store. Therefore, all data available in the training set is used to estimate the parameters for all results presented in this chapter. Hence, the number of samples $S_{\#}$ is not specified in Table 9.6. All data corresponds to approximately 40 minutes of speech per speaker from 170 speakers.

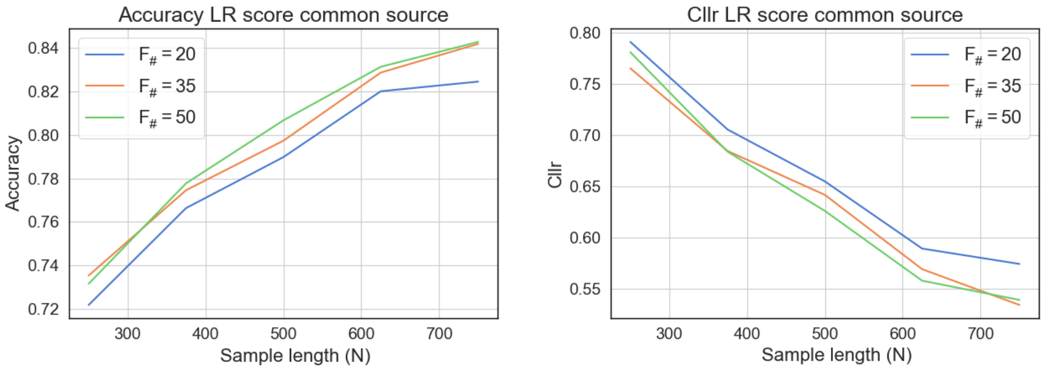
Table 9.6: Table with default parameter values for LR score results.

R	A_{train}	A_{test}	$S_{\#}$	$F_{\#}$	N	$S_{\#s}$
100	170	10	-	50	500	5

9.3.1. Common source LR score method

The common source results start with presenting the performance per sample length N for different values of the number of frequent words $F_{\#}$. Figure 9.17 shows the mean C_{llr} and the accuracy A for the common source LR score method. As expected from previous results, the performance increases for higher sample lengths N. The method is only tested for $N \in [250, 750]$, because the method is unstable for higher sample lengths N. The method implemented in Python shows numerical underflow problems. The best performance is reached for

$N = 750$ and $F_{\#} = 35$, with $A = 0.84$ and $C_{llr} = 0.54$. The results are similar to the results from the distance method. The results are worse than the result obtained with the SVM method.

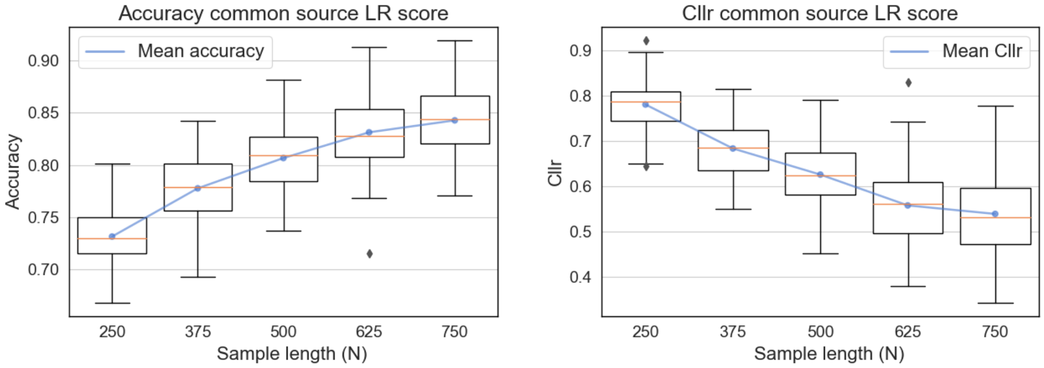


(a) Accuracy A.

(b) C_{llr}.

Figure 9.17: Plots of mean accuracy A and C_{llr} per sample length N for different values of number of frequent words $F_{\#}$ for the common source LR score method.

To further investigate the results, the spread of the results for varying sample lengths is analysed. Figure 9.18 shows the boxplots and mean line for varying sample lengths N. It shows the increasing performance for larger sample length, but also the increase in the spread of the C_{llr} values for larger sample lengths N. This result suggests the estimated distributions do not define the underlying process correctly for larger samples sizes. This can be explained by the decrease of the number of samples for larger sample lengths N. As already earlier denoted, the data set is limited, so increasing the sample length results in a smaller number of samples.

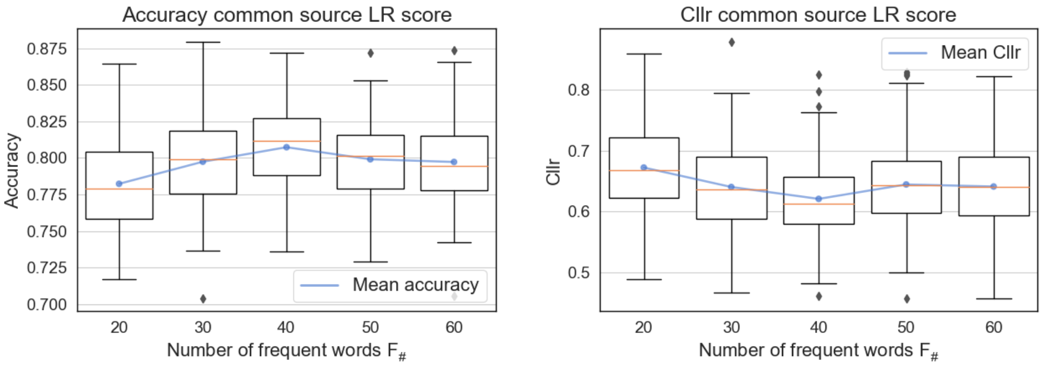


(a) Accuracy A.

(b) C_{llr}.

Figure 9.18: Boxplots of accuracy A and C_{llr} per sample length N for the common source LR score method with $F_{\#} = 50$, including the mean values.

The effect of varying the number of frequent words is shown in Figure 9.19. The plot shows a surprising result, the performance is almost stable for varying number of frequent words $F_{\#}$. The method is only tested for $F_{\#} \in [20, 60]$, because the method is unstable for higher number of frequent words. A possible explanation for the instability for higher number of frequent words $F_{\#}$, is the higher dimensional distribution that has to be estimated. The plots show the best performance at $F_{\#} = 40$, but this trend is not convincing given the spread in the results and the small improvement in performance. This result differs from results obtained with the distance method and the machine learning methods. This can be explained since the LR score method estimates the parameters of a feature-based $F_{\#}$-dimensional distribution. The results suggest trying to estimate a higher dimensional distribution adds more uncertainty and does not lead to a better performance.

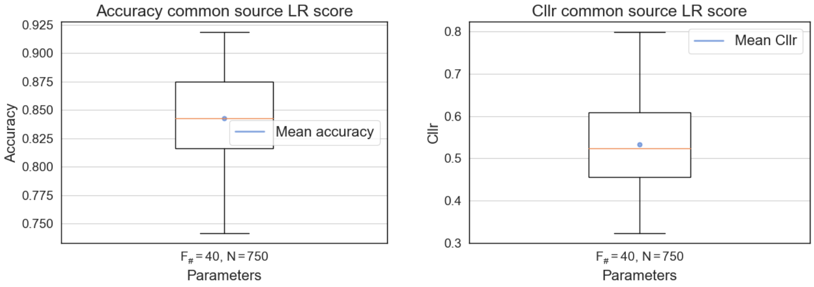


(a) Accuracy A.

(b) C_{llr}.

Figure 9.19: Boxplots of accuracy A and C_{llr} per number of frequent words $F_{\#}$ for the common source LR score method with $N = 500$, including the mean values.

The two results for varying sample lengths N and number of frequent words $F_{\#}$ show that the common source LR score method does not perform better than the baseline method. It supports the choice of complete score-based approaches, as estimating the feature-based likelihood ratio with the specified assumptions does not show better results. To complete the section on the results of the common source score LR, the best results are shown in Figure 9.20, with parameters $F_{\#} = 40$ and $N = 750$. The two boxplots and the mean accuracy A and C_{llr} are 0.84 and 0.53, respectively.

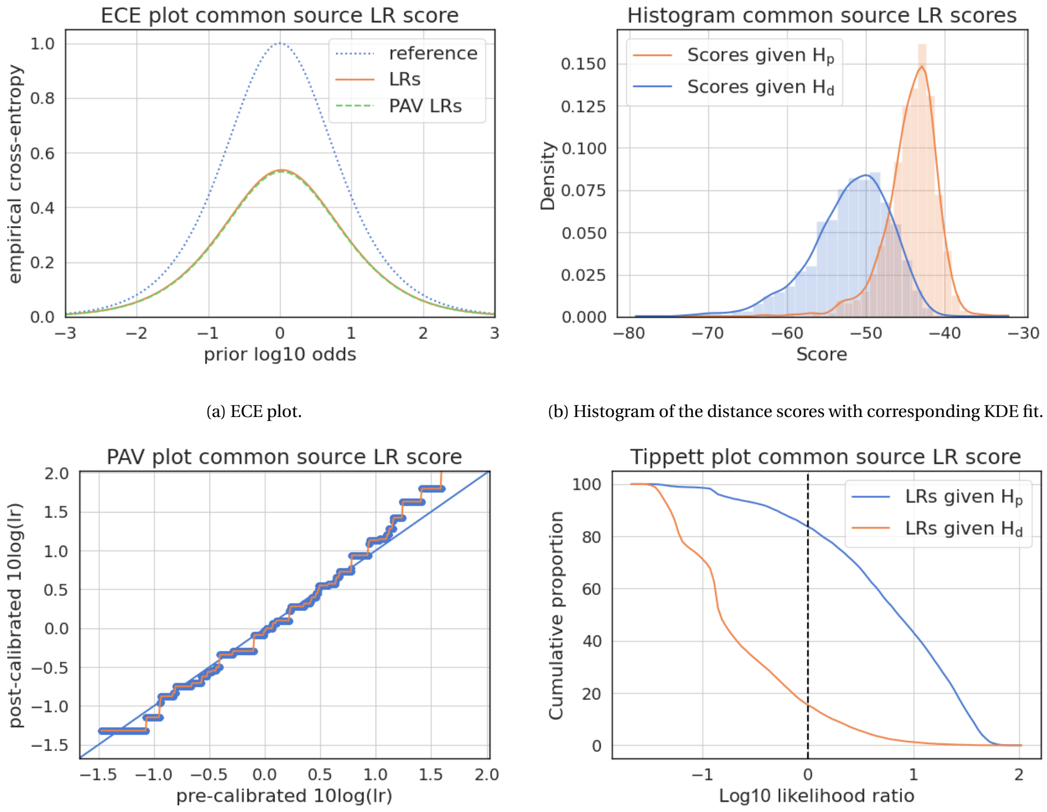


(a) Accuracy A.

(b) C_{llr}.

Figure 9.20: Boxplots of accuracy A and C_{llr}, including the mean values, for parameters $F_{\#} = 40$, $N = 750$ and scoring method common source LR score.

The validation plots for the best performing parameters are shown in Figure 9.21. The upper left corner, Figure 9.21a, shows the ECE plot. The plot shows the method is calibrated very well with almost no calibration loss. This is confirmed by the PAV plot shown in Figure 9.21c. The upper right corner, Figure 9.21b, shows the histogram of scores and the KDE fit of the scores. The fit of the two distributions is accurate. The histogram shows the scores give rise to two overlapping distributions. This shows the discriminating power between the two hypotheses is not sufficient. The lower right corner, Figure 9.21d, shows the Tippett plot of the likelihood ratio values. The main part of the resulting likelihood ratios range from 0.1 to 10. These values indicate that the common source score LR method does not show great discriminating power between the two hypotheses.



(a) ECE plot.

(b) Histogram of the distance scores with corresponding KDE fit.

(c) PAV plot.

(d) Tippett plot of the likelihood ratio values.

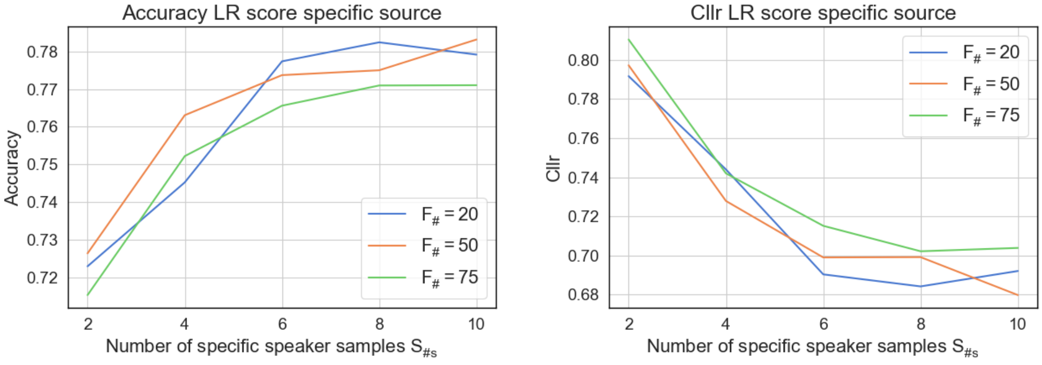
Figure 9.21: Validation plots of the distance method with the best working parameters $N = 750$, $F_{\#} = 40$.

9.3.2. Specific source LR score method

All results so far are obtained using the common source model. In addition to these results, the LR score model is also implemented applied to the specific source model. This gives rise to a different, more specific set competing hypotheses, more interesting for a judge. Since the scores are calculated using a specific speaker data set, this could lead to better or other results. The specific source model is possible for the LR score method only under strong assumptions. For the specific source model besides a background population of alternative speakers, a specific speaker data set is needed. The specific speaker data set is often limited, so an important question is whether it is possible to use the specific source model with limited data. The score LR assumes a parametric normal distribution for the specific speaker distribution. The parameters are estimated using the background data set and the samples of the specific speaker.

To first test the amount of specific speaker data that is needed to model the specific source LR method, the results for varying number of specific speaker samples $S_{\#s}$ are presented in Figure 9.22. The x-axis shows the total number of samples used to estimate $\boldsymbol{\mu}_s$. Note the small number of specific speaker samples to estimate the specific speaker distribution. The amount of samples is very small to estimate an $F_{\#}$-dimensional indexing parameter.

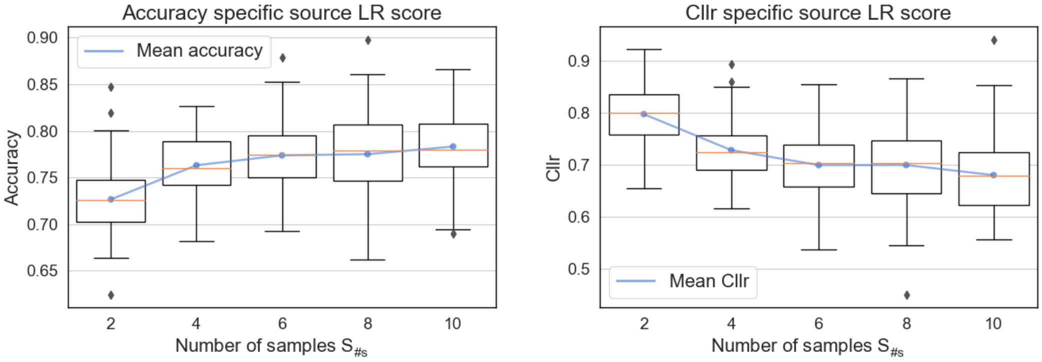
The figure shows the performance increases for a higher number of specific speaker samples $S_{\#s}$. This result is expected, as this means more samples are available to fit the specific speaker distribution. The method is only stable for a relatively low number of frequent words $F_{\#}$, as the co-variance matrix is singular for higher values. Furthermore, the available amount of data per speaker is too small to test for a larger number of specific speaker samples $S_{\#s}$. Keeping the practical implementation in mind, probably not more specific speaker samples will be present in a real forensic case. Therefore, for the next section the number of specific speaker samples is fixed at $S_{\#s} = 5$. The performance for different sample lengths N including the spread is interesting, as the estimated distribution is based on a small number of samples. What is interesting is that the performance for the relatively small sample length $N = 250$ is $A = 0.78$ and $C_{llr} = 0.68$. This is slightly better than the distance method for this sample length, the distance method had an accuracy of $A = 0.77$ and $C_{llr} = 0.74$ with $F_{\#} = 200$. Note that the competing hypotheses are different as the distance method is only applied to the common source model. A comparison of the performance of the methods using different hypotheses should be made carefully. Figure 9.23 shows the spread in the results for varying number of specific speaker samples. It shows a relative large spread in the results is present for all values of $S_{\#s}$, but no difference in the amount of spread for increasing or decreasing number of specific speaker samples.



(a) Accuracy A.

(b) C_{llr}.

Figure 9.22: Plots of mean accuracy A and C_{llr} per number of samples of the specific source $S_{\#s}$ for different values of number of frequent words $F_{\#}$ and sample length $N = 250$. The scoring method used is the specific source LR score.

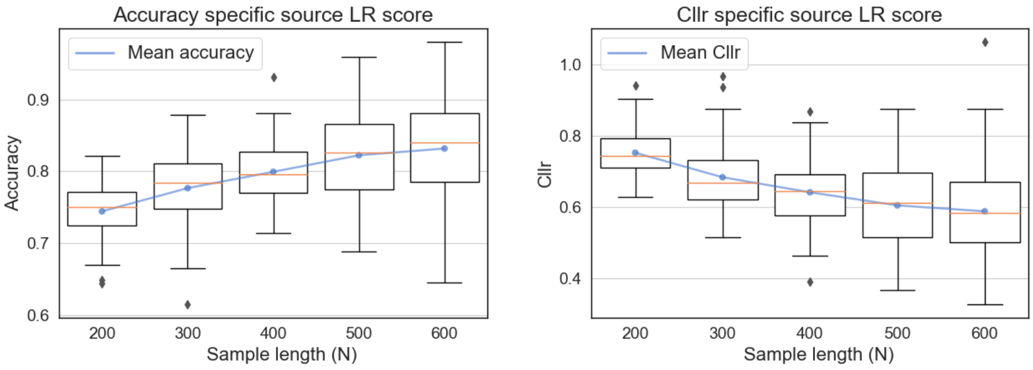


(a) Accuracy A.

(b) C_{llr}.

Figure 9.23: Boxplots of accuracy A and C_{llr} per sample length number of specific speaker samples $S_{\#s}$ for the specific source LR score method with $F_{\#} = 20$ and $N = 250$, including the mean values.

Figure 9.24 shows the performance of the specific source LR score method for varying sample lengths N. The method is only tested in the range $N \in [200, 600]$, as the data per speaker is too small to test for higher sample lengths N. The figure shows an increase in performance and in spread for larger sample lengths N. This can again be explained by the decrease of the number of samples available at higher sample lengths.

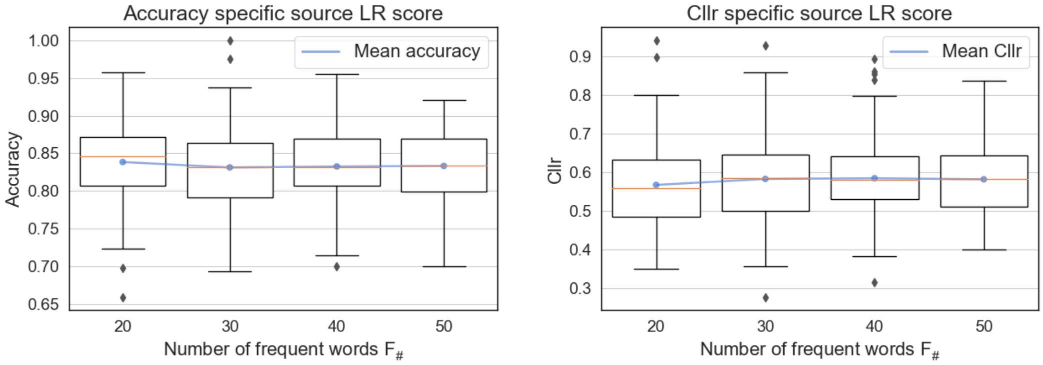


(a) Accuracy A.

(b) C_{llr}.

Figure 9.24: Boxplots of accuracy A and C_{llr} per sample length N for the specific source LR score method with $F_{\#} = 20$ and $S_{\#} = 5$, including the mean values.

To explore the effects of the number of frequent words used, Figure 9.25 shows the mean values and corresponding boxplots for varying number of frequent words $F_{\#}$. It shows no difference in performance for different numbers of frequent words. This agrees with the results found for the common source LR score method.

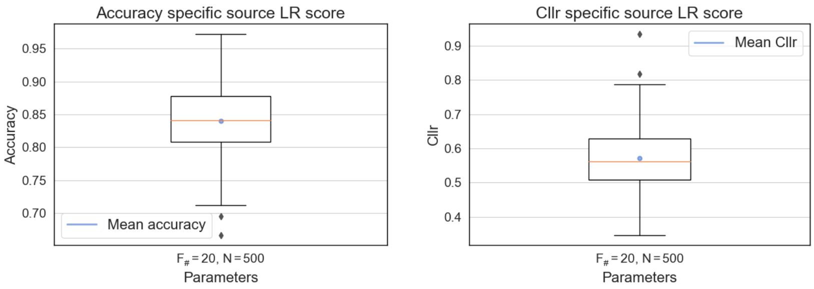


(a) Accuracy A.

(b) C_{llr}.

Figure 9.25: Plots of mean accuracy A and C_{llr} per number of frequent words with $S_{\#s} = 5$ and sample length $N = 500$ with scoring method specific source LR score.

To conclude the specific LR score section, the best obtained results are shown in Figure 9.26, with parameters $F_{\#} = 20$, $N = 500$ and $S_{\#} = 5$. This results in a mean $C_{llr} = 0.59$ and accuracy $A = 0.84$. The results show the method is not suitable for effective speaker recognition yet, further research with a larger data set is needed.

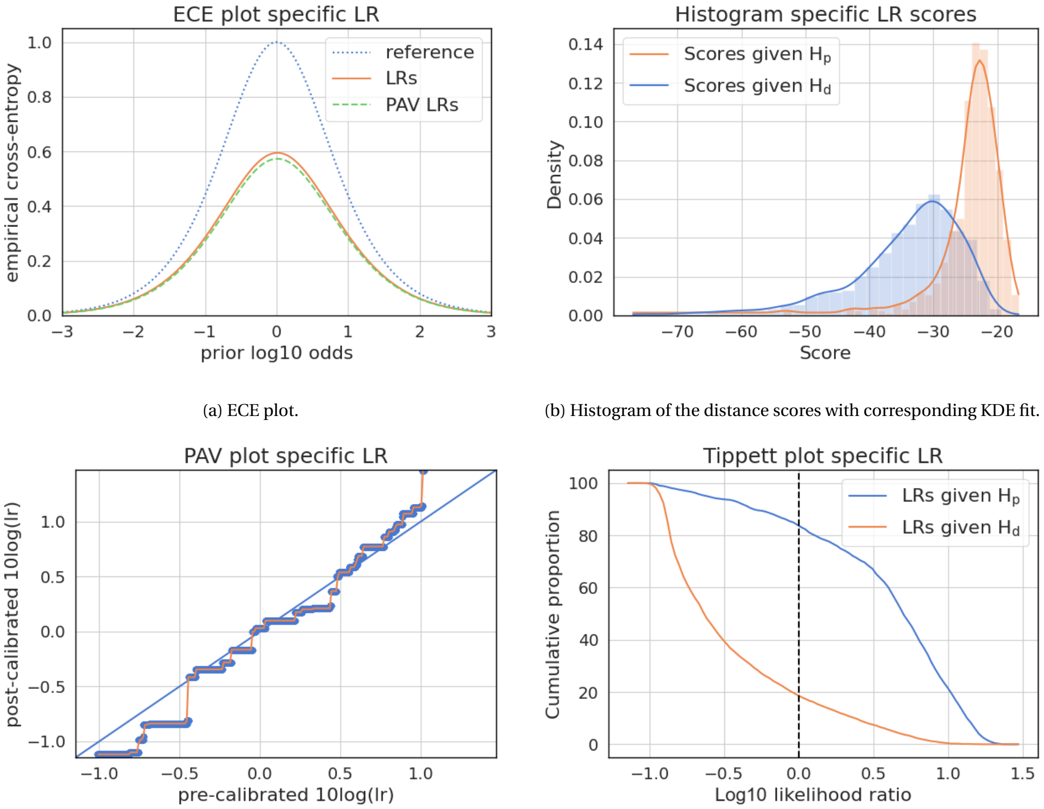


(a) Accuracy A.

(b) C_{llr}.

Figure 9.26: Boxplots of accuracy A and C_{llr}, including the mean values, for parameters $N = 500$, $F_\# = 20$ and $S_{\#s} = 5$, and scoring method specific source LR score.

Figure 9.27 shows the validation plots for the the specific source LR score method with parameters $N = 500$, $F_\# = 20$ and $S_{\#s} = 5$. The upper left corner, Figure 9.27a, shows the ECE plot. The plot shows the method is calibrated well with almost no calibration loss. The lower left corner, Figure 9.27c, shows the PAV plot. The plot shows the calibration for the lower likelihood ratios is conservative. For higher likelihood ratios the method is well-calibrated. The upper right corner, Figure 9.27b, shows the histogram of scores and the KDE fit of the scores. The fit of the two distributions is accurate. The histogram shows that the scores give rise to two overlapping distributions. The lower right corner, Figure 9.27d, shows the Tippett plot of the likelihood ratio values. The main part of the resulting likelihood ratios range from 0.1 to 10. These values show that the specific source score LR method does not show great discriminating power between the two hypotheses.



(a) ECE plot.

(b) Histogram of the distance scores with corresponding KDE fit.

(c) PAV plot.

(d) Tippett plot of the likelihood ratio values.

Figure 9.27: Validation plots of specific source LR score with the best performing parameters $N = 500$, $F_\# = 20$ and $S_{\#s} = 5$.

9.4. Comparison performance all methods

Table 9.7 shows a summary of the results of all three scoring methods: the distance method, the machine learning method and the LR score method. It shows the SVM method performs best for small, average and large sample lengths N $(250, 750, 1500)$. The parameter $S_{\#}$ is not specified, to obtain the results, for all methods the same number of same-speaker pairs is the limiting factor.

Table 9.7: Summarising table containing the best results per method with corresponding parameters.

Method	Sample length (N)	C_{llr}	Accuracy
Distance method $F_{\#} = 200$	250	0.77	0.74
	750	0.52	0.84
	1500	0.35	0.90
Machine learning method SVM $F_{\#} = 200$	250	**0.65**	**0.79**
	750	**0.36**	**0.90**
	1500	**0.24**	**0.94**
Common source LR score method $F_{\#} = 40$	250	0.78	0.73
	750	0.54	0.84
	1500	-	-
Specific source LR score method $F_{\#} = 20$, $S_{\#s} = 5$	250	0.73	0.76
	600	0.60	0.82
	1500	-	-

10 Conclusion and recommendations

In this chapter, first the conclusions and discussion are presented. Afterwards, the recommendations for future research based on the conclusions from this research are presented.

10.1. Conclusion and discussion

The objective of this research was to quantify the evidential value in the transcription of a speech fragment. The value of evidence is quantified by the likelihood ratio. The common source and specific source model for the identification of source question are outlined and discussed for application to speaker recognition. The process from input data to validated likelihood ratio is explored and constructed. Since the method based on text analysis of the transcription is independent from current automatic speaker recognition, it would contribute significantly to the overall evidential value.

The data used is the FRIDA data set consisting of transcriptions of 250 speakers. To transform the raw data to numerical feature vectors, a relevant set of features is needed. Features used in authorship analysis of written text are analysed for implementation in speaker recognition. The most promising category of features are the frequencies of occurrence of frequent words, also called function words.

The feature vectors are transformed to scores, according to the common source model. The common source model is the most suitable approach for modelling the forensic case study, given the available data. However, the common source model is more conservative than the specific source model. Three different scoring methods are proposed in this research. This is the main distinction in the likelihood ratio approaches investigated in this research. As a baseline, a method based on a distance metric is used, where the score is the distance between two feature vectors. To improve upon this baseline, several machine learning algorithms are applied to calculate a score from the feature vectors. The best results are obtained for support vector machines and XGBoost, a decision tree based algorithm. As a third method a score likelihood ratio is calculated under several assumptions based on the feature vector. It is used as score instead of a direct likelihood ratio. The likelihood ratio score method is also applied following the specific source framework, to test if better performance can be achieved. The score likelihood ratio approach is tested as it includes similarity and typicality.

The scores obtained from the scoring algorithms give rise to two probability distributions, the same-speaker and different-speaker score distribution, which are estimated with KDE. Using the estimated distribution, the test scores are transformed to a likelihood ratio. The resulting likelihood ratio values are validated by inspecting several performance characteristics. The main characteristics are discriminating power, accuracy and calibration. The most important corresponding metrics are the cost log-likelihood ratio C_{llr} and the accuracy A. Furthermore, the results are presented in the form of an ECE plot, a PAV plot and a Tippett plot. The Tippett plot shows the distribution of the resulting likelihood ratios given the hypotheses.

The resulting likelihood ratios show that information about the speaker is contained in the transcription of a speech fragment, which is a promising performance. To further test the applicability, a sensitivity analysis is performed for the sample length N, the number of frequent words $F_{\#}$ and the number of samples to train $S_{\#}$. The parameters tested in this thesis can be used as a guideline to explore the possibilities in a real forensic case.

Three main outcomes can be concluded from the results of all methods. First, the sample length N has a large effect on the performance of the likelihood ratio method. In general, higher sample lengths give better performance. It can be concluded for a common source framework, that samples with sample length $N < 750$ do not contain enough information to distinguish effectively between same-speaker and different-speaker pairs. A critical point is that a large sample length implies a smaller number of samples available. Secondly, in the range of number of frequent words $F_{\#} \leq 200$, an increase of frequent words $F_{\#}$ results in a better performance. The performance stabilises for values larger than $F_{\#} = 200$. It shows most of the information is contained in the 200 most frequent words. The performance only decreases slowly for smaller values of frequent words $F_{\#}$, although small values still show discriminating power. For frequent words obtained from the FRIDA data set, as with frequent words obtained from the CGN data set, the same effect is present. Thirdly, all results show a substantial spread in terms of accuracy and the C_{llr}, this is due to differences between distinctive speech per speaker and thus differences in the results. The results are further discussed per scoring method.

The distance method shows best performance with the Jensen–Shannon distance measure. A solid baseline is established with $A = 0.9$ and $C_{llr} = 0.35$ for $N = 1500$ and $F_{\#} = 200$. The results suggest only a small number of samples $S_{\#} = 100$ is needed to train the model accurately. However, it is important to note that if a background data set is used which is less similar to the test set, maybe more samples are needed.

The machine learning methods SVM and XGBoost both show better performance than the other methods for smaller sample lengths N and thus a large amount of samples $S_{\#}$. However, XGBoost needs a too large training set to gain a good performance for large sample lengths N. For both machine learning methods, a relatively high number of samples is needed to train the model. Around $S_{\#} = 3000$ and $S_{\#} = 1000$ samples for the XGBoost algorithm and SVM method, respectively. SVM shows the best performance of all methods, with a peak performance of $A = 0.94$ and $C_{llr} = 0.24$ for $N = 1500$ and $F_{\#} = 200$. Also for smaller samples sizes in the range between $250 \leq N \leq 1500$, SVM outperforms the baseline established by the distance method.

The score likelihood ratio methods are used to find out whether a feature-based method which included typicality could lead to better performance. The results for the specific and common source approach were similar to the baseline set by the distance method for small sample lengths $N \leq 700$. The method is numerically unstable to test for larger sample lengths. As the feature space is large and the method is unstable, the number of sample parameter $S_{\#}$ is not tested. For this method a larger data set and further research is needed.

In summary, in this thesis it has been shown that information about the speaker is contained in transcriptions of speech fragments. This is shown for the case where a limited amount of data is available from a suspect, as often appears in a real forensic case. The complete process from data to likelihood ratio is constructed and the likelihood ratio system using SVM has shown the best results. A sensitivity analysis has been performed to determine a guideline for the used parameters. This research brings us one step closer to the goal of using the method in a real forensic case. For the next steps, a set of recommendations is provided in the next section.

This research is the first step in the goal of using text analysis of transcribed speech fragments to quantify the evidential value of a speech fragment. To use the method in a real forensic case, more research is needed. A set of recommendations is outlined in the next section.

10.2. Recommendations for future work

This thesis shows support that evidential information is contained in transcriptions of spoken text.

- In this thesis the common source framework is used because of the limited available data in most forensic cases. This is the most suitable model regarding the data, but less suitable regarding the question of interest of a judge. Furthermore, no specific suspect is used as known background information. It is interesting to investigate the performance when the specific source model is implemented. The large sample length needed for a good performance in the common source model, could lead to more samples of smaller sample length. The samples can be used to model a specific source approach. Especially for the distance method, where only a small number of samples is needed for a stable performance, it is interesting to further investigate this option.

- The method is developed and tested using only speakers present in the FRIDA data set. For further validation, a study with the speakers present in the CGN data set or in real forensic case data is recommended. It is interesting to investigate whether the guidelines for the constraints on parameters resulting from this thesis, will also apply in a real forensic case. During further testing, also validation criteria need to be determined. Furthermore, the data from the FRIDA data set is immediately used as input for the method. Only a small data selection is performed, based on the amount of data available per speaker. For further research, performance can possibly be improved by more extensive data exploring.

- The feature set selected in this research consists only of frequencies of frequent words. Although this is a feature set often used, it is interesting to investigate if other features can improve the performance. New features can be extracted automatically from the transcriptions, but can also be determined in combination with speech experts. The speech expert can point out important distinguishing features used in a conversation, which are not found by using automated extracted features. It is interesting to test the combination of automatic extracted features and features from a speech expert.

- The research in this thesis was focused on the overall performance of a likelihood ratio system for speaker recognition. A large spread in the performance of the system is shown in the results section. Since the test set contains different speakers for every iteration, no conclusion can be made regarding the stability of the method for one speaker. We recommend to test the stability of the method by testing the method for a single speaker.

- It is recommended to research the assumed independency between the automatic speaker method and the text analysis method. To combine the evidential value obtained from both methods, the dependency structure has to be precisely validated. Using the assumed independency, the text analysis method contributes largely to the overall evidential value.

- The scoring methods used in this thesis are all statistical models or basic machine learning models. It is interesting to investigate whether a convolutional neural network model can achieve a better performance. Several authorship analysis studies already used a form of a convolutional neural network with frequency based features and showed promising results [49]. However, it has to be noted that the explainability of the method becomes more challenging. This is a trade-off important for forensic analysis used in court.

Bibliography

[1] Project Corpus Gesproken Nederlands. Technical report. URL `http://lands.let.ru.nl/cgn/`.

[2] Colin G. G. Aitken, Grzegorz Zadora, and David Lucy. A Two-Level Model for Evidence Evaluation. *Journal of Forensic Sciences*, 52(2):412–419, 3 2007. doi: 10.1111/j.1556-4029.2006.00358.x.

[3] Muna AlSallal, Rahat Iqbal, Vasile Palade, Saad Amin, and Victor Chang. An integrated approach for intrinsic plagiarism detection. *Future Generation Computer Systems*, 96:700–712, 2019. ISSN 0167739X. doi: 10.1016/j.future.2017.11.023.

[4] Alaa Saleh Altheneyan and Mohamed El Bachir Menai. Naïve Bayes classifiers for authorship attribution of Arabic texts. *Journal of King Saud University - Computer and Information Sciences*, 26(4):473–484, 2014. ISSN 22131248. doi: 10.1016/j.jksuci.2014.06.006.

[5] Shlomo Argamon. Measuring the Usefulness of Function Words for Authorship Attribution. In *ACH/ALLC Conference*, 2005.

[6] Annabel Bolck, Haifang Ni, and Martin Lopatka. Evaluating score- and feature-based likelihood ratio models for multivariate continuous data: applied to forensic MDMA comparison. *Law, Probability and Risk*, 14(3):243–266, 9 2015. ISSN 1470-8396. doi: 10.1093/lpr/mgv009.

[7] Niko Brummer and Johan Du Preez. The PAV algorithm optimizes binary proper scoring rules. University of Stellenbosch, 2013.

[8] Tianqi Chen. Introduction to Boosted Trees. University Lecture, University of Washington, 2014.

[9] R. Cook, I. W. Evett, G. Jackson, P. J. Jones, and J. A. Lambert. A hierarchy of propositions: Deciding which level to address in casework. *Science and Justice - Journal of the Forensic Science Society*, 38(4):231–239, 1998. ISSN 13550306. doi: 10.1016/S1355-0306(98)72117-3.

[10] Jan De Leeuw, Kurt Hornik, and Patrick Mair. Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods. *Journal of Statistical Software*, 32(5), 2009.

[11] I N Van Dorp. Statistical modelling of forensic evidence. Delft University of Technology, 2018. MSc thesis.

[12] M. Dominik Endres and E. Johannes Schindelin. A New Metric for Probability Distributions. *IEEE Transactions on Information Forensics and Security*, 49(7):1858–1860, 2003. doi: 10.1109/TIT.2003.813506.

[13] Michael O. Finkelstein and William B. Fairley. A Bayesian Approach to Identification Evidence. *Harvard Law Review*, 83(3):489, 1 1970. ISSN 0017811X. doi: 10.2307/1339656.

[14] Nathaniel Garton, Danica Ommen, Jarad Niemi, and Alicia Carriquiry. Score-based likelihood ratios to evaluate forensic pattern evidence. 2 2020.

[15] Oren Halvani, Christian Winter, and Anika Pflug. Authorship verification for different languages, genres and topics. *Digital Investigation*, 16:S33–S43, 2016. ISSN 17422876. doi: 10.1016/j.diin.2016.01.006.

[16] Rudolf Haraksim. *Validation of likelihood ratio methods used for forensic evidence evaluation: Application in forensic fingerprints*. PhD thesis, University of Twente, Enschede, The Netherlands, 6 2014.

[17] Wolfgang Härdle, Axel Werwatz, Marlene Müller, and Stefan Sperlich. *Nonparametric and Semiparametric Models*. Springer-Verlag Berlin Heidelberg, 1 edition, 2004. ISBN 978-3-540-20722-1. doi: 10.1007/978-3-642-17146-8.

[18] Simon Haykin, New York, Boston San, Francisco London, Toronto Sydney, Tokyo Singapore, Madrid Mexico, City Munich, Paris Cape, Town Hong, and Kong Montreal. *Neural Networks and Learning Machines Third Edition.* Pearson Education, 3 edition, 2009. ISBN 9780131471399.

[19] Amanda B. Hepler, Christopher P. Saunders, Linda J. Davis, and Jo Ann Buscaglia. Score-based likelihood ratios for handwriting evidence. *Forensic Science International*, 219(1-3):129–140, 2012. ISSN 03790738. doi: 10.1016/j.forsciint.2011.12.009.

[20] Fatma Howedi and Masnizah Mohd. Text Classification for Authorship Attribution Using Naive Bayes Classifier with Limited Training Data. *Computer Engineering and Intelligent Systems*, 5(4), 2014. ISSN 2222-1719.

[21] Manuela Hürlimann, Benno Weck, Esther Van Den Berg, Simon Šuster, and Malvina Nissim. GLAD: Groningen lightweight authorship detection. In *CEUR Workshop Proceedings*, volume 1391, 2015.

[22] P. P. Ippolito. SVM: Feature Selection and Kernels, 2019. URL `https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c`.

[23] Shunichi Ishihara. A likelihood ratio-based evaluation of strength of authorship attribution evidence in SMS messages using N-grams. *International Journal of Speech, Language and the Law*, 21(1):23–49, 2014. ISSN 17488893. doi: 10.1558/ijsll.v21i1.23.

[24] F S Kool. Feature-based models for forensic likelihood ratio calculation, Supporting research for the ENFSI-LR project. Delft University of Technology, 2016. MSc thesis.

[25] Fréderique Suzanne Kool, Inoni Nadine van Dorp, Annabel Bolck, Anna Jeannette Leegwater, and Geurt Jongbloed. Overall mean estimation of trace evidence in a two-level normal–normal model. *Forensic Science International*, 297:342–349, 4 2019. ISSN 18726283. doi: 10.1016/j.forsciint.2019.01.047.

[26] Anna Jeannette Leegwater, Didier Meuwly, Marjan Sjerps, Peter Vergeer, and Ivo Alberink. Performance Study of a Score-based Likelihood Ratio System for Forensic Fingermark Comparison. *Journal of Forensic Sciences*, 62(3):626–640, 5 2017. ISSN 00221198. doi: 10.1111/1556-4029.13339. URL `http://doi.wiley.com/10.1111/1556-4029.13339`.

[27] Zhenshi Li. An Exploratory Study on Authorship Verification Models for Forensic Purpose. Delft University of Technology, 2013. MSc thesis.

[28] Miranti Indar Mandasari, Mitchell McLaren, and David A. Van Leeuwen. The effect of noise on modern automatic speaker recognition systems. In *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pages 4249–4252, 2012. ISBN 9781467300469. doi: 10.1109/ICASSP.2012.6288857.

[29] Didier Meuwly, Daniel Ramos, and Rudolf Haraksim. A guideline for the validation of likelihood ratio methods used for forensic evidence evaluation. *Forensic Science International*, 276:142–153, 2017. ISSN 18726283. doi: 10.1016/j.forsciint.2016.03.048.

[30] Catarina Moreira. Learning To Rank Academic Experts. Instituto Superior Tecnico, 2011. MSc thesis.

[31] Geoffrey Stewart Morrison and Ewald Enzinger. Score based procedures for the calculation of forensic likelihood ratios – Scores should take account of both similarity and typicality. *Science and Justice*, 58 (1):47–58, 1 2018. ISSN 18764452. doi: 10.1016/j.scijus.2017.06.005.

[32] Frederick Mosteller and David L. Wallace. Inference in an Authorship Problem. *Journal of the American Statistical Association*, 58(302):275–309, 1963. ISSN 1537274X. doi: 10.1080/01621459.1963.10500849.

[33] NFI. Vakbijlage, De reeks waarschijnlijkheidstermen van het NFI en het Bayesiaanse model van interpretatie van bewijs. Technical report, 2017.

[34] Danica Ommen. *Approximate Statistical Solutions to the Forensic Identification of Source Problem.* PhD thesis, South Dakota State University, 2017.

[35] Danica Ommen and Christopher Saunders. Building a Unified Statistical Framework for the Forensic Identification of Source Problems. *Law Probability and Risk*, 17:179–197, 2018. doi: 10.1093/lpr/mgy008.

[36] Danica M Ommen and Christopher P Saunders. Reconciling the Bayes Factor and Likelihood Ratio for Two Non-Nested Model Selection Problems. 2019.

[37] Danica M. Ommen, Christopher P. Saunders, and Cedric Neumann. A Note on the Specific Source Identification Problem in Forensic Science in the Presence of Uncertainty about the Background Population. 2015.

[38] Soyoung Park and Alicia Carriquiry. Learning algorithms to evaluate forensic glass evidence. *Annals of Applied Statistics*, 13(2):1068–1102, 2019. ISSN 19417330. doi: 10.1214/18-AOAS1211.

[39] P. Jonathon Phillips and Mark Przybocki. Four Principles of Explainable AI as Applied to Biometrics and Facial Forensic Algorithms. 2 2020.

[40] John C. Platt. Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. *Advances in large margin classifiers*, pages 61–74, 1999. doi: 10.1.1.41.1639.

[41] Nektaria Potha and Efstathios Stamatatos. A profile-based method for authorship verification. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8445 LNCS, pages 313–326, 2014. ISBN 9783319070636. doi: 10.1007/978-3-319-07064-3{_}25.

[42] Daniel Ramos, Rudolf Haraksim, and Didier Meuwly. Likelihood ratio data to report the validation of a forensic fingerprint evaluation method. *Data in Brief*, 10:75–92, 2 2017. ISSN 23523409. doi: 10.1016/j.dib.2016.11.008.

[43] Daniel Ramos, Ram P. Krish, Julian Fierrez, and Didier Meuwly. From Biometric Scores to Forensic Likelihood Ratios. In Massimo Tistarelli and Christophe Champod, editors, *Handbook of Biometrics for Forensic Science*, chapter 14, pages 305–327. Springer International Publishing, 2017. ISBN 978-3-319-50671-5. doi: 10.1007/978-3-319-50673-9. URL `http://link.springer.com/10.1007/978-3-319-50673-9`.

[44] Daniel Ramos, Javier Franco-Pedroso, Alicia Lozano-Diez, and Joaquin Gonzalez-Rodriguez. Deconstructing cross-entropy for probabilistic binary classifiers. *Entropy*, 20(3):208, 2018. ISSN 10994300. doi: 10.3390/e20030208.

[45] John A. Rice. *Mathematical Statistics and Data Analysis*. Nelson Education, 2006. ISBN 978-8131519547.

[46] C. Ricotta and J. Podani. On some properties of the Bray-Curtis dissimilarity and their ecological meaning. *Ecological Complexity*, 31:201–205, 9 2017. ISSN 1476945X. doi: 10.1016/j.ecocom.2017.07.003.

[47] Leszek Rutkowski, Maciej Jaworski, Lena Pietruczuk, and Piotr Duda. The CART Decision Tree for Mining Data Streams. In *Information Science*. Czestochowa University of Technology, 2013.

[48] Jacques Savoy. Estimating the Probability of an Authorship Attribution Jacques. *Journal of the Association for Information Science and Technology*, 67(June), 2015. doi: 10.1002/asi.23455.

[49] Prasha Shrestha, Sebastian Sierra, Fabio González, Manuel Montes, Paolo Rosso, and Thamar Solorio. Convolutional Neural Networks for Authorship Attribution of Short Texts. In *European Chapter of the Association for Computational Linguistics*, pages 669–674. Association for Computational Linguistics, 2017. doi: 10.18653/v1/E17-2106.

[50] Efstathios Stamatatos. A survey of modern authorship attribution methods. *Journal of the American Society for Information Science and Technology*, 60(3):538–556, 2009. ISSN 15322882. doi: 10.1002/asi.21001.

[51] Efstathios Stamatatos. Authorship Attribution Using Text Distortion. In *the Association for Computational Linguistics*, volume 1, pages 1138–1149, 2017.

[52] Yi Tang and Sargur N. Srihari. Likelihood ratio estimation in forensic identification using similarity and rarity. *Pattern Recognition*, 47(3):945–958, 3 2014. ISSN 00313203. doi: 10.1016/j.patcog.2013.07.014.

[53] C. F. Tippett, V. J. Emerson, M. J. Fereday, F. Lawton, A. Richardson, L. T. Jones, and Miss S.M. Lampert. The Evidential Value of the Comparison of Paint Flakes from Sources other than Vehicles. *Journal of the Forensic Science Society*, 8(2-3):61–65, 1968. ISSN 00157368. doi: 10.1016/S0015-7368(68)70442-4.

[54] David Van Der Vloed, Jos Bouten, Finnian Kelly, and Anil Alexander. NFI-FRIDA-Forensically Realistic Inter-Device Audio database and initial experiments. In *ISCA Speaker and Language Characterization*, 2014. URL http://www.praat.org/.

[55] Peter Vergeer, Andrew van Es, Arent de Jongh, Ivo Alberink, and Reinoud Stoel. Numerical likelihood ratios outputted by LR systems are often based on extrapolation: When to stop extrapolating? *Science and Justice*, 56(6):482–491, 12 2016. ISSN 18764452. doi: 10.1016/j.scijus.2016.06.003.

[56] David Van Der Vloed, Finnian Kelly, and Anil Alexander. Exploring The Effects Of Device Variability On Forensic Speaker Comparison Using VOCALISE And NFI-FRIDA, A Forensically Realistic Database. In *ODYSSEY*, 2020.

[57] Wikipedia. Kernel density estimation, 2020. URL https://en.wikipedia.org/wiki/Kernel_density_estimation.

[58] Wikipedia. Isotonic regression, 2020. URL https://en.wikipedia.org/wiki/Isotonic_regression.

[59] Meike Willebrands and Menno Groenewegen. Over DNA in auto's, vingersporenonderzoek en automatische sprekervergelijking Jaargang 6. *@NFI*, 2019. URL magazines.forensischinstituut.nl/.

[60] SM Willis, L McKenna, S McDermott, G O'Donell, A Barrett, B Rasmusson, A Nordgaard, CEH Berger, MJ Sjerps, and J Lucena-Molina. ENFSI guideline for evaluative reporting in forensic science. Technical report, European Network of Forensis Science Institutes, 2015.

[61] Shuo Xu, Yan Li, and Zheng Wang. Bayesian multinomial naïve bayes classifier to text classification. In *Lecture Notes in Electrical Engineering*, volume 448, pages 347–352. Springer Verlag, 2017. ISBN 9789811050404. doi: 10.1007/978-981-10-5041-1_57.

[62] Ying Zhao and Justin Zobel. Effective and Scalable Authorship Attribution Using Function Words. In *AIRS 2005*, pages 174–189, 2005. URL papers2://publication/uuid/97C99154-6116-4BBD-85CC-3BE690928DF5.

[63] Jacob Coenraad De Zoete. *Combining Forensic Evidence*. PhD thesis, University of Amsterdam, 2016.

A

Data

A.1. Frequent words FRIDA data set

The 200 most frequent words obtained from the FRIDA data set. First the word is stated, then the absolute number of occurrences and the relative occurrence in percentage.

1. ja: 49919x - 5.19%
2. ik: 32697x - 3.40%
3. uh: 32273x - 3.36%
4. je: 29893x - 3.11%
5. a: 26474x - 2.75%
6. is: 17026x - 1.77%
7. dat: 16568x - 1.72%
8. maar: 14816x - 1.54%
9. die: 14059x - 1.46%
10. en: 14047x - 1.46%
11. niet: 13142x - 1.37%
12. s: 10986x - 1.14%
13. dan: 10873x - 1.13%
14. een: 10849x - 1.13%
15. wel: 9905x - 1.03%
16. man: 9842x - 1.02%
17. nee: 9432x - 0.98%
18. het: 9398x - 0.98%
19. ook: 8553x - 0.89%
20. gewoon: 8487x - 0.88%
21. de: 8399x - 0.87%
22. wat: 7862x - 0.82%
23. v: 7739x - 0.80%
24. t: 7554x - 0.79%
25. weet: 7470x - 0.78%
26. ze: 7148x - 0.74%
27. van: 7139x - 0.74%
28. heb: 6754x - 0.70%
29. hij: 6164x - 0.64%
30. in: 6104x - 0.63%
31. met: 5986x - 0.62%
32. we: 5903x - 0.61%
33. oo: 5604x - 0.58%
34. dus: 5341x - 0.56%
35. echt: 5183x - 0.54%
36. nog: 5048x - 0.52%
37. toch: 4987x - 0.52%
38. okee: 4940x - 0.51%
39. op: 4750x - 0.49%
40. was: 4736x - 0.49%
41. als: 4545x - 0.47%
42. moet: 4514x - 0.47%
43. ga: 4457x - 0.46%
44. of: 4333x - 0.45%
45. he: 4332x - 0.45%
46. me: 3966x - 0.41%
47. zo: 3948x - 0.41%
48. kan: 3806x - 0.40%
49. jij: 3780x - 0.39%
50. er: 3758x - 0.39%
51. daar: 3706x - 0.39%
52. goed: 3665x - 0.38%
53. voor: 3650x - 0.38%
54. naar: 3562x - 0.37%
55. gaat: 3363x - 0.35%
56. ben: 3347x - 0.35%
57. hoe: 3309x - 0.34%
58. gaan: 3297x - 0.34%
59. x: 3265x - 0.34%
60. zijn: 3191x - 0.33%
61. nu: 3191x - 0.33%
62. te: 3017x - 0.31%
63. nou: 2986x - 0.31%
64. ee: 2957x - 0.31%
65. had: 2802x - 0.29%
66. mij: 2784x - 0.29%
67. ie: 2769x - 0.29%
68. bij: 2731x - 0.28%
69. zeg: 2473x - 0.26%
70. u: 2471x - 0.26%
71. want: 2453x - 0.26%
72. aan: 2417x - 0.25%
73. m: 2411x - 0.25%
74. doen: 2402x - 0.25%
75. toen: 2249x - 0.23%
76. effe: 2231x - 0.23%
77. al: 2207x - 0.23%
78. weer: 2187x - 0.23%
79. heel: 2169x - 0.23%
80. aa: 2081x - 0.22%
81. ofzo: 2050x - 0.21%
82. heeft: 1950x - 0.20%
83. geen: 1909x - 0.20%
84. over: 1895x - 0.20%
85. precies: 1886x - 0.20%
86. denk: 1883x - 0.20%
87. k: 1860x - 0.19%
88. twee: 1832x - 0.19%
89. hebben: 1823x - 0.19%
90. zei: 1817x - 0.19%
91. hoor: 1808x - 0.19%
92. waar: 1767x - 0.18%
93. beetje: 1762x - 0.18%
94. om: 1680x - 0.17%
95. veel: 1653x - 0.17%
96. eigenlijk: 1620x - 0.17%
97. klopt: 1614x - 0.17%
98. meer: 1614x - 0.17%
99. één: 1606x - 0.17%
100. uhm: 1604x - 0.17%
101. mm: 1570x - 0.16%
102. keer: 1551x - 0.16%
103. helemaal: 1545x - 0.16%
104. wil: 1540x - 0.16%
105. d: 1465x - 0.15%
106. i: 1403x - 0.15%
107. zou: 1375x - 0.14%
108. w: 1343x - 0.14%
109. alleen: 1340x - 0.14%
110. lekker: 1328x - 0.14%
111. dit: 1308x - 0.14%
112. jou: 1291x - 0.13%
113. volgens: 1273x - 0.13%
114. vvv: 1264x - 0.13%
115. tegen: 1214x - 0.13%
116. j: 1198x - 0.12%
117. kijk: 1193x - 0.12%
118. wie: 1190x - 0.12%
119. mensen: 1184x - 0.12%
120. zit: 1173x - 0.12%
121. niks: 1156x - 0.12%
122. uit: 1146x - 0.12%
123. hier: 1125x - 0.12%

124. kijken: 1123x - 0.12%
125. zeggen: 1122x - 0.12%
126. hee: 1103x - 0.11%
127. komt: 1103x - 0.11%
128. misschien: 1089x - 0.11%
129. allemaal: 1081x - 0.11%
130. das: 1072x - 0.11%
131. zon: 1071x - 0.11%
132. wollah: 1064x - 0.11%
133. hele: 1036x - 0.11%
134. net: 1030x - 0.11%
135. waarom: 1016x - 0.11%
136. ging: 1008x - 0.10%
137. vind: 1005x - 0.10%
138. bent: 1004x - 0.10%
139. enzo: 1001x - 0.10%
140. hem: 1000x - 0.10%
141. iets: 997x - 0.10%
142. hebt: 989x - 0.10%
143. n: 985x - 0.10%
144. drie: 983x - 0.10%
145. mee: 973x - 0.10%
146. kunnen: 969x - 0.10%
147. sowieso: 967x - 0.10%
148. andere: 956x - 0.10%
149. alles: 955x - 0.10%
150. e: 947x - 0.10%
151. vijf: 947x - 0.10%
152. zegt: 942x - 0.10%
153. doe: 907x - 0.09%
154. zeker: 891x - 0.09%
155. altijd: 869x - 0.09%
156. beter: 857x - 0.09%
157. wordt: 849x - 0.09%
158. moeten: 848x - 0.09%
159. kom: 826x - 0.09%
160. zie: 809x - 0.08%
161. ding: 807x - 0.08%
162. jongen: 805x - 0.08%
163. door: 796x - 0.08%
164. mag: 796x - 0.08%
165. leuk: 788x - 0.08%
166. inderdaad: 787x - 0.08%
167. dingen: 778x - 0.08%
168. jaar: 774x - 0.08%
169. uur: 771x - 0.08%
170. vier: 759x - 0.08%
171. anders: 747x - 0.08%
172. best: 724x - 0.08%
173. tijd: 718x - 0.07%
174. g: 709x - 0.07%
175. komen: 705x - 0.07%
176. z: 705x - 0.07%
177. laat: 694x - 0.07%
178. lang: 687x - 0.07%
179. heet: 684x - 0.07%
180. daarom: 683x - 0.07%
181. bedoel: 670x - 0.07%
182. weg: 664x - 0.07%
183. dacht: 662x - 0.07%
184. dr: 653x - 0.07%
185. dag: 648x - 0.07%
186. wa: 648x - 0.07%
187. praten: 647x - 0.07%
188. ken: 646x - 0.07%
189. snap: 646x - 0.07%
190. waren: 636x - 0.07%
191. deze: 629x - 0.07%
192. goeie: 621x - 0.06%
193. omdat: 616x - 0.06%
194. mooi: 612x - 0.06%
195. doet: 611x - 0.06%
196. dats: 607x - 0.06%
197. mijn: 588x - 0.06%
198. natuurlijk: 587x - 0.06%
199. da: 585x - 0.06%
200. serieus: 582x - 0.06%

A.2. Frequent words CGN data set

The 200 most frequent words obtained from the CGN data set. First the word is stated, then the absolute number of occurrences in the FRIDA data and the relative occurrence in percentage.

1. ja: 49916 x - 5.19%
2. uh: 32272 x - 3.36%
3. ik: 32697 x - 3.40%
4. dat: 16568 x - 1.72%
5. en: 14047 x - 1.46%
6. t: 7554 x - 0.79%
7. je: 29889 x - 3.11%
8. maar: 14815 x - 1.54%
9. een: 10849 x - 1.13%
10. dan: 10873 x - 1.13%
11. is: 17026 x - 1.77%
12. die: 14058 x - 1.46%
13. de: 8399 x - 0.87%
14. niet: 13141 x - 1.37%
15. nou: 2986 x - 0.31%
16. wel: 9905 x - 1.03%
17. ook: 8553 x - 0.89%
18. a: 26473 x - 2.75%
19. nee: 9432 x - 0.98%
20. oh: 12915 x - 0.00%
21. dus: 5341 x - 0.56%
22. van: 7139 x - 0.74%
23. in: 6104 x - 0.63%
24. zo: 3948 x - 0.41%
25. ze: 7148 x - 0.74%
26. nog: 5048 x - 0.52%
27. heb: 6753 x - 0.70%
28. of: 4333 x - 0.45%
29. was: 4736 x - 0.49%
30. op: 4750 x - 0.49%
31. k: 1860 x - 0.19%
32. wat: 7861 x - 0.82%
33. we: 5903 x - 0.61%
34. met: 5986 x - 0.62%
35. want: 2453 x - 0.26%
36. daar: 3706 x - 0.39%
37. hè: 6127 x - 0.00%
38. als: 4545 x - 0.47%
39. gewoon: 8486 x - 0.88%
40. moet: 4514 x - 0.47%
41. te: 3017 x - 0.31%
42. weet: 7469 x - 0.78%
43. voor: 3650 x - 0.38%
44. goed: 3665 x - 0.38%
45. toch: 4986 x - 0.52%
46. had: 2802 x - 0.29%
47. heel: 2169 x - 0.23%
48. weer: 2187 x - 0.23%
49. dr: 653 x - 0.07%
50. zijn: 3191 x - 0.33%
51. echt: 5183 x - 0.54%
52. al: 2207 x - 0.23%
53. toen: 2248 x - 0.23%
54. ie: 2769 x - 0.29%
55. hebben: 1823 x - 0.19%
56. bij: 2731 x - 0.28%
57. kan: 3806 x - 0.40%
58. aan: 2417 x - 0.25%
59. naar: 3562 x - 0.37%
60. hij: 6163 x - 0.64%
61. om: 1680 x - 0.17%
62. ben: 3347 x - 0.35%
63. das: 1072 x - 0.11%
64. er: 3758 x - 0.39%
65. uhm: 1604 x - 0.17%
66. heeft: 1950 x - 0.20%
67. zeg: 2473 x - 0.26%
68. leuk: 788 x - 0.08%
69. hoe: 3309 x - 0.34%
70. hoor: 1808 x - 0.19%
71. denk: 1883 x - 0.20%
72. beetje: 1762 x - 0.18%
73. even: 393 x - 0.04%
74. gaan: 3297 x - 0.34%
75. hu: 168 x - 0.02%
76. doen: 2402 x - 0.25%
77. v: 7739 x - 0.80%
78. natuurlijk: 587 x - 0.06%
79. mm: 1570 x - 0.16%
80. meer: 1614 x - 0.17%
81. over: 1895 x - 0.20%
82. helemaal: 1545 x - 0.16%
83. allemaal: 1081 x - 0.11%
84. gaat: 3363 x - 0.35%
85. jij: 3780 x - 0.39%
86. eigenlijk: 1620 x - 0.17%
87. mmm: 2358 x - 0.00%
88. keer: 1551 x - 0.16%
89. mij: 2784 x - 0.29%
90. nu: 3191 x - 0.33%
91. veel: 1653 x - 0.17%
92. m: 2411 x - 0.25%
93. zon: 1071 x - 0.11%

94. x: 3265 x - 0.34%
95. ga: 4457 x - 0.46%
96. vind: 1005 x - 0.10%
97. geen: 1909 x - 0.20%
98. waar: 1767 x - 0.18%
99. zit: 1173 x - 0.12%
100. zei: 1816 x - 0.19%
101. uit: 1146 x - 0.12%
102. oké: 1917 x - 0.00%
103. één: 1898 x - 0.00%
104. zou: 1375 x - 0.14%
105. het: 9398 x - 0.98%
106. mee: 973 x - 0.10%
107. iets: 997 x - 0.10%
108. twee: 1832 x - 0.19%
109. me: 3965 x - 0.41%
110. u: 2471 x - 0.26%
111. mn: 390 x - 0.04%
112. ns: 240 x - 0.02%
113. wij: 454 x - 0.05%
114. wil: 1540 x - 0.16%
115. z: 705 x - 0.07%
116. hele: 1036 x - 0.11%
117. niks: 1156 x - 0.12%
118. kunnen: 969 x - 0.10%
119. precies: 1886 x - 0.20%
120. hier: 1125 x - 0.12%
121. misschien: 1089 x - 0.11%
122. altijd: 869 x - 0.09%
123. geweest: 332 x - 0.03%
124. zeggen: 1122 x - 0.12%
125. net: 1030 x - 0.11%
126. komt: 1103 x - 0.11%
127. hebt: 989 x - 0.10%
128. d: 1465 x - 0.15%
129. erg: 415 x - 0.04%
130. hadden: 511 x - 0.05%
131. dit: 1308 x - 0.14%
132. uur: 771 x - 0.08%
133. effe: 2231 x - 0.23%
134. alleen: 1340 x - 0.14%
135. zal: 295 x - 0.03%
136. kijken: 1123 x - 0.12%
137. wordt: 849 x - 0.09%
138. lekker: 1328 x - 0.14%
139. mensen: 1184 x - 0.12%
140. andere: 956 x - 0.10%
141. moeten: 848 x - 0.09%
142. anders: 747 x - 0.08%
143. week: 450 x - 0.05%
144. zn: 482 x - 0.05%
145. inderdaad: 787 x - 0.08%
146. mooi: 612 x - 0.06%
147. dacht: 661 x - 0.07%
148. waren: 636 x - 0.07%
149. zitten: 543 x - 0.06%
150. bedoel: 670 x - 0.07%
151. zegt: 942 x - 0.10%
152. door: 796 x - 0.08%
153. komen: 705 x - 0.07%
154. drie: 983 x - 0.10%
155. ging: 1008 x - 0.10%
156. jullie: 541 x - 0.06%
157. jaar: 774 x - 0.08%
158. jou: 1291 x - 0.13%
159. vond: 302 x - 0.03%
160. gedaan: 491 x - 0.05%
161. dingen: 778 x - 0.08%
162. tijd: 718 x - 0.07%
163. hé: 963 x - 0.00%
164. dag: 648 x - 0.07%
165. doe: 907 x - 0.09%
166. volgens: 1273 x - 0.13%
167. zoiets: 316 x - 0.03%
168. moest: 487 x - 0.05%
169. weg: 664 x - 0.07%
170. zelf: 528 x - 0.05%
171. omdat: 616 x - 0.06%
172. s: 10986 x - 1.14%
173. tegen: 1213 x - 0.13%
174. mijn: 588 x - 0.06%
175. alles: 955 x - 0.10%
176. gehad: 213 x - 0.02%
177. zich: 295 x - 0.03%
178. verder: 381 x - 0.04%
179. ah: 15 x - 0.00%
180. kijk: 1193 x - 0.12%
181. af: 458 x - 0.05%
182. zij: 556 x - 0.06%
183. best: 724 x - 0.08%
184. tot: 466 x - 0.05%
185. nooit: 544 x - 0.06%
186. zeker: 891 x - 0.09%
187. ons: 427 x - 0.04%
188. eerst: 432 x - 0.04%
189. staat: 407 x - 0.04%
190. morgen: 370 x - 0.04%
191. zat: 437 x - 0.05%
192. geloof: 204 x - 0.02%
193. laat: 694 x - 0.07%
194. zien: 552 x - 0.06%
195. geval: 224 x - 0.02%
196. worden: 466 x - 0.05%
197. doet: 611 x - 0.06%
198. kom: 826 x - 0.09%
199. kun: 140 x - 0.01%
200. tien: 484 x - 0.05%

Number of frequent words

It is interesting to explore what percentage of the original data is used when a number of frequent words $F_{\#}$ is used. The percentage is calculated by dividing the number of words when all non-frequent words are deleted, divided by the total number of words in the FRIDA data set. Table A.1a shows the number of frequent words used, obtained from the FRIDA data set, the percentage used from the original text and the absolute count of frequent words in the FRIDA data set. The FRIDA data set contains 961,671 individual words. Table A.1b shows the same structure, here the frequent words are obtained from the CGN data set.

Table A.1: The number of frequent words, the frequent words divided by the total words and the absolute count of frequent words.

(a) FRIDA data set.

F	Percentage [%]	Absolute count
10	25.7	247761
25	40.3	388248
50	53.5	515470
100	66.1	636443
200	75.9	730483
300	80.5	774440
400	83.3	801416
500	85.3	819896
600	86.6	833550

(b) CGN data set.

F	Percentage [%]	Absolute count
10	22.8	219480
25	37.3	359133
50	48.2	463856
100	60.4	581306
200	70.7	680230
300	76.2	732942
400	79.2	762028
500	81.3	782165
600	82.5	793207

The tables show the surprising fact that around 25% of all words in the FRIDA data is used, when only 10 frequent words obtained from the same data or the CGN data are used.

B

Detailed calculations and settings

B.1. Two-level normal-normal model

Following the two-level normal-normal model, the following likelihood function are used. The derivation is applied for the speaker recognition case study in this thesis, a more extensive derivation can be found in [11].

B.1.1. Common source

The common source likelihood function given H_p, the numerator of the likelihood ratio, is given as

$$\begin{aligned}
&\int f_a\left(\boldsymbol{y}_{u_1}|\mathbf{p},\boldsymbol{\theta}_a\right) f_a\left(\boldsymbol{y}_{u_2}|\mathbf{p},\boldsymbol{\theta}_a\right) g\left(\mathbf{p}|\boldsymbol{\theta}_a\right) d\mathbf{p}\\
=&\int \prod_{j=1}^{2} f_a\left(\boldsymbol{y}_{uj}|\mathbf{p},\boldsymbol{\theta}_a\right) g\left(\mathbf{p}|\boldsymbol{\theta}_a\right) dp\\
=&(2\pi)^{-l}|\boldsymbol{\Sigma}_{\boldsymbol{w}}|^{-1}|\boldsymbol{\Sigma}_b|^{-1/2}\left|\boldsymbol{\Sigma}_b^{-1}+2\boldsymbol{\Sigma}_w^{-1}\right|^{-1/2}\exp\left[-\frac{1}{2}\sum_{j=1}^{2}\left(\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_u\right)-\frac{1}{2}\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\
&\cdot\exp\left[\frac{1}{2}\left(\sum_{j=1}^{2}\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\Sigma_b^{-1}\right)\left(\Sigma_b^{-1}+2\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\sum_{j=1}^{2}\mathbf{y}_u+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right],
\end{aligned}\tag{B.1}$$

where ($\boldsymbol{y}_u = (\boldsymbol{y}_{u_1}, \boldsymbol{y}_{u_2})$).

The common source likelihood function given H_d, the denominator of the likelihood ratio, is given as

$$\begin{aligned}
&\int f_a\left(\boldsymbol{y}_{u_1}|\mathbf{p}_1,\boldsymbol{\theta}_a\right) g\left(\mathbf{p}_1|\boldsymbol{\theta}_a\right) d\mathbf{p}_1\cdot\int f_a\left(\boldsymbol{y}_{u_2}|\mathbf{p}_2,\boldsymbol{\theta}_a\right) g\left(\mathbf{p}_2|\boldsymbol{\theta}_a\right) d\mathbf{p}_2\\
&=(2\pi)^{-l/2}|\boldsymbol{\Sigma}_w|^{-1/2}|\boldsymbol{\Sigma}_b|^{-1/2}\left|\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right|^{-1/2}\exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_1}^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_1}\right)-\frac{1}{2}\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\
&\cdot\exp\left[\frac{1}{2}\left(\mathbf{y}_{u_1}^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\Sigma_b^{-1}\right)\left(\Sigma_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_1}+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]\\
&\times(2\pi)^{-l/2}|\boldsymbol{\Sigma}_w|^{-1/2}|\boldsymbol{\Sigma}_b|^{-1/2}\left|\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right|^{-1/2}\exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_2}^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_2}\right)-\frac{1}{2}\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\
&\cdot\exp\left[\frac{1}{2}\left(\mathbf{y}_{u_2}^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\Sigma_b^{-1}\right)\left(\Sigma_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_2}+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]\\
&=(2\pi)^{-l}|\boldsymbol{\Sigma}_w|^{-1}|\boldsymbol{\Sigma}_b|^{-1}\left|\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right|^{-1}\exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_1}^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_1}\right)-\frac{1}{2}\left(\mathbf{y}_{u_2}^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_2}\right)-\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\
&\cdot\exp\left[\frac{1}{2}\left(\mathbf{y}_{u_1}^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\right)\left(\Sigma_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_1}+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]\\
&\cdot\exp\left[\frac{1}{2}\left(\mathbf{y}_{u_2}^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\Sigma_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_2}+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right].
\end{aligned}\tag{B.2}$$

It then follows the common source likelihood ratio is given as

$$\begin{aligned} LR_{CS} =&|\boldsymbol{\Sigma}_b|^{1/2}\left|\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right|\left|\boldsymbol{\Sigma}_b^{-1}+2\boldsymbol{\Sigma}_w^{-1}\right|^{-1/2}\exp\left[\frac{1}{2}\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\ &\cdot\exp\left[\frac{1}{2}\left(\sum_{j=1}^{2}\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1}+2\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\sum_{j=1}^{2}\mathbf{y}_u+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]\\ &\cdot\exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_1}^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_1}+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]\\ &\cdot\exp\left[-\frac{1}{2}\left(\mathbf{y}_{u_2}^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\Sigma_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_{u_2}+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]. \end{aligned} \tag{B.3}$$

B.1.2. Specific source

The specific source likelihood function given H_p, the numerator of the likelihood ratio, is given as

$$f_s(\boldsymbol{y_u}|\boldsymbol{\theta}_s) = (2\pi)^{-l/2}|\boldsymbol{\Sigma}_s|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{y}_u-\boldsymbol{\mu}_s)^T\boldsymbol{\Sigma}_{\boldsymbol{s}}^{\mathbf{-1}}(\mathbf{y}_u-\boldsymbol{\mu}_s)\right]. \tag{B.4}$$

The specific source likelihood function given H_d, the denominator of the likelihood ratio, is given as

$$\begin{aligned} &\int f_a\left(\boldsymbol{y_u}|\mathbf{p},\boldsymbol{\theta}_a\right)g\left(\mathbf{p}|\boldsymbol{\theta}_a\right)d\mathbf{p}\\ =&\int(2\pi)^{-l/2}|\boldsymbol{\Sigma}_{\boldsymbol{w}}|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{y}_u-\mathbf{p})^T\boldsymbol{\Sigma}_{\boldsymbol{w}}^{\mathbf{-1}}(\mathbf{y}_u-\mathbf{p})\right](2\pi)^{-l/2}|\boldsymbol{\Sigma}_b|^{-1/2}\exp\left[-\frac{1}{2}(\mathbf{p}-\mu_{\mathbf{b}})^T\boldsymbol{\Sigma}_{\boldsymbol{b}}^{\mathbf{-1}}(\mathbf{p}-\mu_{\mathbf{b}})\right]d\mathbf{p}\\ =&(2\pi)^{-l/2}|\boldsymbol{\Sigma}_w|^{-1/2}|\boldsymbol{\Sigma}_b|^{-1/2}\left|\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right|^{-1/2}\exp\left[-\frac{1}{2}\left(\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_u\right)-\frac{1}{2}\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\ &\cdot\exp\left[\frac{1}{2}\left(\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_u+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]. \end{aligned} \tag{B.5}$$

It then follows the specific source likelihood ratio is given as

$$\begin{aligned} LR_{SS} =&|\boldsymbol{\Sigma}_{\boldsymbol{s}}|^{-1/2}|\boldsymbol{\Sigma}_w|^{1/2}|\boldsymbol{\Sigma}_b|^{1/2}\left|\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right|^{1/2}\\ &\cdot\exp\left[-\frac{1}{2}(\mathbf{y}_u-\boldsymbol{\mu}_s)^T\boldsymbol{\Sigma}_{\boldsymbol{s}}^{\mathbf{-1}}(\mathbf{y}_u-\boldsymbol{\mu}_s)\right]\\ &\cdot\exp\left[\frac{1}{2}\left(\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_u\right)+\frac{1}{2}\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right]\\ &\cdot\exp\left[-\frac{1}{2}\left(\mathbf{y}_u^T\boldsymbol{\Sigma}_w^{-1}+\boldsymbol{\mu}_b^T\boldsymbol{\Sigma}_b^{-1}\right)\left(\boldsymbol{\Sigma}_b^{-1}+\boldsymbol{\Sigma}_w^{-1}\right)^{-1}\left(\boldsymbol{\Sigma}_w^{-1}\mathbf{y}_u+\boldsymbol{\Sigma}_b^{-1}\boldsymbol{\mu}_b\right)\right]. \end{aligned} \tag{B.6}$$

B.2. Machine learning algorithm settings

The used machine learning algorithms are readily implemented in Python, with hyper parameters that can be varied for optimal performance per case. For exactness and reproducibility the parameters which are tuned are outlined below.

Naive Bayes
For all calculations the *sklearn.naive_bayes.MultinomialNB* Python class is used. The default parameters are used, outlined in the *sklearn* documentation.

Support vector machines
For all calculations the *sklearn.svm.SVC* Python class is used. The default parameters are used, outlined in the *sklearn* documentation. The following hyper parameters are tuned:

- kernel = linear
- probability = True
- class_weight = 'balanced'

XGBoost
For all calculations the *xgboost.XGBclassifier* Python class is used. The default parameters are used, outlined in the *xgboost* documentation. The following hyper parameters are tuned:

- learning_rate = 0.2
- max_depth = 3

C

Additional results

C.1. Distance method

C.1.1. Different distance metrics

Table C.1 shows that the Shannon-Jenssen distance has the best performance in comparison with the other distance metrics, for varying parameters.

Table C.1: Results in terms of mean C_{llr} and accuracy A for the distance method for varying distance metrics, with $N = 1000$, $F_{\#} = 200$ and $S_{\#} = 2000$.

Distance metric δ	C_{llr}	Accuracy
Jenssen-Shannon	**0.45**	**0.87**
Bray-Curtis	0.46	0.86
Manhattan	0.51	0.84
Euclidean	0.58	0.82
Cosine	0.60	0.82

C.1.2. Tables additional Cllr results

Table C.2 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying sample length N, with $F_{\#} = 200$ and $S_{\#} = 5000$.

Table C.2: Results in terms of mean C_{llr} and accuracy A for the distance method for varying number of sample length N, with $F_{\#} = 200$ and $S_{\#} = 5000$.

Sample length (N)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
250	0.77	0.72	0.05	0.74
500	0.59	0.53	0.06	0.82
750	0.52	0.43	0.09	0.84
1000	0.42	0.30	0.11	0.88
1250	0.40	0.26	0.15	0.88
1500	0.35	0.19	0.16	0.90

Table C.4 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of frequent words $F_{\#}$, with $N = 750$ and $S_{\#} = 5000$.

Table C.3: Results in terms of mean C_{llr} and accuracy A for the distance method for varying number of frequent words $F_{\#}$, with $N = 750$ and $S_{\#} = 5000$

Number of frequent words ($F_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
25	0.76	0.68	0.08	0.74
50	0.71	0.62	0.08	0.77
100	0.68	0.61	0.08	0.78
150	0.62	0.54	0.09	0.80
200	0.59	0.51	0.08	0.82
250	0.58	0.50	0.09	0.84
300	0.53	0.44	0.08	0.84
400	0.52	0.44	0.08	0.84
600	0.52	0.44	0.09	0.85

Table C.3 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of samples used for training $S_{\#}$, with $N = 500$ and $F_{\#} = 200$..

Table C.4: Results in terms of mean C_{llr} and accuracy A for the distance method for varying number of samples to train $S_{\#}$, with $N = 500$ and $F_{\#} = 200$.

Number of samples ($S_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
10	0.70	0.57	0.12	0.80
50	0.65	0.57	0.08	0.80
75	0.62	0.55	0.07	0.81
100	0.63	0.56	0.07	0.81
200	0.62	0.55	0.06	0.81
300	0.62	0.56	0.06	0.81
500	0.61	0.55	0.06	0.81
1000	0.61	0.56	0.05	0.81
2000	0.61	0.55	0.05	0.81
3000	0.60	0.54	0.05	0.81
4000	0.60	0.54	0.06	0.81

C.2. Machine learning results

SVM
Table C.5 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying sample length N, with $F_{\#} = 200$.

Table C.5: Results in terms of mean C_{llr} and accuracy A for the SVM method for varying sample lengths N, with $F_{\#} = 200$

Sample length (N)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
250	0.65	0.62	0.03	0.79
500	0.46	0.41	0.05	0.87
750	0.36	0.29	0.08	0.90
1000	0.31	0.20	0.10	0.92
1250	0.29	0.15	0.14	0.93
1500	0.24	0.11	0.14	0.94

Table C.6 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of frequent words $F_{\#}$, with $N = 750$.

Table C.6: Results in terms of mean C_{llr} and accuracy A for the SVM method for varying number of frequent words $F_{\#}$, with $N = 750$.

Number of frequent words ($F_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
50	0.49	0.41	0.08	0.86
100	0.39	0.31	0.08	0.89
150	0.40	0.31	0.09	0.89
200	0.36	0.28	0.08	0.91
250	0.36	0.28	0.08	0.90
300	0.37	0.28	0.09	0.90
400	0.35	0.27	0.08	0.91

Table C.7 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of samples used for training $S_{\#}$, with $N = 500$.

Table C.7: Results in terms of mean C_{llr} and accuracy A for the SVM method for varying number of samples to train $S_{\#}$, with $N = 500$.

Number of samples ($S_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
100	0.48	0.51	0.06	0.83
250	0.55	0.48	0.07	0.84
750	0.50	0.44	0.06	0.85
1000	0.49	0.43	0.06	0.86
2000	0.49	0.43	0.06	0.86
3000	0.49	0.43	0.06	0.86
5000	0.46	0.41	0.05	0.86
6000	0.46	0.41	0.05	0.86

XGBoost
Table C.8 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying sample length N.

Table C.8: Results in terms of mean C_{llr} and accuracy A for the XGBoost method for varying sample lengths N, with $F_{\#} = 200$

Sample length (N)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
250	0.70	0.65	0.05	0.78
500	0.56	0.46	0.09	0.85
750	0.49	0.33	0.16	0.88
1000	0.51	0.24	0.27	0.89
1250	0.64	0.18	0.47	0.89
1500	1.02	0.14	0.88	0.85

Table C.9 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of samples used for training $S_{\#}$, with $N = 500$.

Table C.9: Results in terms of mean C_{llr} and accuracy A for the XGBoost method for varying number of samples to train $S_{\#}$, with $N = 500$.

Number of samples ($S_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
500	1.10	0.56	0.54	0.81
1000	1.24	0.51	0.73	0.83
2000	0.82	0.49	0.32	0.84
3000	0.63	0.45	0.32	0.84
5000	0.60	0.47	0.13	0.84
6000	0.57	0.46	0.11	0.85
7000	0.55	0.45	0.10	0.85

C.3. LR score results

Common source LR score

Table C.10 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying sample length N, with $F_{\#} = 50$.

Table C.10: Results in terms of mean C_{llr} and accuracy A for the common source LR score method for varying sample lengths N, with $F_{\#} = 50$

Sample length (N)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
250	0.78	0.75	0.03	0.73
375	0.68	0.64	0.04	0.78
500	0.63	0.58	0.05	0.81
625	0.56	0.50	0.06	0.83
750	0.54	0.46	0.08	0.84

Table C.11 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of frequent words $F_{\#}$, with $N = 500$.

Table C.11: Results in terms of mean C_{llr} and accuracy A for the common source LR score method for varying number of frequent words $F_{\#}$, with $N = 500$.

Number of frequent words ($F_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
20	0.67	0.63	0.04	0.78
30	0.64	0.60	0.05	0.80
40	0.62	0.57	0.05	0.81
50	0.64	0.59	0.05	0.80

Specific source LR score

Table C.12 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of specific speaker samples $S_{\#s}$, with $F_{\#} = 20$ and $N = 250$.

Table C.12: Results in terms of mean C_{llr} and accuracy A for the specific source LR score method for varying number of specific speaker samples $S_{\#s}$, with $F_{\#} = 20$ and $N = 250$.

Sample length (N)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
2	0.79	0.73	0.07	0.72
4	0.74	0.67	0.08	0.75
6	0.69	0.61	0.08	0.78
8	0.68	0.60	0.08	0.78
10	0.69	0.60	0.10	0.78

Table C.13 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying sample length N, with $F_{\#} = 20$.

Table C.13: Results in terms of mean C_{llr} and accuracy A for the specific source LR score method for varying sample lengths N, with $F_{\#} = 20$.

Sample length (N)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
200	0.75	0.69	0.06	0.74
300	0.68	0.60	0.09	0.78
400	0.64	0.53	0.11	0.80
500	0.60	0.46	0.14	0.82
600	0.59	0.40	0.19	0.83

Table C.11 shows the table with additional results in terms of C_{llr}^{min} and C_{llr}^{cal} for varying number of frequent words $F_{\#}$, with $N = 500$.

Table C.14: Results in terms of mean C_{llr} and accuracy A for the specific source LR score method for varying number of frequent words $F_{\#}$, with $N = 500$.

Number of frequent words ($F_{\#}$)	C_{llr}	C_{llr}^{min}	C_{llr}^{cal}	Accuracy
20	0.57	0.41	0.15	0.84
30	0.58	0.44	0.15	0.83
40	0.59	0.44	0.14	0.83
50	0.58	0.44	0.15	0.83