A Quantitative Comparison of the Performance of Likelihood Ratio Systems in Trace-Reference Problems

W.G. Versteegh

Delft University of Technology

TU Delft

A Quantitative Comparison of the Performance of Likelihood Ratio Systems in Trace-Reference Problems

by

W.G. Versteegh

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday May 8, 2024 at 14:00 PM

Student number: 4595947
Project duration: June 6, 2023 – May 8, 2024
Thesis committee: Dr. R.J. Fokkink, TU Delft, associate professor
Dr. N. Parolya, TU Delft, assistant professor
Dr. Ir. R.J.F. Ypma, Netherlands Forensic Institute, Principal Scientist

TU Delft

Preface

This thesis has been submitted as the final requirement to obtain the title of Master of Science in Applied Mathematics at Delft University of Technology. The research was conducted in collaboration with the Netherlands Forensic Institute (NFI), from June 2023 until April 2024.

During the past ten months, I had the pleasure of diving into the world of forensic statistics. As a math student whose parents have studied law, I never saw a way to bridge the two fields. When the opportunity arose to do my thesis at the NFI I was immediately intrigued by the prospect of doing mathematical research at an institute that plays a big role in supporting the judicial process.

Having finished my internship I can say that it has been a very pleasurable experience working with so many experts in their respective fields. No matter who you were talking to, you could immediately tell that everyone was not only good at what they were doing but also excited by their work. I learned so many things from talking to everyone at the NFI.

First and foremost, I want to thank Rolf Ypma for the weekly meetings and your guidance. Your supervision and tips every time we met taught me a lot more than just the mathematics that we were discussing. It was a pleasure learning from you.

I also want to sincerely thank Robbert Fokkink for taking a seat on my committee and providing me the opportunity to do my thesis at the NFI. I want to thank Nestor Parolya for his guidance during my thesis project. Although you were often busy, we still found time for each other to meet and discuss the progress.

Personal thanks go to Ivo Chen and my dad for checking parts of my thesis and giving all their helpful advice, for my thesis and life. Another thanks goes to Louise Leibbrandt for making the internship at the NFI even more fun than it already was.

Lastly, I sincerely want to thank all my friends. You always have been there for me throughout my whole time as a student, a time that I have enjoyed greatly because of you, and for that, I cannot thank you enough.

W.G. Versteegh
Delft, April 2024

Abstract

In forensic science, the strength of evidence is calculated mainly by statistical models called likelihood ratio systems. In court cases, the specific-source likelihood ratio system is used by forensic scientists to determine if a trace originates from a known reference, called the trace-reference problem. However, collecting sufficient data to create a specific source model may be time-consuming and costly. If the number of court cases becomes too high this could be problematic. Therefore there is a need for other models that can perform as well as a specific-source model if it is infeasible.

A common-source model could be a solution, as this model can be re-used over cases. To this end, we introduce two common-source systems: a common-source feature-based system and a common-source score-based system. We compare their performance to a specific-source score-based system in a trace-reference setting. The simulations show that the common source feature-based method is the best-performing likelihood ratio system if the dimensionality is not too high, and the sources are equally variable. The analysis shows that the common-source score-based method can work as effectively as a specific-source score-based model in certain scenarios.

Additionally, we researched a preprocessor, known as percentile rank, which aims to consider typicality for score-based methods. For the common-source score-based system, using a percentile-rank preprocessor can improve the performance for large sample sizes, while considering the rarity of the measurements.

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
LR	Likelihood ratio
SS	Specific source
CS	Common source
CSFLR	Common source feature-based likelihood ratio
CSSLR	Common source score-based likelihood ratio
CSSLR-PR	Common source score-based likelihood ratio using percentile rank as a preprocessor
SSFLR	Specific source feature-based likelihood ratio
SSSLR	Specific source score-based likelihood ratio

Symbols

Symbol	Definition
C_{llr}	Log-likelihood ratio cost
N	Sample population size including specific source
N_{as}	Sample population size without specific source, or number of alternative sources
d	Dimension of the data, number of features in the measurements
r	Number of measurements per alternative source
r_{ss}	Number of measurements from the specific source
$f(x, y)$	Probability density (for continuous data) or probability mass function (for discrete data) of the joint distribution of x and y
$F(x)$	Cumulative probability distribution function
$F_n(x)$	Empirical distribution function
μ	Population mean
μ_{ss}	Specific source mean
σ_b	Between-source standard deviation scalar
Σ_b	Between source covariance matrix
σ_w	Within-source standard deviation scalar
$\sigma_{w,i}$	Within-source standard deviation scalar belonging to source i
$\sigma_{w_{ss}}$	Specific source within standard deviation
Σ_w	Within-source covariance matrix
$\Sigma_{w,i}$	Within-source covariance matrix belonging to source i
I	$d \times d$ identity matrix
$\delta(x, y)$	Score function of x and y

1

Introduction

When evidence is found at a crime scene, the primary concern is the origin of the evidence. How strong the evidence is, is decided through a collaboration between legal experts and forensic scientists. Forensic scientists aim to determine the strength of the evidence in light of two competing hypotheses: the prosecutor's hypothesis, stating that two pieces of evidence originate from the same source, and the defence hypothesis, stating the two pieces of evidence do not originate from the same source. This is of particular interest in court cases where a suspect is present and we seek to establish if a trace can be linked to a known suspect, referred to as a trace-reference problem.

To address the trace-reference problem, a statistical model called a likelihood ratio system is made. This LR system takes evidence from the suspect and the disputed trace. It then outputs a number, the likelihood ratio, indicating the strength of evidence for prosecution or defense. Likelihood ratios stem from a logical perspective on criminal court cases called the Bayesian view (see section 2) where the probabilities of the competing hypothesis are calculated after observing the evidence.

Ideally, in the case of the trace-reference problem, one would preferably make a likelihood ratio system that is tailor-made for this suspect: this is called a 'specific source' approach. Unfortunately, this is often impractical due to the need for extensive data from a specific source. Collecting data to make a specific source system can be costly and time-consuming.

One particular point of interest is the applicability of 'common source' systems to trace-reference problems. Common source systems are originally designed to determine if two unknown traces are from the same unknown source. It aims to do this by making a model of the whole population instead. Making a common source model would be a one-time investment since you could re-use this over cases. Some argue that a common source approach cannot be applied to a trace-reference problem. [14] It answers a fundamentally different question than the specific source case, where one of the traces comes from a known source.

However, research by Vergeer [21] indicates that common source methods can still be beneficial in trace-reference problems if a specific source model is impractical. These statements were however qualitative, and do not take into account the practical complications that come into play when constructing likelihood ratio systems. In this thesis, we aim to make these statements quantitative to see how these statements hold in practice. We will do this by simulating datasets of various kinds, each in a way different to see which factors are important in the performance of likelihood ratio systems. We will compare two types of score-based likelihood ratio systems and a common source feature-based approach.

Furthermore, an open issue in forensics is the integration of typicality in score-based likelihood ratio systems. Score-based systems transform the features to a score, leading to a loss in evidential value if the rarity of the features is not considered. Some argue that scores that do not incorporate rarity into account are inappropriate as scores [11]. One solution to this problem could be to use a percentile rank preprocessor, which in some cases can improve the performance of likelihood ratio systems [17].

In this thesis, we will also research cases in general where likelihood ratio systems could benefit from using a percentile rank preprocessor.

1.1. Problem statements

We mentioned the statements proven by Vergeer [21] about the performance of likelihood ratio systems. In particular, he proves the following inequalities about the expected performance for likelihood ratio systems:

$$SSSLR(X,Y) \geq CSSLR(X,Y) \tag{1.1}$$
$$CSFLR(X,Y) \geq CSSLR(X,Y) \tag{1.2}$$
$$CSLR(X,Y) \geq \text{prior} \tag{1.3}$$
$$\tag{1.4}$$

These statements read as follows: A specific source score-based model outperforms a common source score-based model, a common source feature-based model outperforms a common source score-based model, and lastly using any common source model is still better than doing nothing. As mentioned, we aim to make these statements more quantitative to see how this holds in practice. We therefore have the following research questions:

- Does a specific source score-based model always outperform a common source score-based model
- How much better do feature-based models perform compared to score-based models?
- Can a common source approach always be useful for a trace-reference problem? If yes, in what scenarios?
- Can a percentile rank preprocessor include the rarity of measurements in score-based methods?

1.2. Thesis outline

The framework surrounding likelihood ratios is introduced in chapter 2. Both the common source and specific source model are introduced. Furthermore, we explain the difference between feature-based and score-based likelihood ratio systems.

After the theoretical framework, we consider the practical things that come into play when constructing likelihood ratio systems. We introduce our method of simulating the data in chapter 3; a two-level normal-normal model by Lucy and Aitken [6] which is a classical model in forensic evidence.

In chapter 4 we dive into how to make a likelihood ratio system from a given dataset, based on a paper by the NFI [8]. We go over the steps taken and discuss the choices made when constructing a likelihood ratio system.

In chapter 5 we discuss our way of evaluating the performance of a likelihood ratio system. We introduce the metric that we will use to compare the performance of likelihood ratio systems.

In chapter 7 we study and compare the performances of likelihood ratio systems. We simulated datasets of various sizes for our background population and the amount of specific source data. We will research these situations for low and medium levels of uncertainty surrounding our sources.

As an extension to this, we study in chapter 8 the scenario in which each source has its unique inherent uncertainty surrounding its measurements. We will see that whenever each source has

When developing a likelihood ratio system, it's important to consider the number of features in your data. It will be seen in chapter 9 that for high-dimensional data, feature-based likelihood ratio systems will function poorly.

In chapter 10 we will study the effects of using a percentile rank preprocessor as a way of including typicality in score-based likelihood ratio systems. It is argued that for a specific source score-based system, typicality is already taken into account. For common source score-based systems it is seen that using this preprocessor could be beneficial for its performance.

2

The framework of likelihood ratios

Central to this study are the likelihood ratio systems. These are statistical models that given an input of observations output a number called a likelihood ratio, or LR for short. These LRs serve to aid judges in quantifying the weight of evidence and how it relates to the proposed hypotheses. In this chapter, we lay down the framework and methods for likelihood ratios.

Criminal court cases often revolve around two competing hypotheses of events that may have taken place. In the prosecutor's hypothesis (H_1) the suspect will have committed a crime, in the defence hypothesis (H_2) another person may be the perpetrator. The judge will have to rule on whether she is convinced, by lawful evidence E of the truth of H_1. In mathematical language, she has to decide whether the posterior odds

$$\frac{\mathbb{P}(H_1|E)}{\mathbb{P}(H_2|E)}$$

are high enough. Simply applying Bayes' rule we can rewrite these odds as the multiplication of a ratio of likelihoods and prior odds:

$$\underbrace{\frac{\mathbb{P}(H_1|E)}{\mathbb{P}(H_2|E)}}_{\text{Posterior odds}} = \frac{\frac{\mathbb{P}(E|H_1)\mathbb{P}(H_1)}{\mathbb{P}(E)}}{\frac{\mathbb{P}(E|H_2)\mathbb{P}(H_2)}{\mathbb{P}(E)}} = \underbrace{\frac{\mathbb{P}(E|H_1)}{\mathbb{P}(E|H_2)}}_{\text{Likelihood ratio}} \cdot \underbrace{\frac{\mathbb{P}(H_1)}{\mathbb{P}(H_2)}}_{\text{Prior odds}} \tag{2.1}$$

The role of the forensic expert, or expert witness, is to aid in this decision-making,by explaining to what extent evidence supports one hypothesis over the other. As the forensic expert can only give testimony on a small part of the facts and circumstances in the case (namely the evidence that lies in their area of expertise), they cannot make statements on the prior and thus posterior odds. This separates the roles involved in the court.

The expert can make statements on how likely it is to find the evidence under each of the hypotheses - and thus on the likelihood ratio $\frac{\mathbb{P}(E|H_1)}{\mathbb{P}(E|H_2)}$. The judge can use this LR to update her prior belief (to what extent she believed the hypotheses before considering the evidence) to the posterior odds.

In this formulation, we are implicitly conditioning on the relevant background information I which is always present. This could be relevant in certain cases. An example is given in Vergeer et al. (2016)[22] of glass splinters that are found on the clothes of a culprit. If the relevant evidence type is glass but the culprit is someone who often works with glass, then this has implications for the background population of alternative sources and thus should affect the LR. This is however out of the scope, and it is understood that whenever the background information could be relevant this relevancy is already captured in the LR.

2.1. Common source problems

The first type of problem of interest in forensics is the common source problem. In this problem, it is of interest to find out if two traces originate from the same, but unknown source This is often important in the investigative phase of crime where it is unsure if two seemingly unrelated crimes can be related. This might mean the difference between the police trying to find one culprit instead of two. If for example DNA is found at two different crime scenes, trying to find out if the DNA is from the same person or not can make a big difference in the amount of searching law enforcement needs to do.

These types of questions are what will be referred to as a common source problem. A key characteristic of this problem is that both traces come from unknown sources, which are assumed to be from the same background population. Given an evidence set consisting of trace evidence X and reference evidence Y, we have the following hypotheses, corresponding to the trace-reference problems:

$H_{1,c}$: X and Y originate from the same unknown source.
$H_{2,c}$: X and Y originate from two different unknown sources.

A high LR in a common source scenario then captures the idea that, after observing the evidence, it has become more likely that the two pieces of evidence are from the same unknown source. Conversely, a low LR would indicate that it is more likely for this evidence to occur assuming these are not from the same source.

2.2. Specific source problems

In contrast to the common source problem, it is also often of interest if a trace from an unknown source can be matched to a known reference source. This problem is known as the specific source problem. These problems are important in court cases where a suspect is already known.

Given evidence set E consisting of trace evidence X and reference evidence Y, we have the following hypotheses, corresponding to the trace-reference problems:

$H_{1,s}$: X and Y originate from the same known specific source.
$H_{2,s}$: X and Y originate from two different sources. The source of Y is known.

2.3. Feature-based likelihood ratio systems

If possible, one would like to have a feature-based likelihood ratio system. In this type of likelihood ratio system, a direct probability distribution of the features is made using the data available. This comes down to making a probabilistic model of the evidence under H_1 and under H_2, to give an enumerator and denominator of your LR. Feature-based methods are preferable since it is a direct model of the evidence of interest without a step in between.

We have the following definitions for both specific source feature-based LR systems and common source feature-based LR systems as written by Vergeer (2023) [21]. We denote f to mean both the probability distribution function in the case of discrete data and the probability density function in the case of continuous data. In our thesis, only continuous simulated data is considered, but it is understood that the general theory works for both continuous and discrete data.

For the SSFLR, we have given evidence set X, Y with trace evidence X and reference evidence Y:

$$\text{SSFLR} = \frac{f(X, Y|H_{1,s})}{f(X, Y|H_{2,s})}$$

The SSFLR however is often a theoretical optimum. In practice, it would be too much effort and costly to keep sampling a specific source and make a feature-based model. Furthermore, we can argue that all evidence is generated from an overall background distribution and that the specific source under consideration is a realization of a random source. This favors again a common source approach for feature-based likelihood ratio systems.

If you want to make a feature-based model, it is often more logical to build a common source feature-based model, CSFLR. This makes instead a probabilistic model of the features of the whole population.

Given evidence set X, Y with evidence X from the first trace and evidence Y from the second trace, we have the definition for the CSFLR:

$$\text{CSFLR} = \frac{f(X, Y|H_{1,c})}{f(X, Y|H_{2,c})}$$

An example of a common source feature-based model is found in DNA. We have a good understanding of DNA and the frequencies with which alleles occur in the population. Assume we find a DNA profile X on crime scene A and another profile Y on crime scene B. We would like to know if this DNA came from the same but unknown person. The numerator of the CSFLR is the probability of observing the DNA profiles if they originated from the same person. The denominator of the CSFLR is the probability of observing the DNA profiles if they came from different people. This coincides with the overall frequency of the DNA profiles in the population. Intuitively, if there is a match between DNA profiles but it is on alleles that occur frequently in the population, the fact that there is a match should not be strong evidence. Conversely, if there is a match on alleles that do not occur often in the population, then the fact that there is a match is stronger as evidence.

2.4. Score-based likelihood ratio systems

When a feature-based model is infeasible, one resorts to score-based models. In this approach, one looks at the distribution of a function of the features, rather than the features themselves. Score-based LR systems exhibit considerable advantages compared to feature-based models. It can be difficult to create a direct probability model for certain types of evidence, or when there is insufficient data available for the number of features present in the data type.

Score-based methods circumvent these issues since they can handle higher dimensional data more easily by reducing the features to a score using a function of the input data. This function is typically called a *score* function, denoted $\delta(X, Y)$.

Score-based approaches are widespread in forensics and have found applications in speaker recognition [7], glass [20], telecommunications [3], and MDMA tablets [2]. For each data type, it can be unclear which score function is a good idea and it can differ from type to type what score functions are useful.

For the SSSLR, we have given evidence set X, Y with trace evidence X and reference evidence Y:

$$\text{SSSLR} = \frac{f(\delta(X, Y)|H_{1,s})}{f(\delta(X, Y)|H_{2,s})}$$

An example of a specific source score-based likelihood ratio system is found in authorship attribution in text messages. In 2020, EncroChat was hacked by a collaboration between the Dutch and French police [16]. From this hack, a lot of communication between criminals could be read by the police. If we now have a case in which we have a suspect for a crime, we would like to know if the messages that are found to be incriminating can be linked to the suspect at hand, hence the specific source notion.

It is however unclear how to make a good probability distribution of the text messages that someone sends. However, we could try and come up with a score function that describes how similar or dissimilar the trace text messages are to the text messages that we know originate from the suspect.

Note that this specific source approach is only feasible when you have a lot of text that is undeniably from the suspect. For other data types, there might not so easily be this much data at hand, while also being too difficult to have a feature-based model. In that case, we resort to a common source score-based model, CSSLR. For the CSSLR, we have given evidence set X, Y with evidence X from the first trace, and evidence Y from the second trace:

$$\text{CSSLR} = \frac{f(\delta(X, Y)|H_{1,c})}{f(\delta(X, Y)|H_{2,c})}$$

An example of a common source score-based model is speaker recognition, by Morrison. [11]. If we have two audio files of someone speaking, we would like to know if the recordings are from the same person. Audio files are too highly dimensional to make a good feature-based model and there might not be enough data present to make a functional specific source model.

2.5. Similarity and typicality

Two central concepts in forensics are similarity and typicality. Similarity, as the name suggests, captures how similar the trace and reference are. Typicality on the other hand tells us how common or rare the observations are. If we look at for example the likelihood ratio for common source feature-based methods, we can see that similarity and typicality are incorporated in the definition:

$$\text{CSFLR} = \frac{f(X, Y|H_{1,c})}{f(X, Y|H_{2,c})}$$

The numerator captures how frequently both X and Y occur if they are from the same source, whereas the denominator captures how often X and Y occur in general if they are not from the same source. Morrison (2016) gives an interpretation as follows:

$$\text{LR} = \frac{f(X, Y|H_1)}{f(X, Y|H_2)}$$

$$\text{LR} = \frac{\text{Probability of jointly observing the measurements } X \text{ and } Y \text{ when they come from the same source}}{\text{Probability of jointly observing the measurements } X \text{ and } Y \text{ when they come from different sources}}$$

$$\text{LR} = \frac{\text{Similarity of measurements } X \text{ and } Y}{\text{Typicality of the measurements of } X \text{ and } Y \text{ in the general population}}$$

It is understood that whenever X and Y originate from the same source, their joint probability density should be high and is a measure of similarity between X and Y.

However, similarity and typicality disappear in the definition once you introduce score-based methods. Let us take for example the definition of a CSSLR:

$$\text{CSSLR} = \frac{f(\delta(X, Y)|H_{1,c})}{f(\delta(X, Y)|H_{2,c})}$$

It is possible that two different evidence pairs lead to the same score even though the measurements themselves may differ in rarity. Formally, for pairs $(\mathbf{X_1}, \mathbf{Y_1})$ and $(\mathbf{X_2}, \mathbf{Y_2})$ we could have the scores of these paired measurements, $\delta(\mathbf{X_1}, \mathbf{Y_1})$ and $\delta_(\mathbf{X_2}, \mathbf{Y_2})$ are equal, even though it might be that the pair of observations $(\mathbf{X_1}, \mathbf{Y_1})$ is more common than the pair $(\mathbf{X_2}, \mathbf{Y_2})$. In casework, the rarity of the observations could play a big role in evaluating evidence. Rarer evidence is of higher informative value than common evidence, and some score-based methods do not capture this rarity directly.

The relevance of similarity and typicality is perhaps better illustrated using an example. Let's say the data we are investigating are lengths of people, and we observe a trace length of 182 cm. We also have a reference suspect whose length is 180 cm. In a score-based approach and using a distance measure like the absolute difference, this would amount to a score of 2. Now in another case where length is a factor, we have a trace with a length of 222 cm and a reference suspect whose length is 220 cm. Also in this case we would have a score of 2. However intuitively, there should be more information in the second case since lengths of 220 and 222 cm occur very little in the population. Even though we observe the same score, in the second case there should be more evidential value due to the features themselves being uncommon.

It is due to this fact that it is also a desirable property to incorporate rarity in score-based models. Morrison and Erzinger (2017) [11] propose that score-based methods should always include some typicality to be a proper method if you want to use it in practice. This is however a non-trivial task.

Furthermore, there is also (even though less) evidential value in using similarity only. If we go back to our example of lengths, even though in the first case we have heights that are very common in the

population, there is still some information to be gained by observing the difference. The scorers we consider are similarity-only scores. Vergeer [21] shows that LR systems that do not incorporate rarity can still be used since these systems, on average, perform better than using only the prior.

However, it is still of interest to study if we can incorporate typicality into our score-based LR systems. In section 4.1 we will discuss a possible preprocessor based on percentile rank that could incorporate typicality in score-based LR systems. In chapter 10 we do a simulation study to see if models, that use this preprocessor, account for the typicality of the measurements and if these models can perform as well as models that do not include typicality.

2.6. Discriminative property of LR systems

A property of well-performing LR systems is its ability to separate between the densities of the data or scores under H_1 and H_2. Ramos et al. (2013)[15] define *discrimination* as the degree of separation among H_1-true and H_2-true LR values. The smaller the degree of overlap, the higher discriminative power the LR system has and the better it will be able to decide if input traces are from the same source. [22]

Figure 2.1 is an example of a situation in which a hypothetical LR system has high discriminative power. The data represent scores that are outputted by an LR system after inputting a trace and a reference.

In this concrete example, we have 1000 H_1 samples that come from a $N(-3, 1)$ distribution, while we also have 1000 H_2 samples that come from a $N(3, 1)$ distribution. These distributions have little overlap. If for your case you would observe a score of -3, it would result in a high LR since this value is the mean under H_1, but not common at all under H_2.

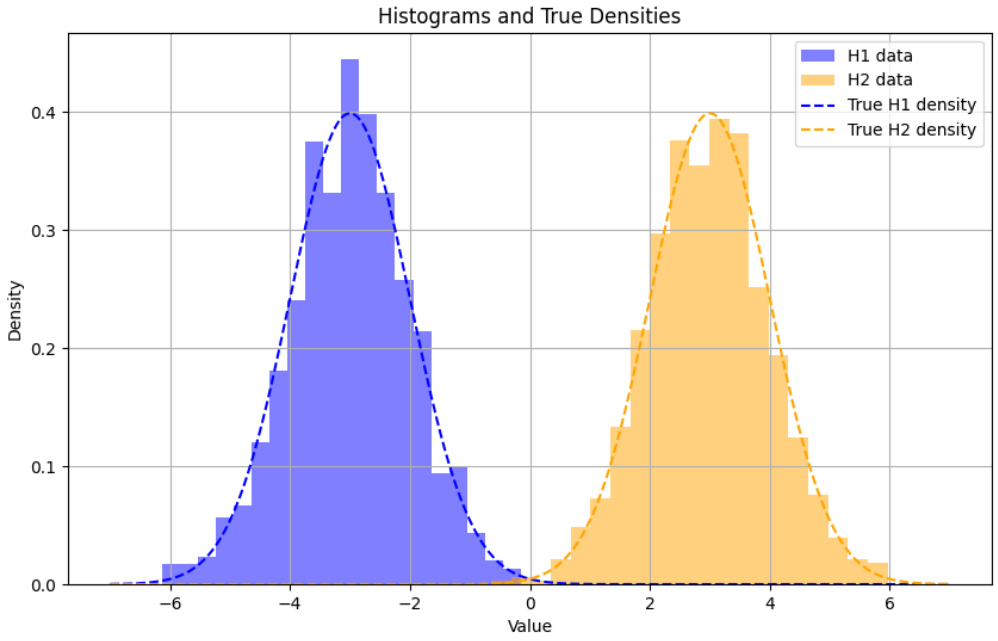


Figure 2.1: Surrogate example of a situation in which the H_1-true and H_2-true densities separate well

In this case, if we observe a score δ that equals -3, the true LR would equal

$$\begin{aligned}\text{LR} &= \frac{f(\delta|H_1)}{f(\delta|H_2)} \\ &= \frac{\frac{1}{\sqrt{2\pi\sigma_1^2}}\exp(-\frac{1}{2\sigma_1^2}(\delta-\mu_1)^2)}{\frac{1}{\sqrt{2\pi\sigma_2^2}}\exp(-\frac{1}{2\sigma_2^2}(\delta-\mu_2)^2)} \\ &= \frac{\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(-3-(-3))^2)}{\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(-3-3)^2)} \\ &= \frac{\exp(0)}{\exp(-18)} \approx 6.57 \cdot 10^7\end{aligned}$$

In practice, you don't know the exact distribution of the data, and other methods have to be used to obtain an LR, see section 4.6. Here in our case, the minimum of the H_2 simulated scores equals about -0.64 and so the score falls outside of the support. See section 4.7 for a more elaborate discussion on how to handle cases when you have to extrapolate.

In figure 2.2 is an example of a situation in which a hypothetical LR system has lower discriminative power. Here in this concrete example, the H_1 data now comes from a $N(-\frac{1}{2}, 1)$ distribution, while the H_2 data comes from a $N(\frac{1}{2}, 1)$ distribution. These distributions exhibit a fair bit of overlap.

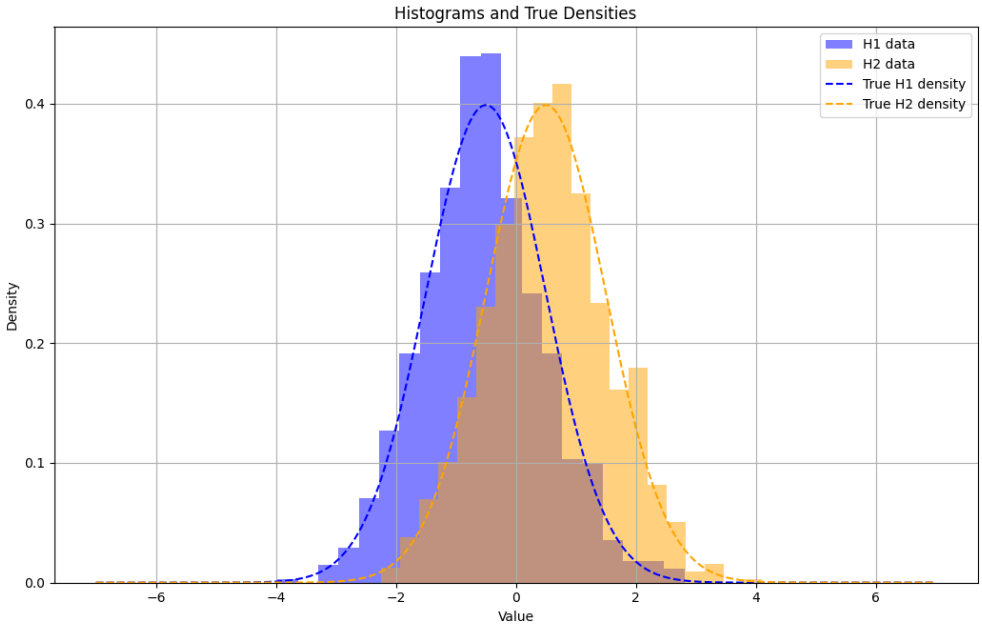


Figure 2.2: Surrogate example of a situation in which the H_1-true and H_2-true densities do not separate as well

If we now observe a score of $-\frac{1}{2}$, the mean of the H_1 data just like in the previous example, the true

LR equals

$$\begin{aligned}
\mathsf{LR} &= \frac{f(\delta|H_1)}{f(\delta|H_2)} \\
&= \frac{\frac{1}{\sigma_1\sqrt{2\pi}}\exp(-\frac{1}{2\sigma_1^2}(\delta-\mu_1)^2)}{\frac{1}{\sigma_2\sqrt{2\pi}}\exp(-\frac{1}{2\sigma_2^2}(\delta-\mu_2)^2)} \\
&= \frac{\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(-0.5-(-0.5))^2)}{\frac{1}{\sqrt{2\pi}}\exp(-\frac{1}{2}(-0.5-0.5)^2)} \\
&= \frac{\exp(0)}{\exp(-0.5)} \approx 1.65
\end{aligned}$$

2.7. Relevant background population

When constructing likelihood ratio systems in forensics, it's crucial to consider relevant background populations. In both the specific source approach and common source approach, the characteristics of the background population come into play when considering the H_2 hypothesis, which states that the trace or traces come from an alternative source. The origin of the set of alternative sources however can have a big impact on the LR that you want to compute.

For example; We find a trace DNA profile Γ at a crime scene and you have a known suspect that exhibits the same DNA profile Γ. To output an LR, you would need to know how common profile Γ is in the relevant background population. This however immediately pinpoints the difficulty, since deciding which background population to use has implications for the rarity of the characteristics. Do you use a background database from the city that the crime occurred in or the whole region? Or do you consider the country?

Another example where the relevancy of the background set can again be found in lengths. Observing a length of 182 in a country like the Netherlands will have less evidential value than in a country like Timor Leste, which is on average the shortest country in the world.

The choice of the background population has a significant impact on the LR that you would observe. Certain characteristics of one choice background population might or might not be as present when considering a different background population. Understanding the demographic characteristics of the relevant background is crucial for an accurate LR.

In this thesis it is assumed that the simulated alternative sources are representative of relevant background sources for the cases at hand. It is however important to keep in mind that when working with real-life data, your dataset might not be representative of the type of case that you are working with.

3

Two-level normal-normal model

One commonly used model for forensic evidence is a two-level normal-normal model. This is a classical model in forensic statistics proposed by Lucy and Aitken [6] where both the between-source variation and within-source variation follow a multivariate normal distribution.

As a motivation, we again go back to our example of lengths. For grown-ups in a certain population, lengths are approximately normally distributed between-sources. When measuring someone's length this varies around someone's "true" length. This randomness is captured in the within-variation.

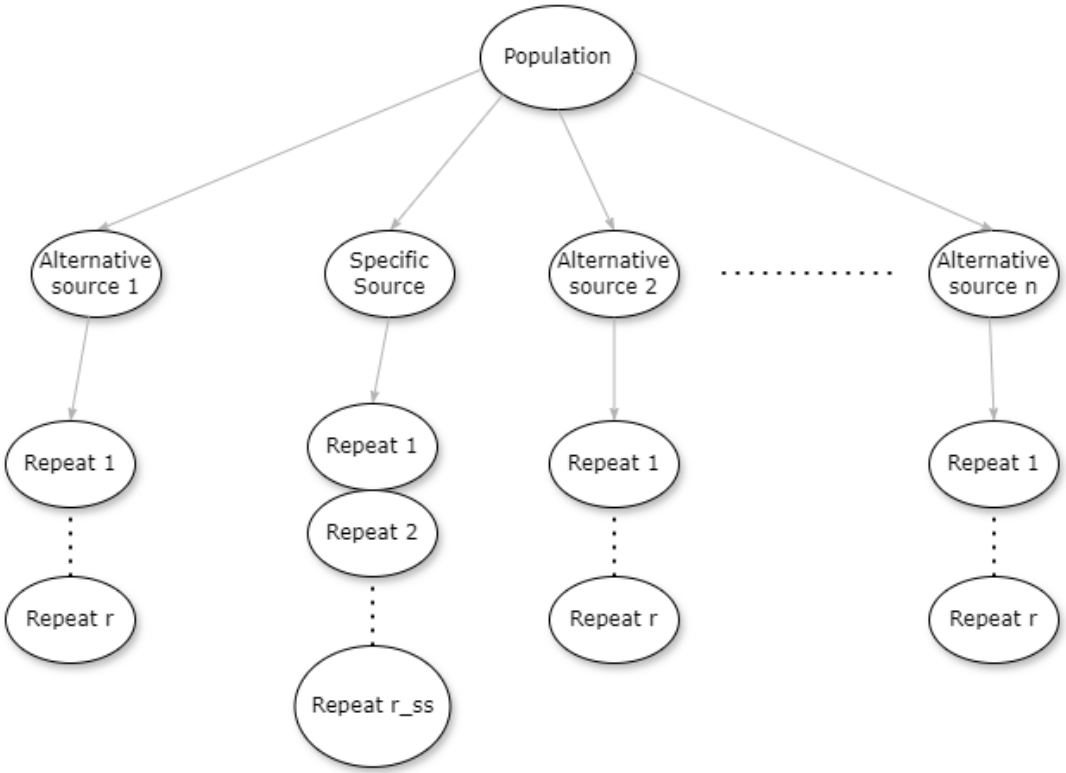


Figure 3.1: Schematic overview of the structure of the datasets considered. Every source mean is generated at the top level from a central population distribution having its own mean and between standard deviation. Then for each source, a number of repeats are generated using the generated mean and the within standard deviation.

3.1. Between-source level

At the highest level each source mean is drawn from the population distribution, using a population mean $\mu_{\mathbf{0}} \in \mathbb{R}^d$ and between-sources covariance matrix $\Sigma_b \in \mathbb{R}^{d \times d}$. The central mean μ_0 determines the point around which our population lies. Since we simulate our data and will center our data anyway (see 4.1), the mean $\mu_{\mathbf{d}}$ will taken to be the null vector **0** of length d.

Unless specified differently, we will simulate the source means from a multivariate normal distribution. However, it is likely that in real life the data is not normally distributed. Normality assumption at the between-source levels is a strong assumption.

The between-source covariance matrix Σ_b determines how varied the features of the sources in our population are, and how correlated they are within our general population. In our case, the covariance matrix is just a multiple of the identity matrix, scaled by the square of the *between-sources standard deviation* σ_b, so $\Sigma_b = \sigma_b^2 I$.

Note that this assumes that the features are independent and all have the same variance, which is a strong assumption. Often in practice, one looks at the estimated sample covariance matrix and decides if it might be necessary to apply dimension reduction methods or utilize methods that can deal with the dependencies.

Taking the same variance for each feature simplifies the intuition mostly and makes the idea of a more spread out population captured in one constant σ_b. Since we will preprocess anyway, even if each feature $f \in \{1, \ldots, d\}$ had their own between standard deviation $\sigma_{b,f}$ after standardization preprocessing the data for that particular feature will have a standard deviation of 1, no matter what the initial $\sigma_{b,f}$ was. We thus have that, given d-dimensional null vector **0** and $d \times d$ covariance matrix $\Sigma_b = \sigma_b^2 I$, the means of our sources are distributed according to:

$$\mu \sim MVN(\mathbf{0}, \Sigma_b)$$

Depending on our sample size N of choice, we will draw for $i = 1, \ldots, N$ several $\mu_{\mathbf{i}}$ as means for our sources according to this distribution. Unless specified, the mean of the *specific source* is also drawn from this distribution. This corresponds to the case where our suspect is a random draw from our background population.

3.2. Within-source level

After a mean has been generated for a source $i = 1, \ldots, N$, several repeat measurements r_i are generated using another multivariate normal distribution with the generated mean $\mu_{\mathbf{i}}$ and the *within-source covariance matrix* $\Sigma_w \in \mathbb{R}^{d \times d}$. In our case, we have that $\Sigma_{w,i} = \sigma_{w,i}^2 I$. Measurement repeats $j = 1, \ldots, r_i$ from a given source i are therefore distributed according to a multivariate normal distribution:

$$\mathbf{X_{j,i}} | \mu_i \sim MVN(\mu_{\mathbf{i}}, \Sigma_{w,i})$$

At the within-source level, the normality assumption is more reasonable than at the between-source level. The within-source randomness is usually incorporated to account for all the randomness that occurs when taking measurements from a source. This could be due to measurement errors or intrinsic randomness from the source. This randomness is captured in the within-source standard deviation, and here the normality assumption is more sensible.

3.3. Joint normal evidence distributions

From general theory on multivariate normals, if a random vector **X** follows a $MVN(\mu_{\mathbf{X}}, \Sigma_X)$ distribution with mean μ_X and covariance matrix Σ_X then we know that we can rewrite X as $X = \mu_X + \mathbf{AZ}$, where **A** is a matrix satisfying $\mathbf{AA}^T = \Sigma_X$, and $\mathbf{Z} = (Z_1, \ldots, Z_d)$ a d-dimensional random vector where each $Z_i, i \in \{1, \ldots, d\}$ follows a one-dimensional standard normal distribution $N(0, 1)$, and for $i \neq j$, we have

that Z_i and Z_j are independent.

Using this general theory we can write down the joint distribution X and Y for the common source and specific source. These full distributions are a generalization to higher dimensions, a one-dimensional derivation is also present in Neumann and Ausdemore [13]. A full derivation can be found in the appendix A.

Joint evidence distribution under common source hypothesis

We have the generative models for both unknown trace evidence X and Y, under the common source hypothesis:

$$(\mathbf{X}, \mathbf{Y})|H_{1,c} \sim MVN(\mu^*, \Sigma_{1,c}) \tag{3.1}$$
$$(\mathbf{X}, \mathbf{Y})|H_{2,c} \sim MVN(\mu^*, \Sigma_{2,c}) \tag{3.2}$$

Where $\mu^* = \begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix}$m with block covariance matrices:

$$\Sigma_{1,c} = \begin{bmatrix} \Sigma_b + \Sigma_{w,x} & \Sigma_b \\ \Sigma_b & \Sigma_b + \Sigma_{w,x} \end{bmatrix}, \Sigma_{2,c} = \begin{bmatrix} \Sigma_b + \Sigma_{w,x} & 0 \\ 0 & \Sigma_b + \Sigma_{w,y} \end{bmatrix}$$

Note that if X and Y originate from the same source we have the same within covariance matrix. In our simulation, we typically take the null vector as the mean for our population traces. In particular we have that X and Y both have mean $\mu_x = \mu_y = \mathbf{0}$, and therefore we have $\mu^* = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$

Joint evidence distribution under specific source hypothesis

Similarly, we have for the generative models for unknown trace evidence X and known reference Y, under the specific source hypotheses:

$$(\mathbf{X}, \mathbf{Y})|H_{1,s} \sim MVN(\mu'_{\mathbf{1,s}}, \Sigma_{1,s}) \tag{3.3}$$
$$(\mathbf{X}, \mathbf{Y})|H_{2,s} \sim MVN(\mu'_{\mathbf{2,s}}, \Sigma_{2,s}) \tag{3.4}$$

Note that in this scenario, the mean vector around which the joint distribution centralizes depends on H_1 or H_2. In the H_1 case, both the trace X and reference Y centralize around the specific source mean μ_{ss} since under H_1 our trace is originating from the specific source. In the H_2 case the (unrelated) trace X now has a mean that is different from the specific source, and instead has a mean that is drawn from at random from the background population.

Under H_1, we have mean $\mu'_{\mathbf{1,s}} = \begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix} = \begin{bmatrix} \mu_{\mathbf{ss}} \\ \mu_{\mathbf{ss}} \end{bmatrix}$, with block covariance matrix $\Sigma_{1,s} = \begin{bmatrix} \Sigma_{w,x} & 0 \\ 0 & \Sigma_{w,x} \end{bmatrix}$

Under H_2, we have mean $\mu'_{\mathbf{2,s}} = \begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mu_{\mathbf{ss}} \end{bmatrix}$, with block covariance matrix $\Sigma_{2,s} = \begin{bmatrix} \Sigma_b + \Sigma_{w,x} & 0 \\ 0 & \Sigma_{w,y} \end{bmatrix}$
Also again here, we have that if X and Y are from the same (specific) source, they have the same within covariance matrix $\Sigma_{w,x}$

The trace measurement X is a draw from the specific source distribution with the same mean and covariance as the reference Y under H_1. However, under H_2, it is an independent draw from the background population.

4

Likelihood Ratio Systems

After we have generated our dataset we can construct our LR system of choice. In this chapter, we will go over the steps taken to go from a dataset to a LR system. These steps are based on the paper by Leegwater et al. [8]

4.1. Data preparation

We need to take some prerequisite steps before we can make a LR system.

Before starting we will split the dataset into a training set and a hold-out validation set. This will ensure that when we eventually want to evaluate our LR system, we can do so by using data that is not used to train the model.

In practice you might want to build multiple LR systems and see which one works best for your data. If you want to compare multiple setups for LR systems (like different score functions or calibrators) you can also split off a selection set from your training set. You can then use your smaller training set to train multiple configurations of LR systems and test them on your selection set to decide which setup works best. After choosing your setup you recombine the training and selection set, use this larger dataset to train a new LR, and validate it on the validation set that you have kept apart.

Standard preprocessor

A common method for preprocessing numerical data involves Z-score normalization, also known as standard scaler. This preprocessor will standardize the data per feature. That means that, per feature, it will transform the data such that after the transformation, the data has mean 0 and standard deviation 1.

Given a dataset consisting of measurements $\{X_1, \ldots, X_n\}$, where each $X_i \in \mathbb{R}^d$, with d the dimension of the data. We have per feature $k \in \{1, \ldots, d\}$ the data from each measurement for that feature $\{X_{1,k}, \ldots, X_{n,k}\}$. This data has mean μ_k and sample standard deviation s_k. We, therefore, have linear functions $T_k(x) = \frac{x-\mu_k}{s_k}$ that transform our data in that feature such that our transformed features will have sample mean 0 and sample standard deviation 1.

In our case, we take the training set and preprocess it in this matter. Important here is that the transformation functions made from the training set are also used to preprocess the validation set when validating the LR system. The point of splitting off a validation set is to see how your system performs on data you have never seen before. The only data you have seen before is the training data and it is only sensible to use the transformation functions you create from the training data.

Percentile rank preprocessor

Another preprocessing method is a percentile rank preprocessor, as proposed in Matzen et al. [9]. This preprocessing method aims to incorporate typicality in the score, see section 2.5.

The percentile rank preprocessor, just like the standard preprocessor, transforms the features of a vector feature-wise. In particular, again for feature $k \in \{1, \ldots, d\}$ it takes from each measurement its outcome in the k-th feature $\{X_{(1,k)}, \ldots, X_{(n,k)}\}$ and computes the empirical distribution function for that feature

$$F_k(x) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{X_{(j,k)} \leq x\}$$

The idea behind using percentile rank is that after the transformation, values more common will be further apart while values in the tail will be more attracted to each other after using the above percentile rank transformation [9]. Sergidou et al. [17] show that percentile rank can have benefits for forensic speaker comparison.

4.2. Common source feature-based approach

In this section, we show how we construct a common source feature-based model. Recall that with the feature-based approach, we mean calculating the LR using a distribution of the features directly. In particular, for the CSFLR we are after the following LR:

$$\text{CSFLR} = \frac{f(X, Y|H_{1,c})}{f(X, Y|H_{2,c})}$$

We can simplify this LR a bit. Let's say we are given an evidence set $\{X, Y\}$. By definition of conditional probability, we can rewrite the LR for feature-based methods [2]

$$CSFLR = \frac{\mathbb{P}(X, Y|H_1)}{\mathbb{P}(X, Y|H_{2,c})} = \frac{\mathbb{P}(Y|X, H_1)}{\mathbb{P}(Y|X, H_{2,c})} \frac{\mathbb{P}(X|H_1)}{\mathbb{P}(X|H_{2,c})}$$

The second term equals 1 since the distribution of features of the trace does not depend on the assumption of whether or not the trace came from the same source. The denominator in the first term can be rewritten to $\mathbb{P}(Y|H_{2,c})$ because given H_2, the sources are assumed independent of each other so the distribution of $Y|X, H_2$ is just the distribution of $Y|H_2$, which will be approximately equal to $\mathbb{P}(Y)$

We are thus interested in the simplified LR:

$$CSFLR = \frac{\mathbb{P}(Y|X, H_1)}{\mathbb{P}(Y|H_2)} \tag{4.1}$$

In practice, this will boil down to estimating the densities from the data. Bolck et al.[2] describe how this can be done for discrete, univariate, and multivariate continuous data, assuming a two-level model structure from our data. Often the enumerator is close to 1, bar measurement errors. If the traces are truly from the same source, we can expect to measure Y when given X and vice versa. Therefore the LR is approximately proportional to the relative frequency of the measurement in the population.

Parametric approach

One way to get a feature-based model is to use a parametric approach. If we believe that our data has a certain structure we can try to fit existing probability distributions on the data. The parameters for the distributions are estimated from the data. After we have fitted a probability distribution, we can get the densities for the numerator and denominator and get our LR from there. This however is not as general, and real-life data might not follow obvious probability distributions.

Kernel Density Estimation

A more general approach is (multivariate) Kernel Density Estimation, as described by Silverman [18]. This method uses a non-parametric method to estimate the probability density. Using a kernel function $K_{\mathbf{H}}(\mathbf{x})$ with choice bandwidth matrix $\mathbf{H}$ the kernel density estimation of a point $\mathbf{x} \in \mathbb{R}^d$, given data $\mathbf{X_1}, \ldots \mathbf{X_n} \in \mathbb{R}^d$ is given by:

$$\hat{f}_{\mathbf{H}}(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_{\mathbf{H}} \left(\mathbf{x} - \mathbf{X_i} \right)$$

A common choice for the kernel function $K(x)$ is a Gaussian kernel:

$$K_{\mathbf{H}} \left(\mathbf{x} \right) = (2\pi)^{-\frac{d}{2}} |\mathbf{H}|^{-\frac{1}{2}} \exp \left(-\frac{1}{2} \mathbf{x}^T \mathbf{H}^{-1} \mathbf{x} \right)$$

The bandwidth matrix $\mathbf{H}$ serves as the covariance matrix and determines the amount of smoothing. Preferably, one takes the bandwidth to be as small as the data allows, but also here there is a trade-off to make in the bias and variance of your estimate. A larger bandwidth will lead to a smoother estimate but might obscure the underlying structure too much. A small bandwidth could capture this, but taking a small bandwidth might lead to overfitting. A rule of thumb for the bandwidth is Silverman's rule, which gives a bandwidth optimal in mean square error in one dimension. For general dimension d, the optimal bandwidth for feature k is given by

$$h_k^* = \left(\frac{4}{n(d+2)} \right)^{\frac{1}{d+4}} \hat{\sigma_k}, k = 1, \ldots, d$$

Here σ_k denotes the sample standard deviation for feature k. From now on, whenever we write h, we mean the optimal bandwidth h^*. Bolck et al.[2] describe how one can use a Kernel Density Estimator with a Gaussian kernel to compute the likelihood ratio using a Bayesian framework. The numerator in the LR can be seen as the posterior distribution of trace Y, given the distribution of the other trace X assuming they are from the same source. The denominator can be seen as the prior distribution of the trace Y. From there the LR can be computed.

We are given dataset with n sources having r_i measurements. Each source therefore has an estimated source mean,

$$\mathbf{\bar{z}_i} = \frac{1}{r_i} \sum_{j=1}^{r_i} \mathbf{z_{ij}}, 1 \leq j \leq n$$

From this we have a prior distribution for the source means using a KDE [1]:

$$f(\theta) = (2\pi)^{-\frac{d}{2}} |h^2 T_0|^{-\frac{1}{2}} \frac{1}{n} \sum_{i=1}^{n} exp \left(-\frac{1}{2} (\theta - \mathbf{\bar{z}_i})^T (h^2 T_0)^{-1} (\theta - \mathbf{\bar{z}_i}) \right)$$

Here $\mathbf{T_0}$ is the sample between-source covariance matrix estimated from the data. Combining this matrix with the Silverman bandwidth we get our choice for bandwidth matrix $\mathbf{H}$:

$$\begin{aligned} \mathbf{H} &= h^2 \mathbf{T_0} \\ &= \left(\frac{4}{n(d+2)} \right)^{\frac{2}{d+4}} \hat{\Sigma_b} \end{aligned}$$

We then input an unknown trace X and trace Y, each following distribution $N(\theta_x, \Sigma_x)$ and $N(\theta_y, \Sigma_y)$ respectively, where θ_x, θ_y follow from $f(\theta)$. We then compute the posterior distribution and from there our LR of interest which is given by Bolck et al.[2]:

$$LR = \frac{f(\mathbf{Y}|\mathbf{X}, H_1)}{f(\mathbf{Y}|H_2)} = n \frac{|\mathbf{U_{hn}}|^{-\frac{1}{2}} \sum_{i=1}^{n} \exp\{-\frac{1}{2}(\mathbf{X} - \mathbf{\bar{z}}_i)^T (\mathbf{U_{hx}})^{-1} (\mathbf{X} - \mathbf{\bar{z}}_i)\} \exp\{-\frac{1}{2}(\mathbf{Y} - \mu_{\mathbf{hi}})^T (\mathbf{U_{hn}})^{-1} (\mathbf{Y} - \mu_{\mathbf{hi}})\}}{|\mathbf{U_{h0}}|^{-\frac{1}{2}} \sum_{i=1}^{n} \exp\{-\frac{1}{2}(\mathbf{X} - \mathbf{\bar{z}}_i)^T (\mathbf{U_{hx}})^{-1} (\mathbf{X} - \mathbf{\bar{z}}_i)\} \sum_{i=1}^{n} \exp\{-\frac{1}{2}(\mathbf{Y} - \mu_{\mathbf{hi}})^T (\mathbf{U_{h0}})^{-1} (\mathbf{Y} - \mu_{\mathbf{hi}})\}} \tag{4.2}$$

with

$$\mathbf{U_{hx}} = h^2\mathbf{T_0} + \frac{1}{n_x}\Sigma_x$$
$$\mathbf{U_{h0}} = h^2\mathbf{T_0} + \frac{1}{n_y}\Sigma_y$$
$$\mathbf{U_{hn}} = \mathbf{T_{hn}} + \frac{1}{n_y}\Sigma_y$$
$$\mu_{\mathbf{hi}} = h^2\mathbf{T_0}\left(h^2\mathbf{T_0} + \frac{1}{n_x}\Sigma_x\right)^{-1}\mathbf{X} + \frac{1}{n_x}\Sigma_x\left(h^2\mathbf{T_0} + \frac{1}{n_x}\Sigma_x\right)^{-1}\bar{\mathbf{z}}_\mathbf{i}$$
$$\mathbf{T_{hn}} = h^2\mathbf{T_0} - h^2\mathbf{T_0}\left(h^2\mathbf{T_0} + \frac{1}{n_x}\Sigma_x\right)^{-1}h^2\mathbf{T_0}$$

Note that for the computation of the LR, the CSFLR needs to use the within-covariance matrix of trace X and reference Y. These within-covariances are unfortunately unknown and so we need to use an estimate. To make feature-based systems using our simulated data, the lir package written by the NFI was used. This system is by default a CSFLR. The estimate used by this CSFLR is the mean of the within-covariance matrix from all sources, denoted by $\hat{\Sigma}_w$.

We also plug in only one measurement from the trace and reference, so we also have $n_x = n_y = 1$. We also have $\mathbf{H} = h^2\mathbf{T_0}$. Therefore above formulas simplify a bit, since now $\mathbf{U_{hx}} = \mathbf{U_{h0}}$. We get:

$$\mathbf{U_{hx}} = \mathbf{H} + \hat{\Sigma}_w$$
$$\mathbf{U_{h0}} = \mathbf{H} + \hat{\Sigma}_w$$
$$\mathbf{U_{hn}} = \mathbf{T_{hn}} + \hat{\Sigma}_w$$
$$\mu_{\mathbf{hi}} = \mathbf{H}\left(\mathbf{H} + \hat{\Sigma}_w\right)^{-1}\mathbf{X} + \hat{\Sigma}_w\left(\mathbf{H} + \hat{\Sigma}_w\right)^{-1}\hat{\mathbf{z}}_\mathbf{i}$$
$$\mathbf{T_{hn}} = \mathbf{H} - \mathbf{H}\left(\mathbf{H} + \hat{\Sigma}_w\right)^{-1}\mathbf{H}$$

4.3. Specific source pairing for score-based methods

If a feature-based method is not attainable, we resort to score-based methods. We like to make a specific source score-based system, to compute the following LR:

$$\text{SSSLR} = \frac{f(\delta(X,Y)|H_{1,s})}{f(\delta(X,Y)|H_{2,s})}$$

Not that now we have to make a distribution of the scores under H_1 and H_2. Unlike the feature-based approach, we now have to make pairs of our observations. In feature-based, the features can be on their own and we do not require the pairing of measurements from our dataset to make a direct model of our population.

However, in score-based LR systems, computing a score necessitates an input trace and reference measurement for the hypotheses that we stated: $H_{1,s}$: X and Y originate from the same known specific source.

$H_{2,s}$: X and Y originate from two different sources. The source of Y is known. To get an idea of the scores under H_1 for our model, we need to combine the measurements of the specific source with themselves. For scores under H_2, we need to pair measurements from our specific source with measurements from alternative sources to get an idea of what scores under H_2 look like.

There are multiple ways of pairing possible. We however selected the one that can be seen in figure 4.1. In the parts after that, we go into detail of how we make the pairs for the specific source score-based model.

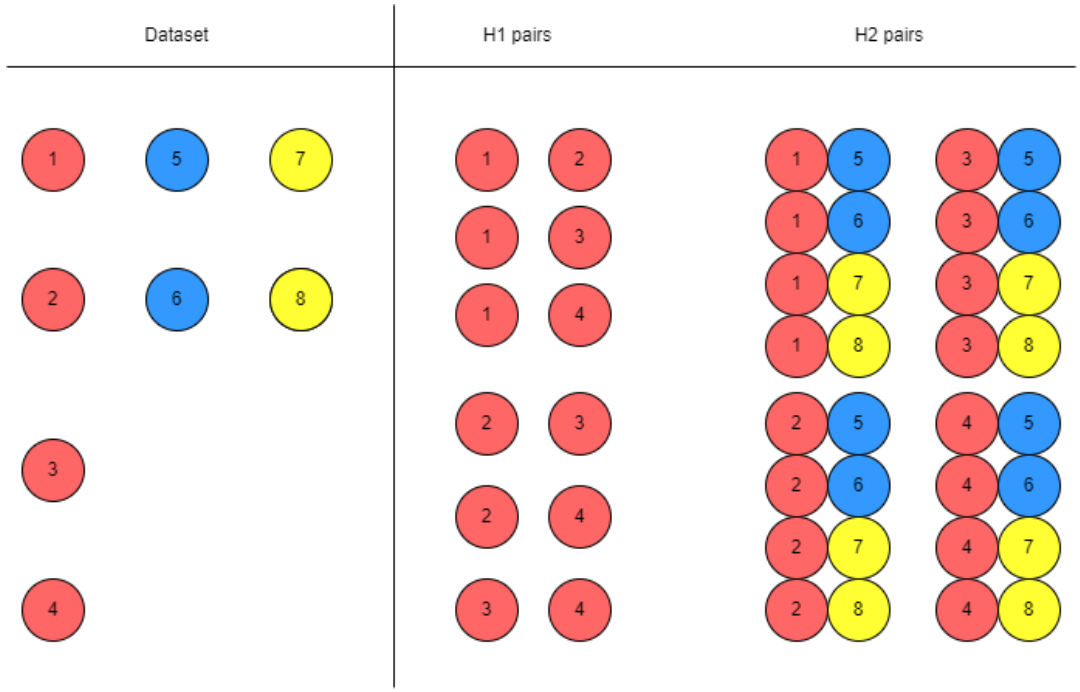


Figure 4.1: This is a small example of how our pairs are made in the specific source setting. We are given a set with 4 repeats of the specific source and two alternative sources each having 2 repeats. We make 6 H_1 pairs and 16 H_2 pairs. All possible pairings are made, and each combination is present only once. Each repeat is equally present in both H_1 and H_2

Pair construction for the prosecution hypothesis

In our dataset, we have r_{ss} repeats of the specific source. For making the H_1-pairs, you only need the measurements from the specific source. A way of making pairs is to take all possible combinations of the repeats, yielding $\binom{r_{ss}}{2} = \frac{r_{ss}(r_{ss}-1)}{2}$ pairs. Other ways to create H_1-pairs could be considered as well. For more discussion on this topic, see section 4.3.1.

Pair construction for the defence hypothesis

For making the H_2-pairs we need to match the specific repeats to the repeats of the alternative sources. Recall that we are given a specific source with r_{ss} repeats, and N_{as} alternative sources each having an equal amount of measurements r. In our simulation, we take the Cartesian product between the set of repeats from the specific source and the set of repeats from the alternative sources, leading to $r_{ss}N_{as}r$ pairs for H_2. Also in this way, every repeat will be included an equal amount of times, precisely r_{ss} times, and so every alternative source is equally represented in the H_2 scenario.

Other ways of making H_2 pairs could be considered, since with this method we can see also some dependencies arising. Let's take an example where we match the first repeat of the specific source s_1 to repeats a_1, a_2 of an alternative source A. Likely a_1 and a_2 will be close as they are measurements from the same source. It follows logically that scores $\delta(s_1, a_1)$ and $\delta(s_1, a_2)$ will be also close to each other. A way to remove this dependency is to match each repeat from the specific source to only one repeat from another alternative source, giving us $r_{ss}N_{as}$ pairs. However, this throws away data and is therefore undesirable.

4.3.1. Dependencies

In our specific source approach, we have some choices to make when we are creating our pairs. In particular, we have dependencies that arise when you make your pairs, which can be done in multiple ways. In this thesis, we have decided to make all possible combinations, both in H_1 and H_2. We discuss the advantages and disadvantages of this way of making pairs.

Dependencies in H_1 pairs

For the H_1 pairs, we have chosen to make all possible combinations for the repeats that are given. This has a couple of advantages.

In the first place it provides the most information possible about our specific source. In this way, we are maximally informed about the possible values that we could observe under H_1.

Secondly, in this way, there is no bias in which pairs are included. For example, if I have 4 repeats $r_1, \ldots, r_4$ of my specific source I could make 2 pairs $(r_1, r_2), (r_3, r_4)$ using two repeats each. There is a priori no reason to use the pair (and then the scores) between r_1 and r_2 over the score between r_1 and r_3 for example. Using all pairs possible prevents selection bias.

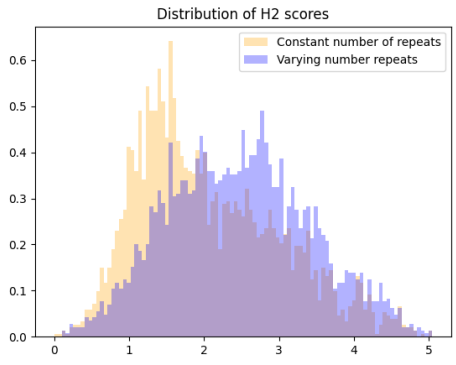
On the other hand, there are also downsides involved. Since combinations scale like $\mathcal{O}(n^2)$, you quickly get more pairs than necessary to get a good model H_1. This might lead to longer computation times. Another weakness of making all combinations is that you lose independence between pairs and their scores. In particular, if I have repeats s_1, s_2 and s_3 from my specific source, and I get low scores $\delta(s_1, s_2)$ and $\delta(s_2, s_3)$,then we will likely also get a low score $\delta(s_1, s_3)$. Losing independence between pairs however might not be as big of a deal here since we are using it to model the H_1-scores of the specific source, where each score is already likely to be close.

There is always a trade-off between dependencies and information, especially when you have too little data. We however are now aware of the risks involved.

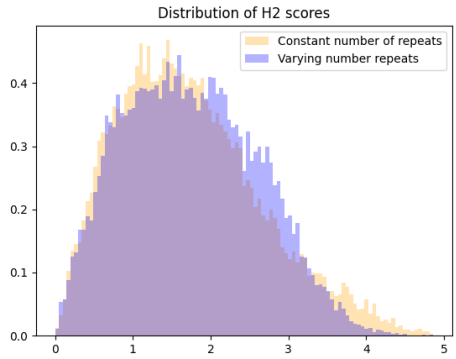
Dependencies in H_2 pairs

For the H_2 pairs, we match every repeat of the specific source to all repeats from every alternative source. In our simulation, we take r_m the same for all alternative sources. In practice, this might not always be the case, and repeats per source may vary depending on your dataset. You might have 2 repeats from one source but have 5 or so from other sources in your dataset. This creates an imbalance in the H_2 pairs you make.

Let's say we have r_{ss} repeats from our specific source S, and we have two alternative sources A and B, each having 2 and 5 repeats respectively in our dataset. By matching S to A, we get $2r_{ss}$ pairs under H_2 involving source A. On the other hand, we get $5r_{ss}$ pairs involving source B under H_2. This leads to an over-representation of source B in the H_2-pairs and gives bias in scores under H_2 since it will falsely believe B is more present in our population relative to other sources.



(a) Distributions of the two scenarios in a small dataset



(b) Distributions of the two scenarios in a larger dataset. It can be seen that the influence of sources with larger amounts is less prevalent

Figure 4.2: Effects of having a varying amount of repeats per source can have some consequences on the H_2 distribution of the scores, especially in cases where you have a small dataset

This can especially be a problem in smaller datasets, where the relative over-representation is more

present. In figures 4.2a and 4.2b we see the effects. For a dataset containing 20 sources, we plot the score distributions in the case where each source has a constant 5 measurements and a case in which each source has a random amount $r_i \in \{2, 5, 10\}$. In the figure, we see how the distributions can differ a lot for these scenarios. When you have a larger dataset these influences are less noticeable.

As long as we have an equal amount of repeats per alternative source in our dataset, we ensure that no alternative source is over-represented under H_2. With simulation, we can exactly decide the number of repeats for each source. In practice, a way to circumvent this issue is to reduce the number of repeats from sources that are over-represented in the data set to ensure each source is present equally in the H_2 scenario.

4.4. Common source pairing for score-based methods

In the common source approach, we aim to answer if we can relate two unknown traces X and Y to the same unknown source. Now we are after the following LR:

$$\text{CSSLR} = \frac{f(\delta(X,Y)|H_{1,c})}{f(\delta(X,Y)|H_{2,c})}$$

We restate the hypotheses associated with a common source approach:

$H_{1,c}$: X and Y originate from the same unknown source.
$H_{2,c}$: X and Y originate from two different unknown sources.

Again we need to make our pairs according to this approach to get an idea of how scores look like under H_1 and H_2 However contrary to the specific source approach, we can drop the restriction that in all pairs there needs to be a measurement from the specific source. An overview of the pairing can be seen in figure 4.3.

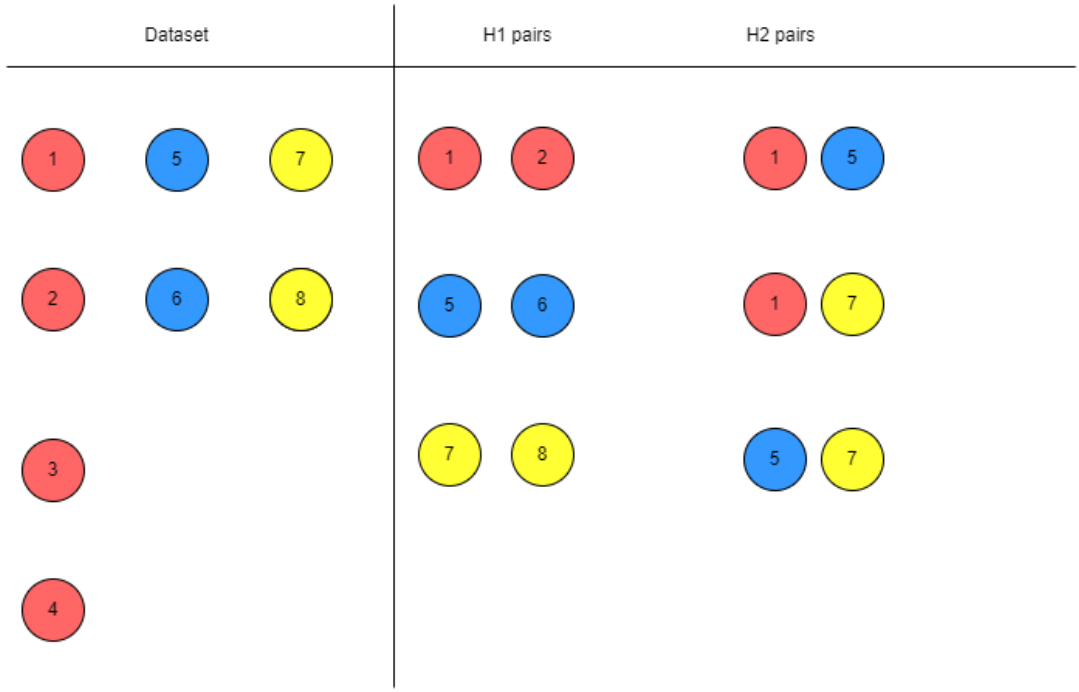


Figure 4.3: In this picture a small example can be seen of how our pairs are made in the common source setting. Like the example before, we are given a dataset with 4 repeats of the specific source and two alternative sources each having 2 repeats. For the H_1 pairs we only take the first two repeats of the specific source to prevent imbalance in the pairs. This gives us $3\binom{2}{2} = 3$ pairs. For the H_2 pairs, we take the first repeat of each source and make all possible combinations between sources, giving us $\binom{3}{2}$ = 3 pairs as well. Notice how in this case it gives us an equal amount of pairs, but the way of getting those numbers is fundamentally different, and the amount of H_1 and H_2 pairs in general can differ vastly.

Pair construction for the prosecution hypothesis

For making the H_1 pairs for a common source approach we proceed differently than for a specific source approach. In our simulated dataset, we could have an imbalance in the amount of repeats we have per source if we consider the scenarios where we have more repeats from our specific source than from our alternative sources.

This was no problem for the specific source scenario, as we were able to make pairs just fine without having an imbalance in how often each repeat was included. However, in the common source approach, this imbalance is more prevalent.

If we want to make H_1 pairs in the common source scenario, we have to match repeats from each source to themselves to model the H_1 distribution from the whole population, not just the specific source. We also want to research the cases where we have a dataset where we have more repeats from our specific source, more than others. But if we create all possible combinations now using all repeats from our specific source this will create an imbalance. This imbalance is better demonstrated by an example.

For example, let's say we have 50 repeats from our specific source in our dataset and 2 repeats from other 100 sources. Making all combinations, we would be able to make $\binom{50}{2} = 1225$ pairs using the specific source, and 100 $\binom{2}{2} = 100$ pairs using all other sources, giving us a total of 1325 pairs. However, it is immediately clear that of these 1325 pairs, 1225 are pairs created using one source only. This is biased a lot towards the specific source, and undesirable if we want to make a good model of the whole population which is the aim of the common source model.

A workaround is to cut the amount of repeats from the specific source down to the amount that is more in line with the alternative sources to prevent this bias. If in our dataset we have only 2 repeats per alternative source, then to combat the bias we should also only take 2 repeats from the specific source to make an H_1 pair. Or in the case where we have 5 repeats per alternative source, we can decide to also use this amount of repeats from our specific source.

In general, given a dataset consisting of N distinct sources each having r repeats, we are able to make $N \cdot \binom{r}{2}$ pairs for the H_1 scenario.

Pair construction for the defence hypothesis

For making the H_2 pairs we only take the first repeat from each source including the specific source, and make combinations between all sources. This gives us $\binom{N}{2}$ pairs for the H_2 scenario.

The reason to take only the first repeat despite having more options is that we have more than enough information about the H_2 distribution using only the first repeat. For example, if we have a dataset with 100 sources with at least 2 repeats each, we can already make $\binom{100}{2} = 4950$ pairs for H_2. This is a large number of (conditionally) independent pairs to model our H_2 population. However, let's say we make all possible combinations between two sources each having r repeats. We would make r^2 pairs between them, giving us an even larger amount of H_2 pairs. In practice, this is often unnecessary.

In general, given a dataset consisting of N distinct sources each having r repeats, we are able to make $\binom{N}{2}$ pairs for the H_2 scenario, independent of the number of repeats per source.

4.4.1. Dependencies

Also in our common source approach, we have some choices when creating our pairs.

Dependencies in H_1 pairs

Analogous reasoning is followed here for the specific source H_1 approach, but now that logic is repeated for each source. Within each source, there is a maximal information gain by taking all possible combinations, while using each repeat an equal amount. The only problem would arise if a source is

over-represented in the dataset. As we discussed, we can remove this by decreasing the amount of repeats for that source to be more in line with the rest of the dataset.

Dependencies in H_2 pairs

Since we only take one repeat per source and relate it to the first repeat from other independent sources, the dependencies are reduced to a minimum. We have in this case conditional independence between the pairs: I have first repeats a_1, b_1, c_1 from sources A, B, and C, I have (among others) the pairs (a_1, c_1) and (b_1, c_1), both including c_1 and are therefore not independent. However, a_1 and b_1 are from different sources, and are independent. In particular, even given the value of c_1, the joint distribution of a_1 and b_1 is unaffected by c_1 and fits the definition of conditional independence.

4.5. Computing scores

Recall with a score-based approach we meant that we turn to LR systems in which we look at the distribution of a function of the features. This function is the *score* function and is usually denoted as $\delta(x, y)$, where x and y are the input feature vectors. Depending on the approach, y can either be the known reference in a specific source case, or y can be a trace measurement. There are a couple of choices, mainly split up into two categories: (Dis)similarity scores or statistical models.

Dissimilarity scores

Dissimilarity or similarity scores capture how (dis)similar two input feature vectors are. Depending on which function you take, a high score corresponds to high similarity or low similarity. An advantage of using a similarity score is that you don't need to train a model to output a score, and as such these score functions are also applicable in cases where you do not have a lot of data. For similarity scores, one has for example Pearson correlation and cosine similarity. [8] On the other hand you have dissimilarity scores where a score of 0 corresponds to perfect similarity. Common choices are the L_p-norms as score functions:

$$\delta_p(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|^p$$

In this thesis, the L_1-norm is considered as a dissimilarity score. Multiple scorers could be considered but this is out of the scope of this thesis. The benefit of using the L_1-norm is that the score remains interpretable. A large score corresponds to a large distance between measurements. Note that using the L_1-norm as a scorer does not account for rarity. As mentioned before, it is possible that two different measurement pairs $(\mathbf{x_1}, \mathbf{y_1})$ and $(\mathbf{x_2}, \mathbf{y_2})$ lead to the same score even though the measurements themselves may be more common or less common.

Statistical models

Another way to get a similarity score is to use statistical models, such as those from the machine learning literature. The model parameters are trained such that it produces high scores for H_1-true pairs and produces low scores for H_2-true pairs.

As long as the output is a single number, any model can be used in principle [8]. Different machine learning methods might work better on other types of data. An example of a model that can be used for this purpose is Support Vector Machine, which is (among others) applied for gunshot residue [9]. In practice, many machine learning methods could be tested, from which the best-performing method may be selected.

4.6. Calibration of scores

Since we are not interested in the scores themselves, we have to find a way to go from scores to probabilities and likelihood ratios. Going to probabilities from scores is called calibration and is an important step for likelihood ratio systems. A well-calibrated likelihood ratio system puts out LRs that reflect the true underlying probabilities. A classical example is a weather forecaster that gives a forecast of rain with a probability of 0.9. If in around nine out of ten days when he predicts rain it rains, we can speak of a "well-calibrated" forecaster. In other words, the probabilities should reflect real-life empirical observations. In this section, we will go over how we can go from scores to probabilities.

After a score has been assigned to each pair both the H_1 and H_2, have to go from score to a corresponding probability for observing that score. Also here there are some choices, like choosing between again a (one-dimensional) Kernel Density Estimation on the distributions of the scores or fitting a function that relates the score to the posterior probability that the comparison is H_1.

When using the KDE approach, we take the scores found under H_1 and H_2. The KDE gives two fitted densities for the scores, $s_{H_1}(\delta)$ and $s_{H_2}(\delta)$. The score-based likelihood ratio that you end up with can be computed as follows:

$$LR(X,Y) = \frac{s_{H_1}(\delta(X,Y))}{s_{H_2}(\delta(X,Y))}$$

Another option that you can use is to find a mapping from scores to odds, for example by using a logistic regressor or isotonic regression. We however stick to Kernel Density Estimation as our calibrator of choice, as this is not the scope of this thesis.

4.7. Bounding Likelihood Ratios

In the section 2.6 we talked about how discrimination is a desirable property of LR systems. However, a problem arises when we have systems that separate too well. Let's say we have a case where the densities under H_1 and H_2 are well-separated and we observe a new data value that is very common under H_1. Since the densities separate well, by extended reasoning this data value will be less common under H_2, and this data value will likely be located in the tail of the density under H_2. However, since we are now in the tail of the density, there is by definition less data present. This can lead to uncertainty in the exact value of the LR, and in some cases could give LRs that differ by several orders of magnitude from the true LR.

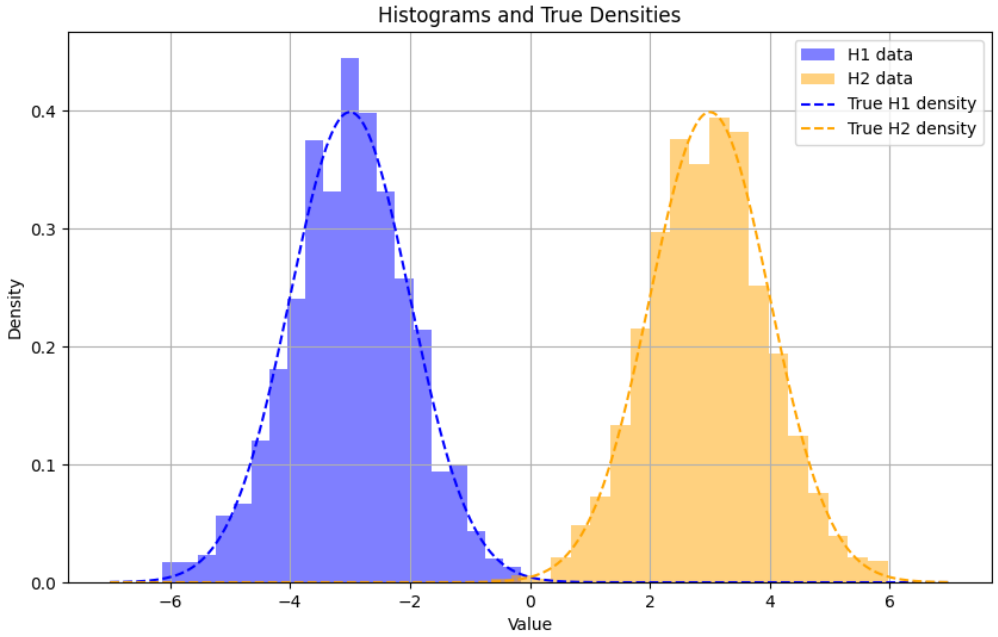


Figure 4.4: Surrogate example of a situation in which the H_1-true and H_2-true densities separate well

If we recall our example for well-separated densities given in section 2.6, had the densities seen in figure 4.4 The computed LR from the LR system based on a KDE yields an LR way bigger than the true LR if you knew the true densities. In this case using the KDE as a calibrator outputted an LR of the order 10^{33}, while the true LR was in the order of 10^{7}.

The problem is that LRs outputted by the system are sensitive to how the tail for both H_1 and H_2 densi-

ties is modeled. If we do not combat this problem there could be serious errors in casework. A solution to this problem is a fully Bayesian method, proposed by Brümmer [5] and Morrison [12]. These are preferred to frequentist methods since frequentist models often concern themselves with parameter estimation which is less effective in cases with little data.

A particular solution to this problem is presented by Vergeer et al.(2016) [22], called the ELUB (Empirical Upper and Lower Bound) method, where the LR outputted by the system is bounded based on the size of the data sets used. These bounds protect us from extreme LRs in cases where there is little data in the tails present. Intuitively, a dataset of 100 samples should not support an LR that is a million or higher. Bounding the LRs is necessary since it prevents the risk of producing too extreme LRs in court cases since it might lead to a wrongful indictment of a suspect.

We proceed to include ELUB bounders in the construction of our LR system since this is a desirable property for LR systems to have. Intuitively, an LR system should be less confident or more confident about the LRs that it outputs based on the amount of data is has seen. The broader the range of the bounds, the better. We do keep in mind that this could come into play in the performances of our LR systems.

5

Evaluating Likelihood Ratio Systems

After you have built your LR system based on your training data, we need to evaluate the quality of the LR system. Using the validation set we have split off before, we can assess the quality of the LR system that we have built by testing it on the validation data that the LR system has not seen before.

5.1. Validation pairs

We construct the H_1 and H_2 pairs to evaluate from our validation data. We have some measurements from the specific source and alternative sources. For validation, we need to pair our measurements such that our score-based methods can evaluate them. Recall that we have two choices for pairing; the common source pairing and the specific-source pairing. But for our comparison, it is only sensible to make validation pairs using the specific source pairing.

In a trace-reference problem, we have two options for our trace; either our trace originates from the reference, or not. But we do not doubt the origins of our measurement from the reference, that one is always fixed. We therefore can only plug in pairs in which at least one measurement originates from the specific source. This corresponds with a specific source pairing approach using data from our validation set.

This also makes sense considering the following; if we would make our pairs using common source pairing, there would be at least one pair of H_1 measurements that originate from alternative sources and not the specific source. But since we want to validate our likelihood ratio systems on the same data, we would also plug this pair into a specific source likelihood ratio system. This is however not made for that.

For a feature-based method, we did not need to pair to make the model, but for validating a feature-based model we also need to input measurements from our specific source and alternative sources. We can however take the part of the pair that belongs to the specific source as our reference measurement, and the other part as our trace measurement.

5.2. Strictly proper scoring rules

The judicial system is not interested in likelihood ratios for their own sake, but in how much they improve decisions based on posterior probabilities (the judge's verdict). The benefit of LR systems can therefore be measured in how they improve posterior probability distributions compared to prior probability distributions[21]. One way of evaluating is using a Strictly Proper Scoring Rule, or SPSR for short. The evaluation of the posterior distribution will tell us the benefit of using the LR system to update the prior as opposed to not using the LR system. If the prior that you are considering is an uninformative prior, i.e. $\mathbb{P}(H_1) = \mathbb{P}(H_2)$ any improvement that follows from the LR system will reduce some uncertainty present. [4]. In the case of an uninformative prior, the posterior probability equals the likelihood ratio.

A common method that is used as an SPSR for LR systems is the Empirical Cross-Entropy (ECE) [15]. This is a measure that indicates better performance when the likelihood ratio leads to the correct

decision [10]. The ECE is an interpretable and general performance metric for the LR system in situations where the forensic evaluator makes no decision and in which the value of the prior can differ over cases. However, as Meuwly [10] states, ECE is more useful to gain insight into what happens when you consider a range of prior probabilities. Usually, the prior probabilities are unknown to the forensic scientist.

5.2.1. Log-likelihood-ratio cost

A special case however is to use the ECE when you consider prior odds equal to 1, which is equivalent to assuming that H_1 and H_2 are equally probable. This yields the log-likelihood-ratio cost, C_{llr}, and is defined as [4]:

$$C_{llr} = \frac{1}{2}\Big(\frac{1}{N_{ss}}\sum_{i=1}^{N_{ss}} \log_2(1+\frac{1}{LR_{ss}}) + \frac{1}{N_{ds}}\sum_{j=1}^{N_{ds}} \log_2(1+LR_{ds})\Big)$$

Here, N_{ss} are the number of H_1 pairs you can make from the validation set, and N_{ds} are the number of H_2 pairs. The C_{llr} is an average of two underlying averages. The first sum is the average over all the H_1 true pairs in our validation set. For each of these pairs, we have a score and therefore an LR which should be greater than 1 since these are paired measurements from the same source. As a consequence the $log_2(1+\frac{1}{LR_{ss}}$ is now below 1. The inverse logic applies for the second sum but for H_2 pairs.

A (nearly) perfect LR system will have a C_{llr} very close to zero since it will predict a large LR for H_1-true pairs and a small LR for H_2-true pairs. An LR system that does not add information will have a C_{llr} of 1. A system with a C_{llr} more than 1 means the system is providing misleading information if the LR system is well-calibrated. If there is a system with C_{llr} greater than one there must have been at least one case in the validation set where the LR system outputted an LR lower or greater than one for H_1 or H_2 cases respectively. Indeed, if the LR system outputs a high LR for a H_1-pair, $\log_2(1+\frac{1}{LR_{ss}})$ will be small and result in a lower C_{llr} . Conversely, a low LR for a H_1 pair results in a higher C_{llr} . The same logic but inverse also applies for H_2 pairs.

Other measures for evaluation are possible but the C_{llr} is a one-size-fits-all measure to compare likelihood ratio systems. Since we are more interested in comparing LR systems to each other we continue using the C_{llr} as our measure of choice.

6

Demonstrative use case on ecstasy pills

In this chapter, we aim to make the previous procedure more concrete by applying it to a real-life dataset containing information about MDMA tablets, also known as ecstasy. Based on this dataset we construct a CSSLR, a SSSLR, and a CSFLR. This chapter serves as an example to explain the underlying choices present when constructing likelihood ratio systems. The dataset can be found at `https://github.com/NetherlandsForensicInstitute/lr-benchmark/blob/dev_asr/resources/drugs_xtc/xtc_data.csv`

6.1. Data synopsis

The data consists of batches of MDMA tablets. Each batch is identified by a batch number. There are 160 different batches. The batches correspond to the sources. Each batch has several repeat measurements, which are 2,4,5 or 6 repeat measurements. The features considered are diameter, thickness, weight, and purity.

Some data impurities are present. For example, batch 53 measurement 6 has a purity of 0 while all other measurements have a purity of about 50. Furthermore, batch 79 has a very high within-source standard deviation for purity compared to others. The first 4 measurements have a purity of around 38 while measurements 5 and 6 are significantly higher at 53 and 52 respectively. Lastly, batch 113 has two measurements where the first one has a thickness of 3.82 while the other measurement has a thickness of 8.87. For all these measurements it could be decided to include or exclude the data. In this example only the measurement having purity 0 is dropped and batches 79 and 113 are included. It is however clear that removal of outliers is beneficial for your LR system.

In table 6.1 an overview can be seen of the estimated between and within standard deviations. It can be seen that due to including batch 113 we get a the maximum standard within deviation exceeds the between standard deviation.

	std_between	std_within_mean	std_within_max
Diameter	0.801863	0.013587	0.096003
Thickness	0.750069	0.104882	3.570889
Weight	62.999614	5.940892	40.897025
Purity	10.867344	0.708953	7.521081

Figure 6.1: Overview standard deviations data set

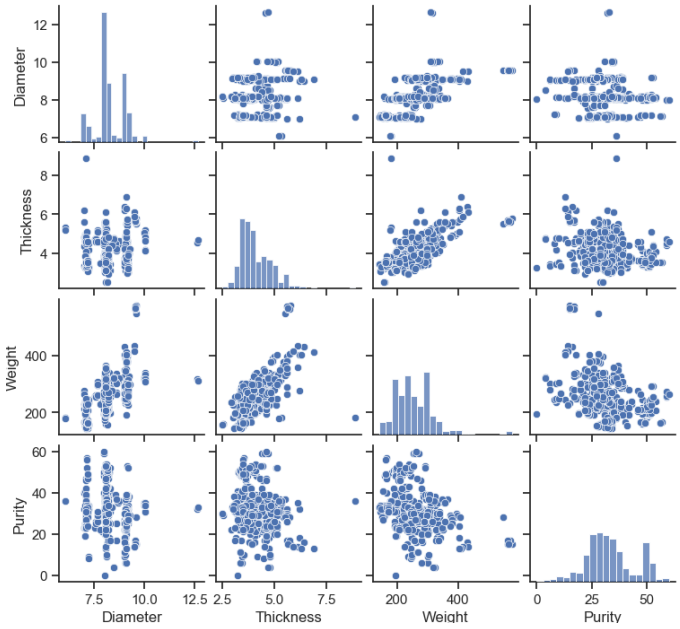


Figure 6.2: Correlations

	Diameter	Weight	Thickness	Purity
Diameter	1.00	0.62	0.098	-0.23
Weight	0.62	1.00	0.65	-0.46
Thickness	0.098	0.65	1.00	-0.30
Purity	-0.23	-0.46	-0.30	1.000

Table 6.1: Table of correlations between the features

Table 6.1 shows that weight is correlated with diameter and width, which makes sense. A decision therefore could be to drop the weight parameter. Here we do not do that since we already have a small amount of features and a comparatively small dataset and this chapter is for demonstrative purposes only.

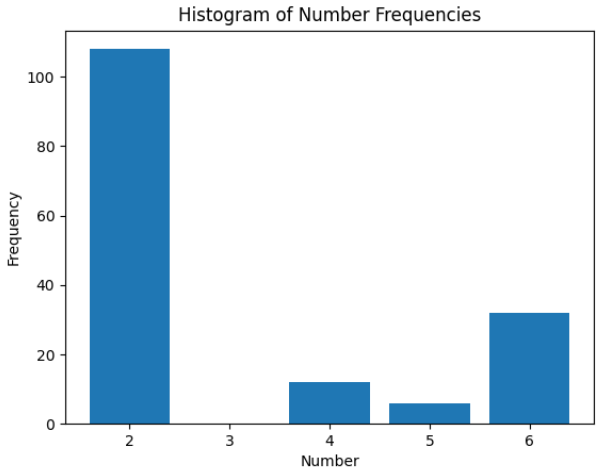


Figure 6.3: Histogram of the frequency of the amount of repeats present in our dataset. 109 sources have 2 repeats, 12 sources have 4 repeats, 6 sources have 5 repeats, and 33 sources with 6 repeats

6.2. Data preparation

As described, we split off a validation set that contains 20% of the batches. In this example case we make a specific source model for the batch identified by ID 49 since it is the first batch to have at least 5 measurements required to make a specific source model. Batch 49 has 6 measurements. The theoretical reason we require at least 5 measurements to make a (small) specific source model is so that we can put 3 measurements in our training set and 2 in the validation set.

We could of course put all measurements in our training set and no measurements in our validation set but then we would have no way of validating an H_1 true pair. By splitting into 3/2 we will ensure we get at least one H_1 pair in our validation set and give us the minimum to compute a C_{llr} for a validation.

We therefore a source that has at least 5 measurements, and batch 49 satisfies the needs as it has 6 measurements. After the split, we end up with a training set containing 4 measurements of our specific batch 49 as well as additional 394 measurements from other sources. In our validation set, we have 2 specific source measurements and an additional 94 measurements from other sources. Keep in mind that some other sources might have more repeats than others. This could affect the C_{llr} . In this example, we do not alter the amount of repeats from over-represented sources to be more in line with the others. The amount of repeats per source only matters in our construction in the amount of H_1-pairs for CSSLR, where we took all possible pairings using those measurements.

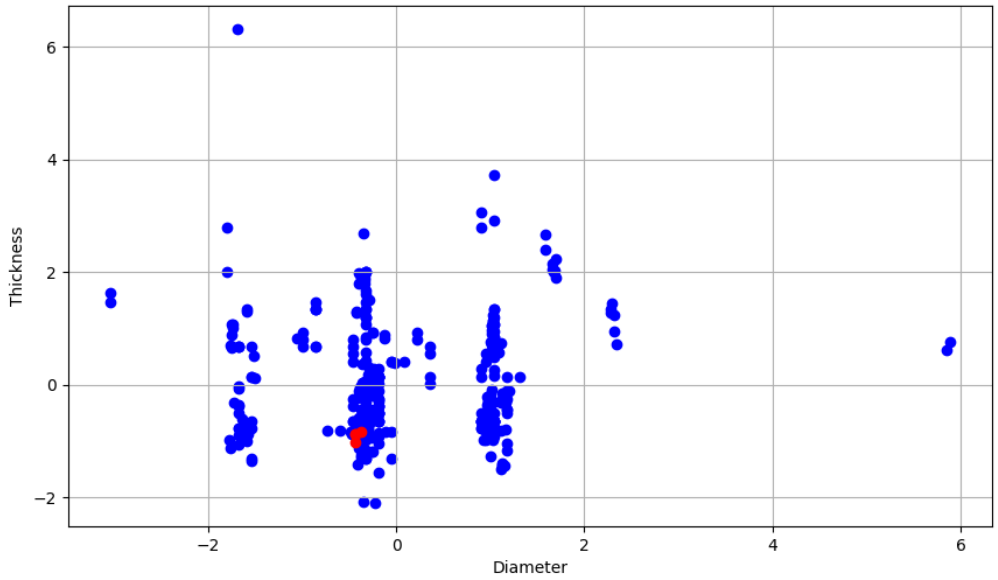


Figure 6.4: Data visualization of the training set, with in red batch 49 within the preprocessed data set, using diameter and thickness as features

6.3. Making pairs

Based on these numbers we can compute the amount of pairs we make using our training set. The pairing approach we took was the one described in sections 4.4 for CSSLR pairing and 4.3 for SSSLR pairing. For the CSFLR we do not have pairing in the train set, but the pairs we validate in the validation set are the same as for the other two score-based systems. Since we are interested in the effects of using a common source model applied to a specific source, we apply the specific source pairing on our validation set for our CSSLR as well. In table 6.2 it can be seen how many pairs are made.

	H_1 train	H_2 train	H_1 validation	H_2 pairs validation
SSSLR	6	1576	1	189
CSSLR	601	8128	1	189

Table 6.2: Amount of pairs in each set for each hypothesis

6.4. Computing LRs

CSFLR

For the CSFLR we proceed as described. Based on the preprocessed data, the Silverman bandwidth equals $h^* = 0.518$, so $(h^*)^2 = 0.268$, and we have an estimated between covariance matrix:

$$\hat{\Sigma_b} = \begin{bmatrix} 1.00895166 & 0.09128142 & 0.59646645 & -0.24000784 \\ 0.09128142 & 0.87275091 & 0.63946352 & -0.28643703 \\ 0.59646645 & 0.63946352 & 0.989065 & -0.4721239 \\ -0.24000784 & -0.28643703 & -0.4721239 & 0.98995146 \end{bmatrix}$$

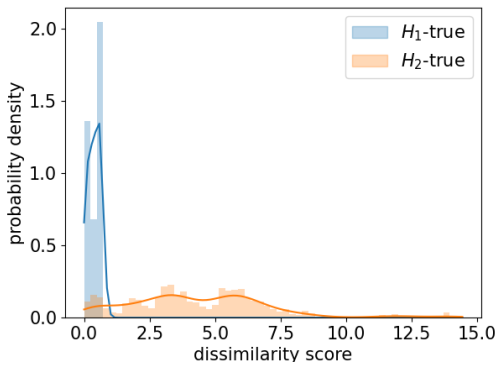
This together yields the bandwidth matrix **H**:

$$\mathbf{H} = (h^*)^2\hat{\Sigma_b} = \begin{bmatrix} 0.27104728 & 0.02452207 & 0.16023623 & -0.0644763 \\ 0.02452207 & 0.23445797 & 0.17178707 & -0.07694915 \\ 0.16023623 & 0.17178707 & 0.26570488 & -0.12683254 \\ -0.0644763 & -0.07694915 & -0.12683254 & 0.26594302 \end{bmatrix}$$

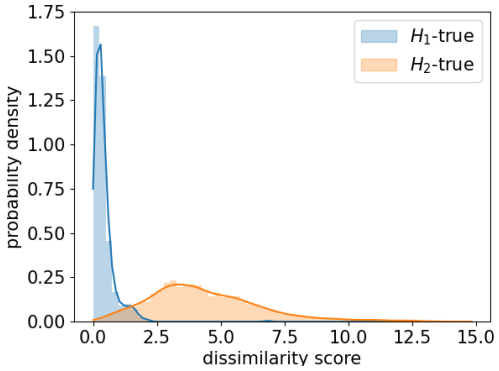
Using this bandwidth matrix we fit a KDE on the data, and combined with formula 4.2 we compute an LR for the pairs in our validation set.

CSSLR and SSSLR

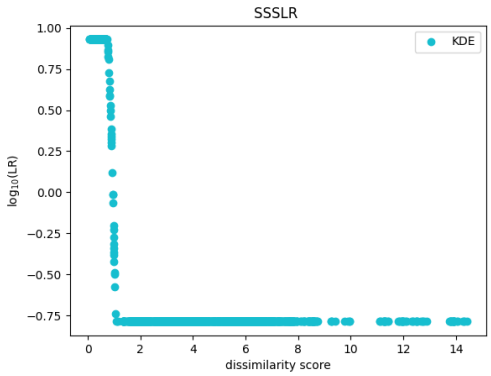
For each pair in the training sets a score is computed using an $L1$-scorer and a KDE is fitted on these scores. In the figures below is seen what the distributions look like.



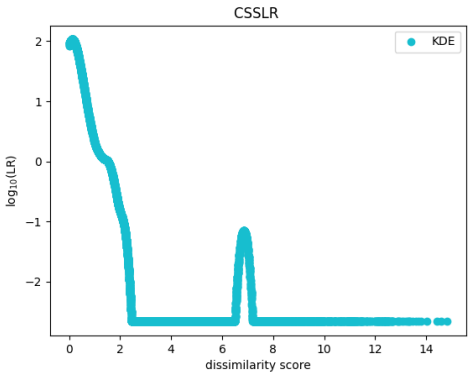
(a) Score distribution and KDE fitted on the scores for the SSSLR using the training data



(b) Score distribution and KDE fitted on the scores for the CSSLR using the training data



(a) Scatterplot of the scores and corresponding $\log_{10}$ LR for the SSSLR method



(b) Scatterplot of the scores and corresponding $\log_{10}$ LR for the CSSLR method

In figure 6.6a and 6.6b it can be seen how the LRs decrease as the score increases for both score models. A higher score should mean more dissimilar inputs. Therefore, it is desirable if the LRs decrease as the score increases. We see that this happens for the SSSLR. As the score increases the log(LR) quickly decreases to the lowest possible value allowed by the ELUB bounds.

Something weird however happens with the CSSLR plot where the score is around 7. We get an increase in log(LR) even though the score increases. This is likely due to a source that has an outlier as a measurement. It leads to a large score even though it corresponds to an H_1 pair. Fitting a KDE for H_1 scores will then have (although) a small peak at that score. This will lead to a bump in H_1 density leading to a higher LR.

Indeed, if we take the maximum H_1 score for the CSSLR model, it is 6.84. It is likely due to including batch 113 which had a large within standard deviation for thickness. The second largest score is significantly smaller at 2.12. This explains the random increase in LR as the score increases, where the opposite should be expected.

6.5. Compare CLLRs

Based on the methods above we end up with the following C_{llr} for each method:

Model	C_{llr}
SSSLR	0.3508
CSSLR	0.3590
CSFLR	0.4806

The SSSLR is the best-performing LR system for this batch, based on this split, scorer, and calibrator. It is clear that using a different split yields a different LR system. One could also apply k-fold cross-validation to verify if this is also the best choice for other splits.

7

Performances for identical within variability

In this chapter, we will consider a situation where all sources in our population have the same variability. This will serve as a starting point and we will expand on this in different ways in chapter 8 where we consider different within-variability and chapter 9 where we consider high-dimensional LR systems.

Simulation setup

We will simulate two-dimensional data from N sources and r_{ss} specific source measurements, with a constant of $r = 5$ repeat measurements per source. We take $r_{ss} \in \{10, 50\}$ to represent little or sufficient information about our specific source and take $N \in \{20, 100\}$ to represent a small or large background dataset, yielding four different configurations.

After determining the sizes of our dataset, we can simulate our measurements. Recall the two-level model defined in 3.1. We now have the scenario in which we have equal within covariance matrices $\Sigma_{w,i}$ and $\Sigma_{w,j}$ for all sources $i, j = 1, \ldots, N$. Therefore, we can speak of a general within covariance matrix Σ_w in this section. The within-covariance matrix is also a scalar multiple of the identity matrix, so $\Sigma_w = \sigma_w^2 I$.

We will compare different difficulties of the underlying dataset by increasing the within-standard deviation scalar. More precisely, we take $\sigma_w \in \{1, 25, 50, 100\}$ to represent the low, medium, high, and highest variability. The terms ”low” and ”high” are relative to the between-source variability, which we keep at a constant $\sigma_b = 100$. We assumed before that we had independent features. This would give a between covariance matrix that equals $\Sigma_b = \sigma_b^2 I$.

A higher within standard deviation σ_w corresponds with more uncertainty surrounding the mean of our source. This would make it more difficult to tell which measurement belongs to which source, so we expect worse performance if this uncertainty becomes too high. We then generate our data in the following way:

Data generation procedure

1. Specify $\mu, \Sigma_w, \Sigma_b, n, r, r_{ss}, d$
2. Generate n d-dimensional source means according to $MVN(\mu, \Sigma_b)$ where f is the distribution of the population source means
3. Use the first generated source mean as μ_{ss} and generate r_{ss} measurements for the specific source using $MVN(\mu_{ss}, \Sigma_w)$
4. Use the other $n - 1$ alternative source means and generate r measurements for the each of the alternative sources using $MVN(\mu_i, \Sigma_w)$

After generating the dataset, we can make our LR systems. However, to remove random error from just simulating one dataset, we repeat a 100 times and compute for each dataset the corresponding

C_{llr} per LR system. The differences in C_{llr} between the methods will be our measure of performance to decide if a model performs better or worse than another.

Since we use an L_1-scorer and simulate independent features that are identically distributed, we can use this to try to get an explicit formula for the scores under H_1 and H_2, since we have that the absolute differences give a folded normal distribution [19]. An explicit derivation is found in the appendix A. Here we write down the expected values of the H_1 scores

$$\mathbb{E}[\delta(X,Y)|H_{1,c}] = d \cdot \frac{2\sigma_w}{\sqrt{\pi}}$$
$$\mathbb{E}[\delta(X,Y)|H_{1,s}] = d \cdot \frac{2\sigma_w}{\sqrt{\pi}}$$

From the derivation, we can see a consequence of assuming identical sources. We have the same mean of the scores under H_1 for the SSSLR and CSSLR since the models have the same within variability. Another key thing to note from the derivation is that for a specific source scenario, the H_2 scores will depend on the mean of the specific source.

$$\mathbb{E}[\delta(X,Y)|H_{2,s}] = \sum_{i=1}^{d} \left(\sqrt{\frac{2(\sigma_b^2 + 2\sigma_w^2)}{\pi}} e^{-\frac{-\mu_{ss,i}^2}{2(\sigma_b^2+2\sigma_w^2)}} + \mu_{ss,i} \left[1 - 2\Phi\left(-\frac{\mu_{ss,i}}{2(\sigma_b^2 + 2\sigma_w^2)} \right) \right] \right)$$

Here $\Phi(x)$ denotes the cumulative distribution function of a standard normal random variable. The formula stems from using properties of the folded normal distribution. It captures the notion that if a specific source is further away from the population mean, on average the distance between an alternative source and the specific source will be larger, while the H_1 scores remain unaffected. This could lead to better separation of the scores and therefore a better-performing LR system.

7.1. Little specific source information

First, we consider the scenario with fewer measurements from our specific source with a small or large background population. We expect the SSSLR to perform poorly compared to other models, as it has less information. We will also vary the size of the background population.

7.1.1. Small background population

In this scenario, we have a dataset containing little data from our specific source, but now a larger background population, In particular, we again take $r_{ss} = 10$, but now we have $N = 100$ sources.

By increasing the sample size but not the amount of data from our specific source, we expect especially the CSSLR and feature-based model CSFLR to increase performance when compared to the case where you have a small background population since these are the models that should directly benefit from having more background data. SSSLR does benefit as well but only in the amount of H_2 pairs it could make.

Small background population with low variability

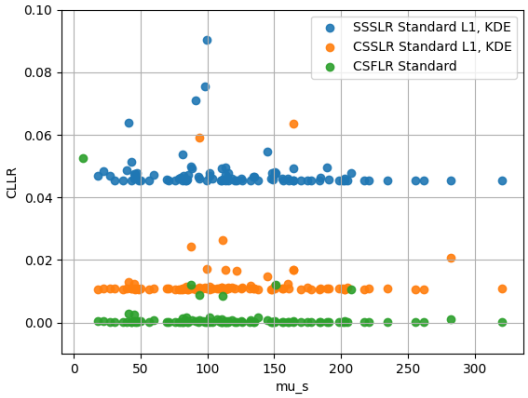


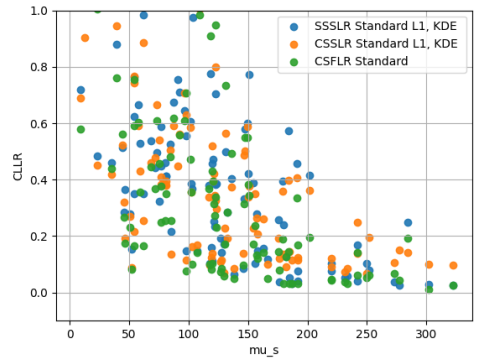
Figure 7.1: Scatterplot of CLLRs for the three likelihood ratio systems

In figure 7.1 we see how the systems perform for the case with $\sigma_w = 1$. It can be seen that all systems have a very low C_{llr} already. Simulating a specific source further from the center does not affect the performance, as this is overshadowed by the fact that the data separates well.

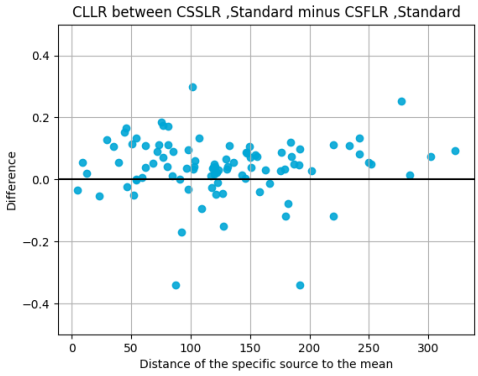
In this case, the SSSLR model performs relatively the worst of these three models. The CSSLR model performs slightly better. In this scenario, a CSSLR can outperform an SSSLR. The feature-based model CSFLR performs the best and has a near-perfect performance.

It is likely that, for the score-based models, the ELUB bounds are attained. However, the CSSLR model has wider ELUB bounds than the SSSLR due to having access to more data. Having wider ELUB bounds allows the CSSLR to produce higher and lower LRs in the H_1 and the H_2 scenario respectively, which in return means that the CSSLR can produce a lower C_{llr} .

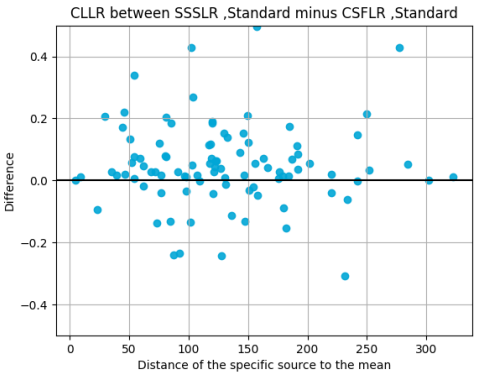
Small background population with medium variability



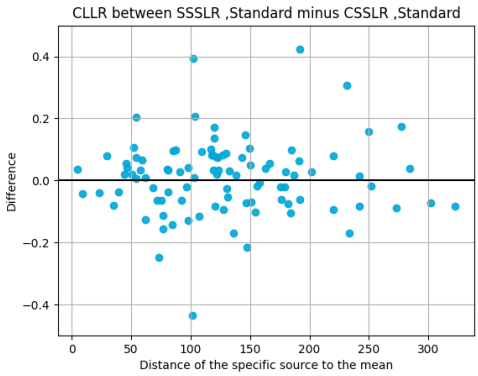
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR



(d) difference between SSSLR and CSSLR

Figure 7.2: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 25$

Now we increase the within-variability. In figure 7.2 we can see that this has a big effect on the performances of our LR systems, they all perform way worse than for $\sigma_w = 1$. We also see a trend that as our specific source moves away from the center of our population all models generally perform better than when our specific source is more centered, even the specific source model. The C_{llr} seems to decrease stochastically monotone as the distance of the specific source increases.

Furthermore, the feature-based method seems to perform the best. It at least outperforms the CSSLR: We see that the difference in C_{llr} is generally positive, which means that the C_{llr} of the CSSLR is on average higher so the CSSLR performs worse. The same holds for the difference between the SSSLR and CSFLR, although here there seems to be a bit more spread in the differences. It does not matter if you use a CSSLR or SSSLR performance-wise as the difference is centered around 0. This would argue in favor of the CSSLR as this can be re-used over cases.

The analysis was also done for higher variabilities where $\sigma_w = 50, 100$. These figures can be found in the appendix, in figures B.1 and B.2 respectively. However, the same conclusions follow for these difficulties for a medium within-variability, the only difference is the worse performance in C_{llr} .

7.1.2. Large background population

In this scenario, we have a dataset containing little data from our specific source, but now a larger background population, In particular, we again take $r_{ss} = 10$, but now we have $N = 100$ sources.

By increasing the sample size but not the amount of data from our specific source, we expect especially the CSSLR and feature-based model CSFLR to increase performance when compared to the case where you have a small background population since these are the models that should directly benefit

from having more data. SSSLR does benefit a bit as well but only in the amount of H_2 pairs it could make.

Large background population with low variability

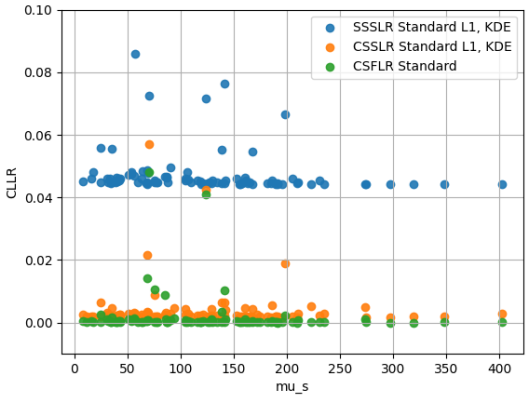


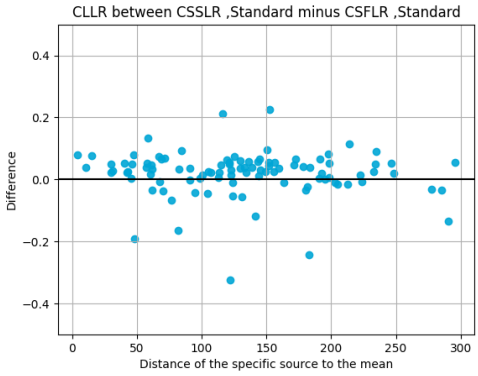
Figure 7.3: Scatterplot of CLLRs for the three likelihood ratio systems

Again we consider the scenario where we have a dataset that separates well. However, we now have a larger background population than before, while still having little information from our specific source. We see similar things happening here as the small background case. Again the CSFLR outperforms the other two models, although the CSSLR method now performs better than before. This is to be expected since we have a larger population and the CSSLR takes into account the whole population to model the specific source.

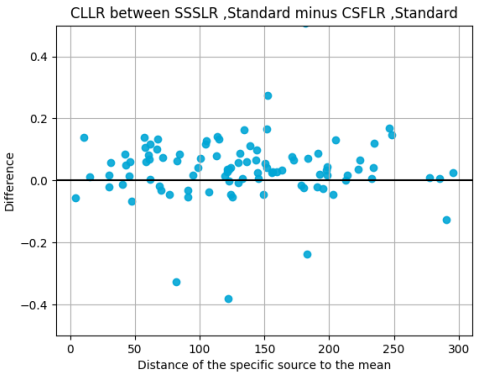
The SSSLR performs about the same as before, as in it performs worse than both the CSFLR and CSSLR. This is to be expected as not a lot changed for the SSSLR other than more H_2 pairs to work with but this is also true for the CSSLR.

Large background population with medium variability

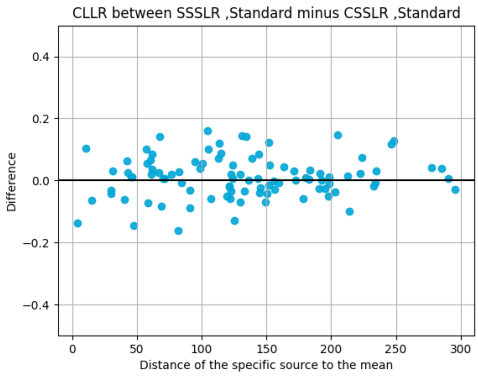
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR



(d) difference between SSSLR and CSSLR

Figure 7.4: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 25$

For this case, we see similar trends as for the case where we have a small background population but also some other things.

In figure 7.2a we saw that our models, in a lower background population, have a lot spread in their C_{llr} . This spread is reduced if we take a larger background population. As a consequence also the differences between models seem to be less varied compared to the small population counterpart, see figure 7.2. We now see the feature-based method performing slightly better than both the CSSLR and SSSLR. The SSSLR and CSSLR performed about equally.

The analysis was also done for higher variabilities where $\sigma_w = 50, 100$. These figures can be found in the appendix, in figures B.3 and B.4 respectively. However, the same conclusions follow for these difficulties for a medium within-variability.

7.1.3. Conclusions

The absolute performance of LR systems is greatly influenced by the relation between the within-source standard deviation and between-source standard deviation. The smaller the within-source standard deviation, the better the absolute performance across the board. We also saw that all models benefitted from a rarer specific source in cases with a medium within variability.

Relative performance-wise, the feature-based CSFLR performed the best across the board. In all scenarios, it performed either equal or better than the score-based methods. In low within-variability cases, it was even able to nearly approach zero. For medium variability cases, it was still the best-performing method. Our data is easy enough to get an accurate KDE even though we have little data available. This is amplified by using a Gaussian Kernel while simulating data that is from a multivariate

normal distribution. A different kernel could yield different results.

The CSSLR can be an effective model as a replacement for the SSSLR in a trace-reference case if we have little specific source data. No matter the background size, the CSSLR performed better than the SSSLR if we had low variability, and equally as well if we had medium variability. The CSSLR improved upon itself as N increased, even getting close to the feature-based method in a low variability setting.

The SSSLR performed the relative worst of the three in all scenarios. It was strictly worse than the CSSLR and CSFLR for low-variability scenarios. The lack of data available for the SSSLR also means that the ELUB bounds are not as broad so the SSSLR model cannot output as high LRs. In medium variability cases, it performed equally as well as a CSSLR.

7.2. Sufficient specific source information

In this scenario, we have a dataset containing sufficient data from our specific source, but with a smaller background population, In particular, we again take $r_{ss} = 50$, but now we have $N = 20$ sources. By increasing the specific source information but not the background population we expect the SSSLR to perform better, relative to the CSSLR and CSFLR.

7.2.1. Small background population

In this scenario, we have a dataset containing sufficient data from our specific source but with a smaller background population, In particular, we again take $r_{ss} = 50$, but now we have $N = 20$ sources. By increasing the specific source information but not the background population we expect the SSSLR to perform better, relative to the CSSLR and CSFLR.

Small background population with low variability

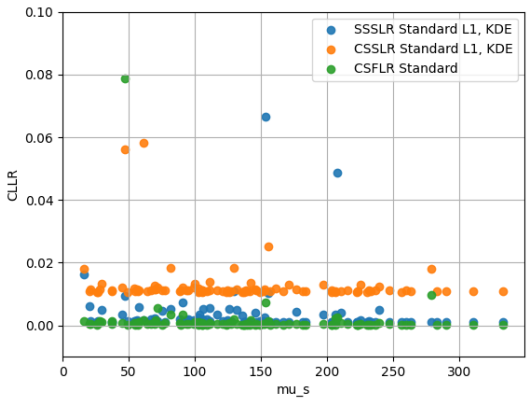
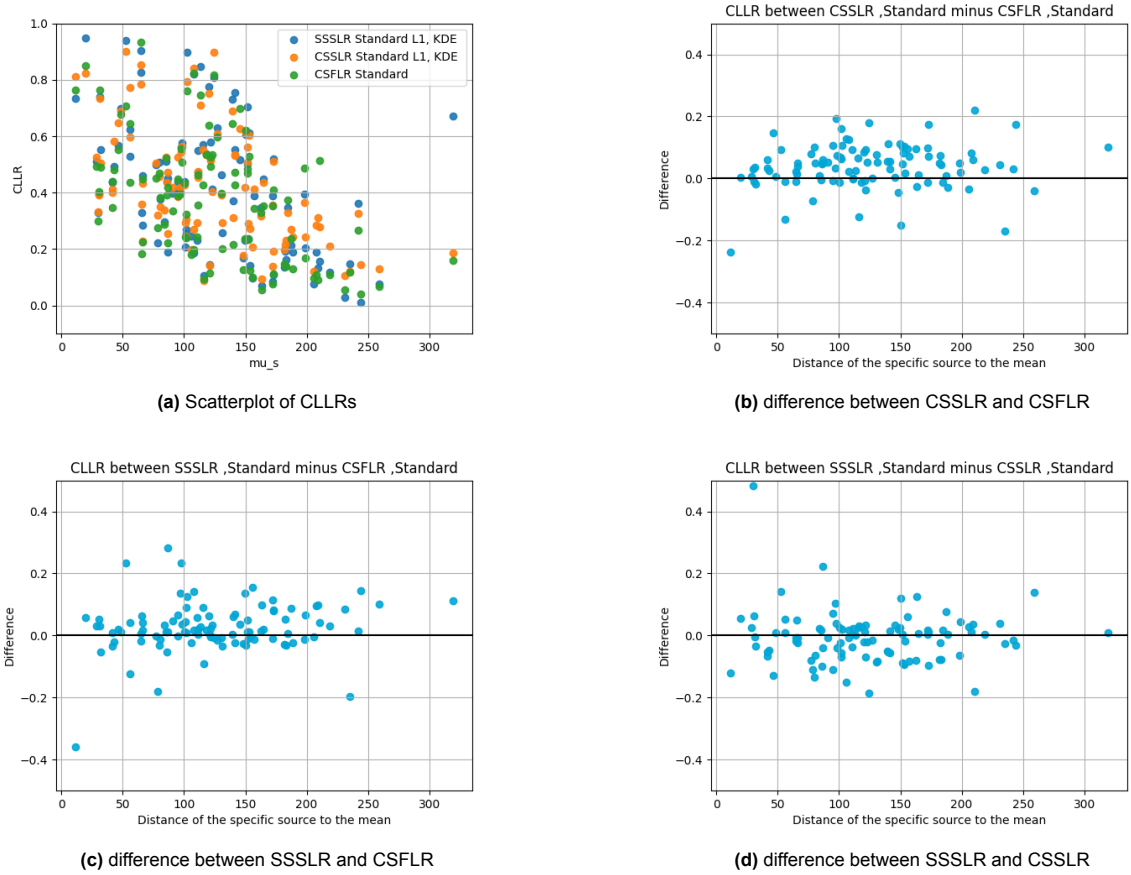


Figure 7.5: Scatterplot of CLLRs for the three likelihood ratio systems

In figure 7.5 we see something new happening: The SSSLR model outperforms the CSSLR model. As expected, increasing the amount of specific source measurements boosted the performance of the SSSLR performance and is here enough to surpass CSSLR in performance. The CSFLR model still performs the best in terms of absolute performance, although the SSSLR comes close in performance.

Small background population with medium variability



(a) Scatterplot of CLLRs

(b) difference between CSSLR and CSFLR

(c) difference between SSSLR and CSFLR

(d) difference between SSSLR and CSSLR

Figure 7.6: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 25$

When increasing the difficulty of the underlying population, the same patterns appear. All models improve when the specific source becomes rarer and the CSFLR method again outperforms the score-based models.

Between the SSSLR and the CSSLR, there is no big difference. This is surprising since one could expect that especially in a situation where you have a good approximation of your specific source while having few alternative sources, it would be logical to see that the SSSLR would outperform the CSSLR.

An explanation for this is found again in that all sources have the same within-variability. This means that despite having only 20 alternative sources, of which 16 end up in the training set, we are still able to get $16 \cdot \binom{5}{2} = 160$ training scores for H_1 for the CSSLR. This is plenty to compute a score distribution that resembles that of one of the SSSLR, which has way more scores for H_1. However, these are scores all from the same distribution.

Therefore, the CSSLR can accurately replace the SSSLR. It is however unlikely that in a real-life scenario, one has a dataset for which every source has the same inner variability as other sources. For these results see chapter 8

The analysis was also done for higher variabilities where $\sigma_w = 50, 100$. These figures can be found in the appendix, in figures B.5 and B.6 respectively. However, the same conclusions follow for these difficulties for a medium within-variability.

7.2.2. Large background population

In the last scenario, we consider a dataset containing sufficient data from our specific source and a large background population, In particular, we again take $r_{ss} = 50$, but now we have $N = 100$ sources.

Large background population with low variability

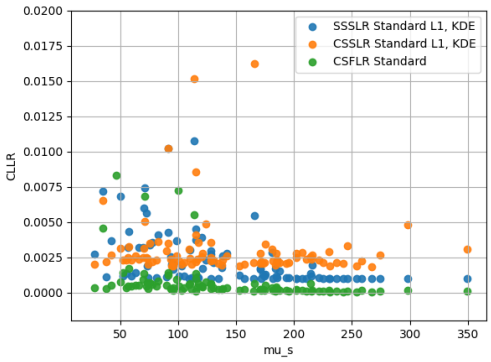
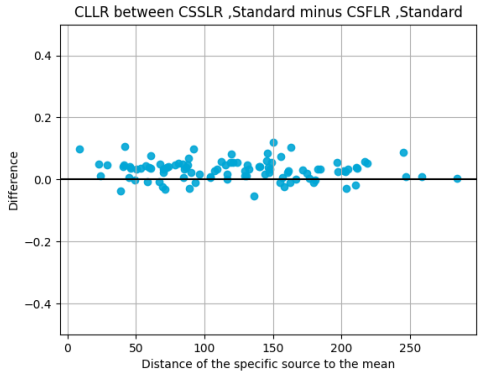


Figure 7.7: Scatterplot of CLLRs for the three likelihood ratio systems

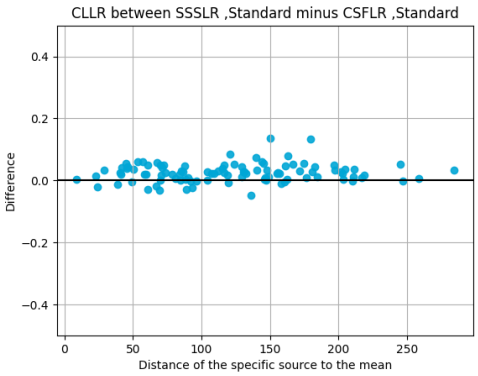
Again we consider the scenario where we have a dataset that separates well. In figure 7.7 we see similar things happening here as the small background case in figure 7.5. In terms of absolute performance, all models improve again. The worst performing model relatively speaking is the CSSLR, but even that one has C_{llr} of 0.0025 on average. Again the CSFLR outperforms both score-based models.

Large background population with medium variability

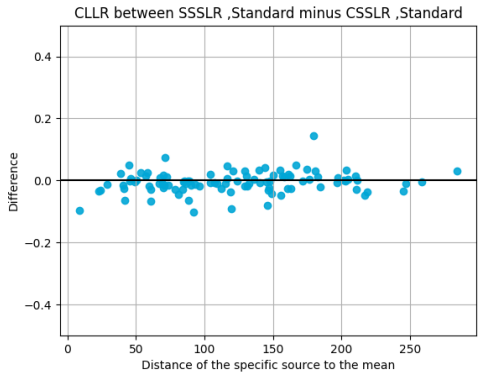
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR



(d) difference between SSSLR and CSSLR

Figure 7.8: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 25$

As our last scenario, we consider the case of a large background. Between the CSSLR and SSSLR, there is still not a big difference to be seen. Similar things happen here even though there is a larger background population present. Just as for the case where the size was 20 alternative sources, we have that the SSSLR and CSSLR perform around equally. This time we have a difference that has a lower variance.

The analysis was also done for higher variabilities where $\sigma_w = 50, 100$. These figures can be found in the appendix, in figures B.7 and B.8 respectively. However, the same conclusions follow for these difficulties for a medium within-variability. The only difference was that the absolute performance of all models decreased as difficulty increased. Nothing changed for the relative performances of models.

7.2.3. Conclusions

To conclude, in cases where we have a sufficient amount of specific source data, the SSSLR improved considerably as expected. We still have that the absolute performance of LR systems is greatly influenced by the relation between the within-source standard deviation and between-source standard deviation. Also here we saw that all models benefitted from a rarer specific source in cases with a medium within variability.

For a low variability case, the absolute performances of all models improved. The improvement of the SSSLR was the strongest as it now surpassed the CSSLR in low-variability cases, and was equally as good as the CSSLR for medium and higher variabilities.

This could also be due to having wider ELUB bounds as a consequence of having more data. It is now able to output higher LRs for the H_1 case and lower for the H_2 case.

The CSSLR does not perform poorly by any means, but still does not make sense to make CSSLR model if you have this much specific source data at hand. For medium variability, it performs equally but if we can make an SSSLR we should.

7.3. Conclusions

In conclusion, this chapter solidifies our intuition about LR systems. If we have access to a feature-based model, we should use this one as it performs the best in all cases.

However, when this is unfeasible we should make a specific source score-based model if we can. The SSSLR performs equal or better than a CSSLR in all scenarios when we have enough information.

If we cannot make an SSSLR due to having too little information about our specific source, a CSSLR will do just fine if the variability of the underlying data is not too high. It can still add information over the prior.

However, for cases where we had $\sigma_w = 50$ or higher, we saw that systems could have a C_{llr} greater than one, which suggests that on average these systems give misleading information. In this case, one should even consider not using an LR system in the first place.

7.4. Applications on glass data

In this section, we make a practical example to make the previous findings more tangible. This is based on a glass dataset used in the paper by the NFI that also described our steps of building LR systems [8].

In here they use a glass dataset from 320 sources of glass shards, consisting of two repeats each, which can be found at `https://raw.githubusercontent.com/NetherlandsForensicInstitute/elemental_composition_glass/main/duplo.csv`. This is part of a larger dataset containing five repeats per source. The dataset containing the other three repeats can be found at `https://raw.githubusercontent.com/NetherlandsForensicInstitute/elemental_composition_glass/main/duplo.csv`. Combining these two files gives us a dataset of 320 sources where each source has five repeats each. This would barely be enough to make a specific source model. The features consist of relative element concentrations found in the glass shards. From the combined dataset, we can estimate the between and within variances per element. This can be seen in the overview below:

	std_between	std_within_mean	std_within_max
K39	0.364540	0.011183	0.036226
Ti49	0.218870	0.022099	0.088237
Mn55	0.356963	0.011889	0.050480
Rb85	0.369021	0.022896	0.077233
Sr88	0.187598	0.017142	0.071697
Zr90	0.229569	0.028874	0.133441
Ba137	0.350275	0.018164	0.098842
La139	0.237110	0.026319	0.114090
Ce140	0.364245	0.015810	0.063262
Pb208	0.374391	0.027304	0.151423

Figure 7.9: Overview of the between and within variations of the glass data set

From this, we can see that generally speaking, this dataset separates well. For almost all elements the mean within deviation is of a smaller order of magnitude than that of the between within deviation. This suggests that we are in a low variability scenario, and it suggests that using a CSSLR can be useful.

Indeed, if we take the specific source to be the source identified by ”subject 1”, we can construct all models for this specific source. On top of that, we evaluated the models on a pair of H_1 and H_2 inputs that were not seen by the training data, which yielded an LR for all three models.

Name	C_{llr}	LR under H_1	LR under H_2
SSSLR Standard L1 KDE	0.597	1.2	0.25
CSSLR Standard L1 KDE	0.0592	11.9	0.0004
CSFLR Standard	2.866 e-05	25170.7	2.6e-51

Table 7.1: C_{llr} for glass data using subject 1 as the specific source

From the table, we observe the same things that we saw in our simulations. The CSSLR and CSFLR perform well, while the SSSLR model does not perform as well. It even performs significantly worse than the CSSLR relatively. This suggests that when in a low-variability setting, using a common source method can be effective for a trace-reference problem.

8

Performances for non-identical within variability

In the previous chapter, we assumed that each source and equal within standard deviation. This might however not be the case in practice. As we already saw for our example of ecstasy and glass, the maximum within standard deviation could differ from the average standard within deviation. Especially in the case of ecstasy, we saw in chapter 6 that the CSFLR was the worst performing method, contrary to what the theory from Vergeer [21] would suggest and to what was shown in chapter 7. Assuming equal within deviations can be misleading in practice. In this chapter, we expand on the base case of equal within variance by now considering cases where each source has its own within-standard deviation, coming from a prior distribution.

For each alternative source $i \in \{1, \ldots, N\}$ we draw a particular within deviation $\sigma_{w,i} \sim U[1, \sigma_{max}]$, where σ_{max} represents the maximum standard deviation possible for a source. In each particular simulation, the σ_{max} will be increased to allow a larger spread of possible within deviations. This will lead to a population with more variability. We will take, in line with previous chapters, $\sigma_{max} \in \{25, 50, 100\}$. We skip over the case of $\sigma_{max} = 1$ since this is just the same as identical sources with $\sigma_w = 1$.

To compare only the effects of varying the within-covariance, we fix the sizes of the background population and the amount of specific source measurements. In these simulations, we have $N = 100$ background sources with each $r = 5$ repeats and $r_{ss} = 50$ measurements from our specific source. This makes sure we have a CSSLR and SSSLR that should perform optimally. The experimental setup is the same, we simulate a 100 times a different dataset using this configuration and compute C_{llr} for all of them. Again the within-covariance matrix is a scalar multiple of the identity matrix, so $\Sigma_{w,i} = \sigma^2_{w,i} I$, but now differs per source i. We also have a particular specific source within deviation $\sigma_{w,ss}$. We then have the joint distributions for the pairs under H_1 and H_2

$$(\mathbf{X}, \mathbf{Y})|H_{1,c} \sim MVN(\mu^*, \Sigma_{1,c}) \tag{8.1}$$
$$(\mathbf{X}, \mathbf{Y})|H_{2,c} \sim MVN(\mu^*, \Sigma_{2,c}) \tag{8.2}$$

With $\mu^* = [\mathbf{0}, \mathbf{0}]$ and $\Sigma_{1,c} = \begin{bmatrix} (\sigma_b^2 + \sigma_{w,x}^2)I & \sigma_b^2 I \\ \sigma_b^2 I & (\sigma_b^2 + \sigma_{w,y}^2)I \end{bmatrix}$, $\Sigma_{2,c} = \begin{bmatrix} (\sigma_b^2 + \sigma_{w,x}^2)I & 0 \\ 0 & (\sigma_b^2 + \sigma_{w,y}^2)I. \end{bmatrix}$

Similarly, we now have for our joint specific source distribution:

$$(\mathbf{X}, \mathbf{Y})|H_{1,s} \sim MVN(\mu'_{\mathbf{1,s}}, \Sigma_{1,s}) \tag{8.3}$$
$$(\mathbf{X}, \mathbf{Y})|H_{2,s} \sim MVN(\mu'_{\mathbf{2,s}}, \Sigma_{2,s}) \tag{8.4}$$

With fixed $\mu'_{\mathbf{1,s}} = [\mu_{\mathbf{ss}}, \mu_{\mathbf{ss}}]$ and block covariance matrix $\Sigma_{1,s} = \begin{bmatrix} \sigma^2_{w,ss} I & 0 \\ 0 & \sigma^2_{w,ss} I \end{bmatrix}$

Under H_2, we have mean $\mu'_{\mathbf{2,s}} = \begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mu_{\mathbf{ss}} \end{bmatrix}$, with block covariance matrix $\Sigma_{2,s} = \begin{bmatrix} (\sigma_b^2 + \sigma_{w,x}^2)I & 0 \\ 0 & \sigma_{w,ss}^2 I \end{bmatrix}$

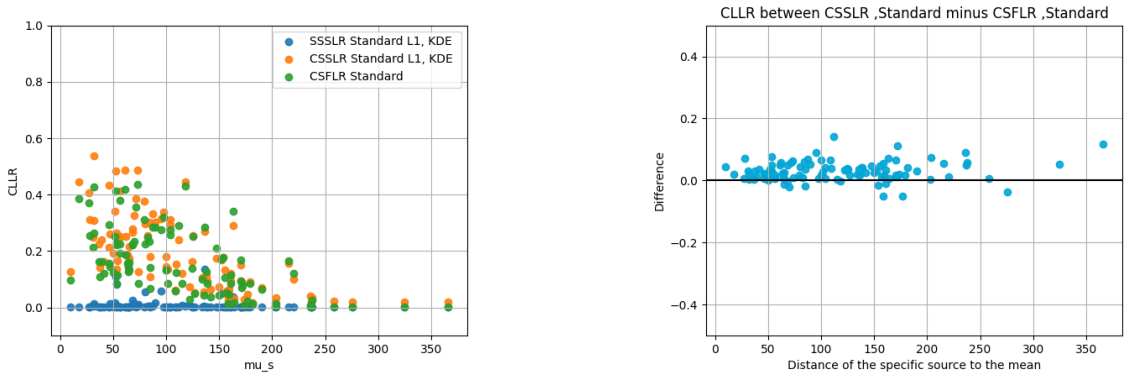
Three different scenarios will be considered. The first will be where the specific source has a low variability compared to the alternative sources. The second case we consider is where the specific source within-deviation is the average of the within-deviations of all sources. This will make the specific source behave similarly but not identically to alternative sources. The last case is where the specific source has a high within-deviation and therefore a difficult source.

Since we are simulating normals and using the absolute distance between features as the scorer, we could again write down an explicit formula for the distribution of the H_1 and H_2 scores for the common source and the specific source scenario. The only thing that changes in the case of non-identical sources is that we have to replace the σ_w by σ_x and σ_y respectively.

8.1. Minimal specific source uncertainty

Our first scenario is when we have a population in which our specific source has a minimal within-standard deviation compared to alternative sources. In particular, in each of the following simulations this section we have a specific source within-deviation of $\sigma_{w,ss} = 1$, while the alternative sources draw their within deviation from a prior distribution.

Medium variability of alternative sources



(a) Scatterplot of CLLRs

(b) Difference between CSSLR and CSFLR

Figure 8.1: CLLRS and differences between systems when for the specific source we have $\sigma_{w,ss} = 1$, but for each alternative source we have in the population we have $\sigma_w \sim U[1, 25]$

We start by considering the case where $\sigma_{max} = 25$. In figure 8.1 we immediately see the effects on the models. Firstly the SSSLR performs well, similar to identical sources with low within-variability. This makes sense since not a lot has changed for the SSSLR. The distribution for the scores under H_1 did not change. The distribution for the H_2 scores changed somewhat, but not too much.

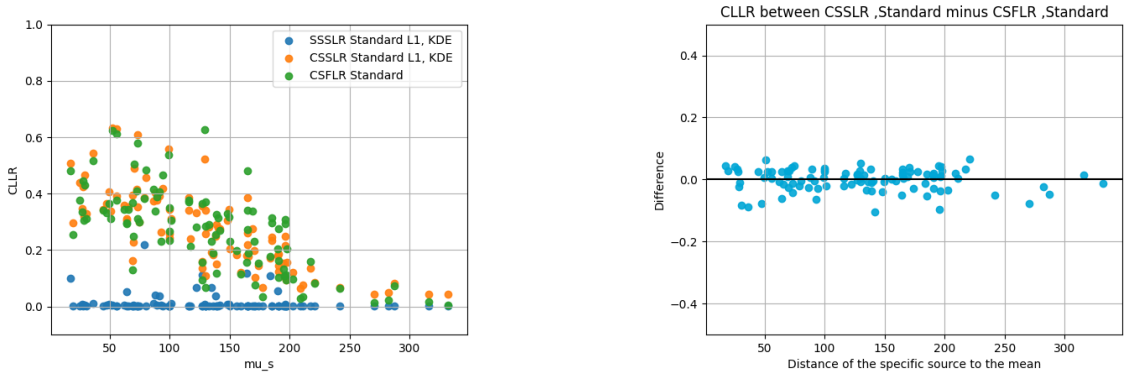
Next, we see that the CSFLR performs now worse than before. In formula 4.2 we saw that the feature-based method uses the mean within-covariance matrix to estimate the within covariance to compute the LR for a new trace and reference measurement. However, we now use the mean within-covariance matrix which is different from the true specific source within-covariance matrix. This gives us less accurate LRs.

We know the specific source has a within standard deviation of 1, so $\Sigma_{w,ss} = I$. For an alternative source, however, we have a within covariance matrix of $\sigma_{w,i}I$. The mean within-covariance matrix that the CSFLR uses, is the average over all sources, and therefore we have $\bar{\Sigma}_w = \frac{1}{n}\sum_{i=1}^{N} \sigma_{w,i} I$. The expected within covariance matrix is equal to $13I$ since $\mathbb{E}[\sigma_{w,i}] = \frac{25+1}{2} = 13$. The CSFLR therefore uses a within covariance larger than the actual occurring variance present in the specific source, and overestimates. This leads to poorer performance.

In figure 8.1b we see that the CSFLR still outperforms the CSSLR as the difference is above 0. The data is not difficult enough for feature-based to perform worse than the score-based counterpart. The CSSLR gets H_1 scores from alternative sources with larger within-standard deviation. As a result, the CSSLR has a different distribution for the H_1 scores and produces lower LRs for H_1 since the H_1 density has shifted away from the scores more likely for the specific source.

The score-based SSSLR has H_1 scores that separate very well from the H_2 scores like in the previous case for identical sources, due to having a low within standard deviation. It therefore performs well, which is logical.

High variability of alternative sources



(a) Scatterplot of CLLRs

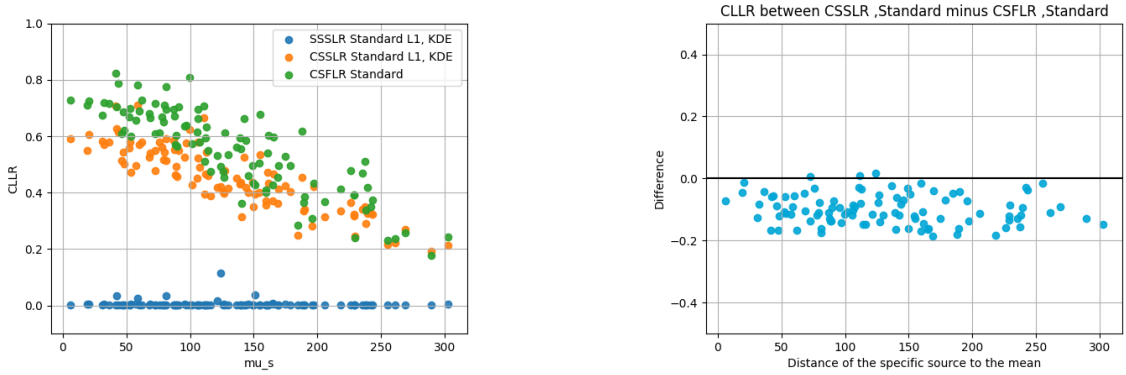
(b) Difference between CSSLR and CSFLR

Figure 8.2: CLLRS and differences between systems when for the specific source we have $\sigma_{w,ss} = 1$, but for each alternative source we have in the population we have $\sigma_w \sim U[1, 50]$

We continue by considering the case where $\sigma_{max} = 50$, in line with the previous chapter. It has the same sample size and specific source repeats, although now we have even more within-variability in our alternative sources. We generally see the same things happening except for one thing. The difference between the CSSLR and CSFLR now centers around 0, which means that the feature-based method has lost some performance, in an absolute sense but also compared to the CSSLR.

At some point, the mean within-deviation matrix is too far away from the true within-deviation matrix to produce reliable LRs. This of course also holds for the CSSLR but this fact seems to impact the CSFLR more.

Highest variability of alternative sources



(a) Scatterplot of CLLRs

(b) Difference between CSSLR and CSFLR

Lastly, we consider the case where $\sigma_{max} = 100$. We now see a clear distinction between the three models.

First of all the SSSLR is still by far the best performing method. Surprisingly the difference between the CSSLR and CSFLR now is below 0, which means that the CSFLR performs strictly worse than the CSSLR. The trend of losing more performance compared to the CSSLR continued.

The feature-based model now has an estimate for the within-standard deviation that is too far from the true specific source within deviation to give accurate LRs. More precisely, the fitted probability density for the CSFLR is too far off the true probability density surrounding the specific source.

Conclusions

In this section, we considered the case where we have a specific source with low within-variability compared to other sources in the population, in increasing spread.

We saw that the SSSLR as expected does very well in this case. Due to the low within deviation the scores under H_1 and H_2 separate well. This is true independent of the within-deviations of the alternative sources. It therefore performs equally as well in the three scenarios.

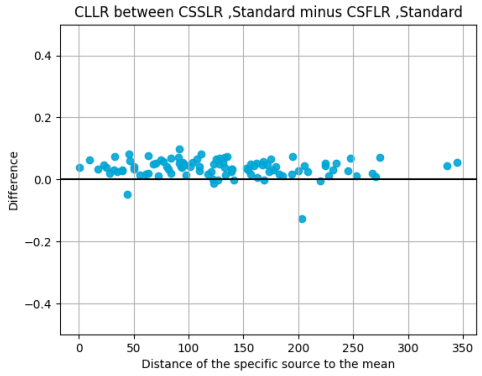
The CSFLR method also loses performance as the σ_{max} increases. As explained in section 8.1 the CSFLR uses the mean within covariance matrix from the population to estimate the within-covariance matrix and applies this per source. But as the σ_{max} increases, this estimated mean within-covariance matrix goes farther from the constant within-deviation of the specific source.

8.2. Average specific source uncertainty

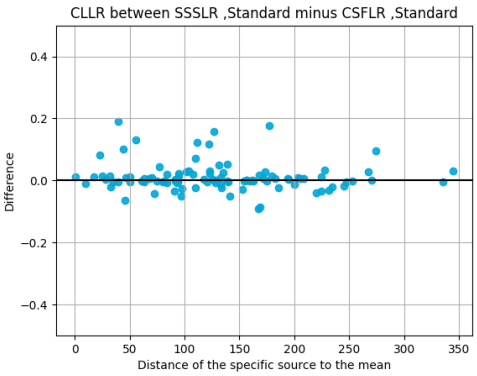
Our second scenario is when we have a population in which our specific source has an average within standard deviation compared to the alternative sources in the population. We again have for each alternative source random within deviation $\sigma_{w,i} \sim U[1, \sigma_{max}]$. For the within-deviation of the specific source we take the midpoint, and so $\sigma_{w,ss} = \frac{1+\sigma_{max}}{2}$.

Medium variability of alternative sources

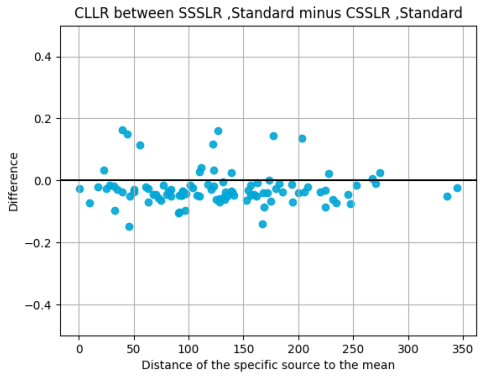
(a) Scatterplot of CLLRs



(b) Difference between CSSLR and CSFLR



(c) Difference between SSSLR and CSFLR



(d) Difference between SSSLR and CSSLR

Figure 8.4: CLLRS and difference between systems when for each alternative source we have in the population we have $\sigma_w \sim U[1, 25]$, but $\sigma_{w,ss}$ is equal to the midpoint of the interval

We skip again over the case where $\sigma_{max} = 1$ and go straight to the case where $\sigma_{max} = 25$. In figure 8.1 we see that the CSSLR performs worse than the SSSLR and the CSFLR.

It makes sense for the CSFLR to perform better than in the previous scenario where we only had a specific source within-deviation of $\sigma_{w,ss} = 1$. The expected mean within-covariance matrix used by the CSFLR now coincides with the true specific source within-covariance matrix.

The SSSLR performs worse than the case where we had a minimal within deviation of one, which is logical. Now it has a score distribution for H_1 that now overlaps again more with the H_2 scores. It still however outperforms the CSSLR. The CSSLR performs the worst now of the three models which again is explained by the H_1 scores that come from sources with another within deviation.

For higher $\sigma_{max} \in \{50, 100\}$ the trends described above continued, but with generally poorer performance due to each source having a higher within-variability. These plots can be found in the appendix B.2.1

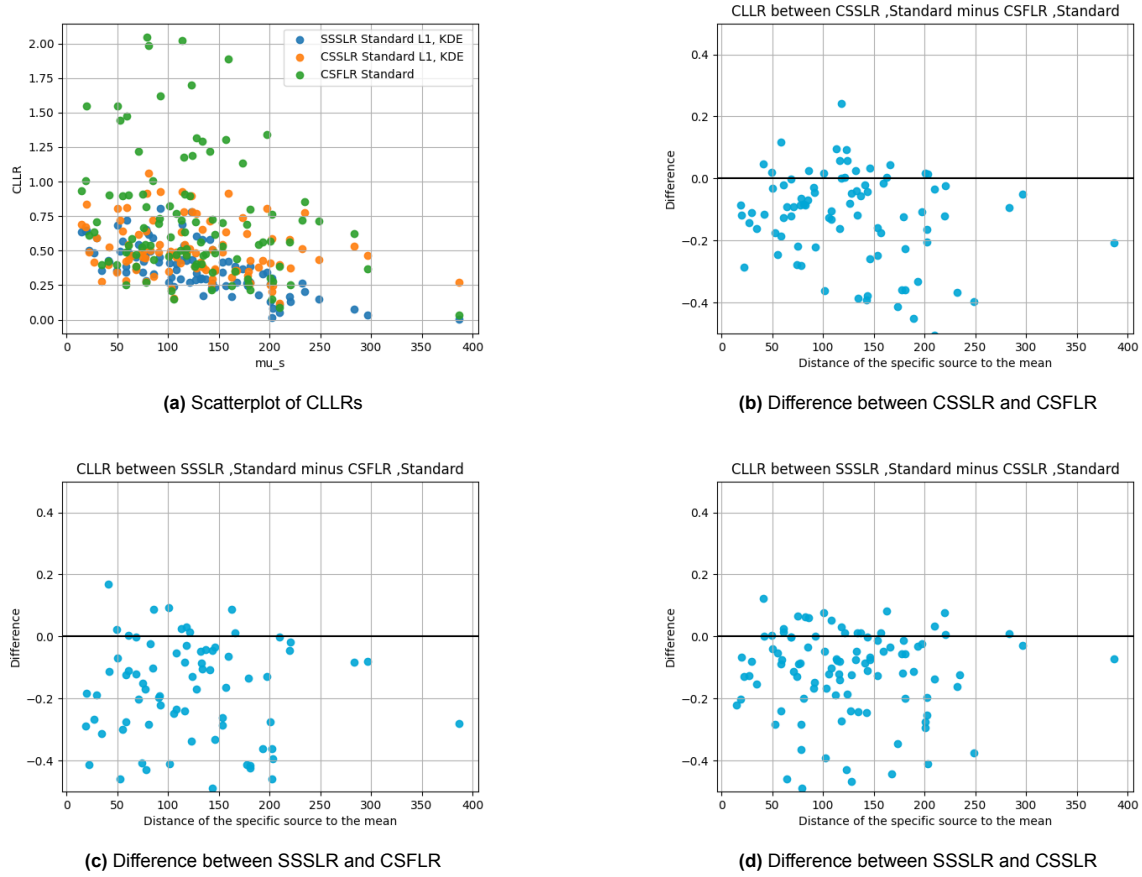
Conclusions

In this section, we considered the case where we have a specific source with an average within deviation compared to the population. We still see that the SSSLR performs the best, and the CSFLR can match the performance. This is coincidental since its expected mean within-covariance matrix now equals the true underlying within-covariance matrix from the specific source. The CSSLR model is in all cases the worst-performing model.

8.3. Maximal specific source uncertainty

Our last scenario is when we have a population in which our specific source has the maximal within standard deviation compared to the alternative sources in the population. We simulate $N = 100$ background sources with each $r = 5$ repeats and $r_{ss} = 50$ measurements from our specific source. We again have for each alternative source a random within-deviation $\sigma_{w,i} \sim U[1, \sigma_{max}]$. For the within deviation of the specific source now we take the endpoint so we have $\sigma_{w,ss} = \sigma_{max}$ in each scenario.

Medium variability of alternative sources



(a) Scatterplot of CLLRs

(b) Difference between CSSLR and CSFLR

(c) Difference between SSSLR and CSFLR

(d) Difference between SSSLR and CSSLR

Figure 8.5: CLLRS and difference between systems when for each alternative source we have in the population we have $\sigma_w \sim U[1, 25]$, but $\sigma_{w,ss}$ is equal to the endpoint of the interval, $\sigma_{w,ss} = 25$

In figure 8.5 it can be seen that for $\sigma_{max} = 25$ all models perform poorly. In particular, we get for the CSFLR outliers with a C_{llr} above 1. The differences are clear in which model outperforms which. Both score-based models outperform the feature-based model, and the SSSLR outperforms the CSSLR.

Previously, in the case where we had a minimal specific source within deviation $\sigma_{w,ss}$ we saw that the CSFLR started better than the CSSLR but lost performance as the σ_{max} increased and eventually became worse. Here however it is already worse from the get-go.

For a minimal specific source within-deviation, we also had that the CSFLR performed worse than the SSSLR. But we still had that the C_{llr} never exceeded 1, see figure 8.5a. Again, the CSFLR performs worse than the SSSLR which is to be expected. However, the feature-based method now underestimates the true variance of the specific source, which is more detrimental than over-estimating the variance.

The best-performing method is still the score-based SSSLR although it has a lot of variance in its performance. There is again a slight downward trend to be seen for the SSSLR, as well as for the

CSSLR. The CSSLR also outperforms the CSFLR here.

The simulations were also done for $\sigma_{max} \in \{50, 100\}$, but these yielded the same results but more extreme. For $\sigma_{max} = 100$, most models had a C_{llr} of above 1.

Conclusions

In this section, we considered the case where we have a specific source with the maximal within-deviation possible compared to other sources in the population.

We still see that the SSSLR performs the best of the three. The CSFLR method performs the worst of the three methods at all stages. As explained before, this is due to the model using a within covariance that assumes a smaller variance than the actual variance from the specific source. This leads to poor performance and large outliers.

Therefore, when you have a situation in which you have a population for which each source has a widely different within deviation, and a specific source whose within variation is also large, then it is possible that none of these three models would give you an LR system that should be used in practice.

8.4. Conclusions

In this chapter, we researched the scenario in which each source had a unique within-variability. We saw that, in this case, the performance of the CSFLR is heavily dependent on its estimation of the within-variance of the specific source. If the specific source behaves differently than other sources in our population, the CSFLR performs poorly compared to the optimal SSSLR. This effect is even stronger if the feature-based method underestimates the specific source within-variability.

In all cases, the SSSLR was the best-performing method. This is logical since it is the only method with access to the true underlying distribution of the specific source, which now differs from other sources. The CSSLR cannot properly replace the SSSLR in this case. The only case where both the CSFLR and CSSLR were somewhat comparable to the SSSLR was when the specific source had an average within-variability.

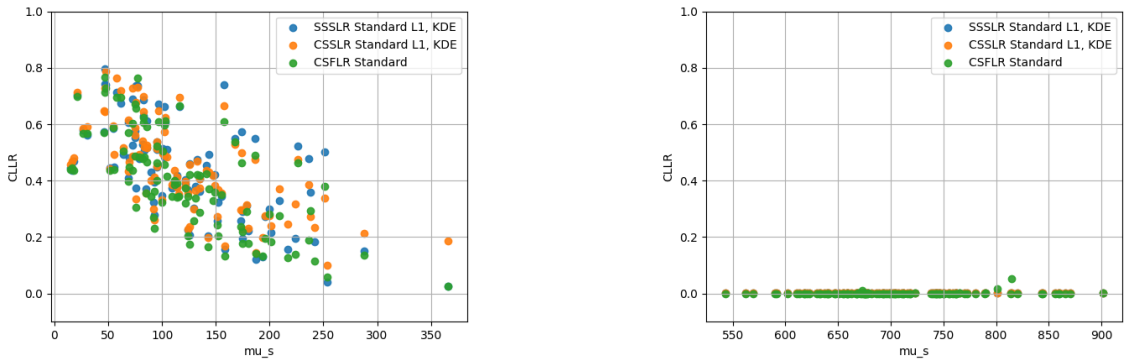
9

High-dimensional likelihood ratio systems

Another factor to consider for our investigation in the distinction between feature-based and score-based models is the dimensionality of the underlying data. It is known that higher dimensionality can be detrimental to the accuracy of correct estimation if the sample size is not adequately large enough. When this happens the data becomes sparse. Score-based methods do not suffer from this phenomenon as there is no parameter estimation being done for the features. It is expected that score-based models will outperform feature-based models when the dimensionality becomes high.

In this chapter, we investigate the effects of increasing dimensionality for the CSSLR, SSSLR, and CSFLR. We consider three levels of within-variability, assuming identical sources.

9.1. Medium within-variability



(a) Dimension is 2

(b) Dimension is 50

Figure 9.1: Performance of the likelihood ratio system as the dimension increases for a medium difficulty of $\sigma_w = 25$

In figure 9.1 we see what happens when we increase the dimension for within deviation of $\sigma_w = 25$. We see that for this difficulty all models improve as the dimensionality increases.

A possible explanation could be that for a low dimensional scenario there is a higher probability for an alternative source to be similar to the specific source, simply because there are fewer features to differ in.

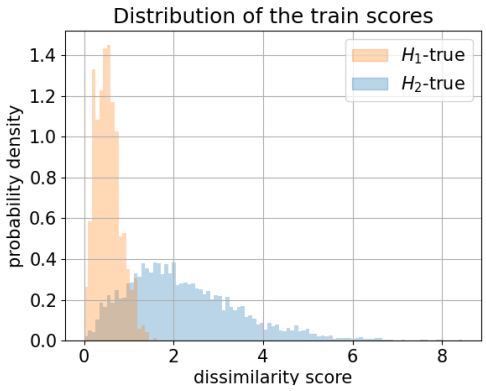
If we take as an example our ecstasy dataset: Suppose we have two batches of tablets, A and B, which are very similar in terms of the diameter and weight of the tablets, but A and B differ a lot in their purity.

Assume we only build a feature-based model for ecstasy tablets using diameter and weight as our features. Assume we have a trace tablet X which secretly is from batch A, and Y as our (known) reference measurement coming from batch B.

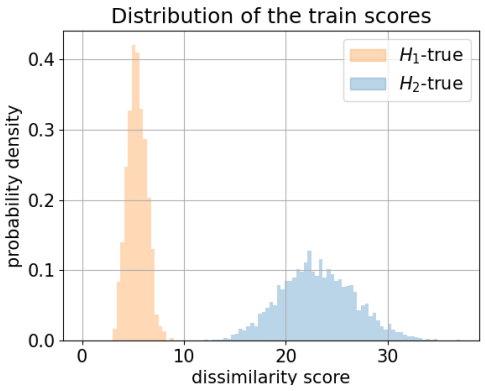
The hypothesis is now that X originates from batch B. However since we only look at the diameter and weight, we would get that X and Y are very similar, even though the truth is that they are from different batches.

That is also what is happening here. Initially, due to the higher dimension, the models can make a better distinction between the H_1 and H_2 distribution of the features and scores. Our dataset consisting of 100 alternative sources and 5 repeats is sufficiently large enough to have an accurate enough feature-based model for this dimension.

We can also this phenomenon in the distribution of the scores. If we look at the distribution of the scores for the CSSLR for dimensions 2 and 20 for example, we see that the distribution of the scores separate really well for higher dimensions.

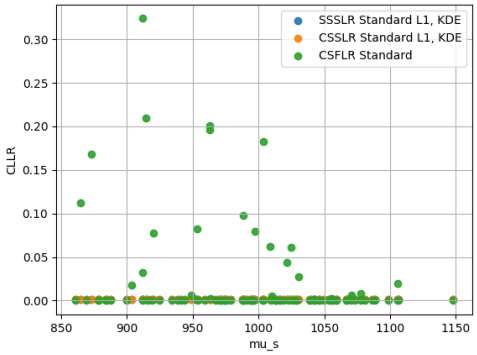


(a) Score distribution for the CSSLR for dimension equal to 2

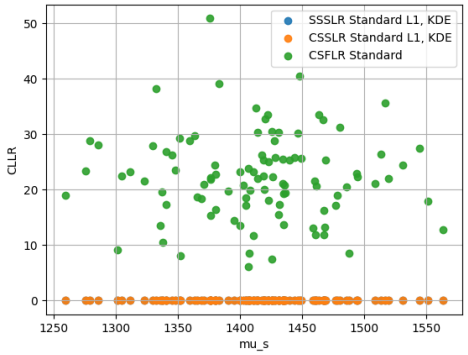


(b) Score distribution for the CSSLR for dimension equal to 20

However if we extend the dimension even further we see that the feature-based method still breaks down at some point. In figure 9.3b we see what happens when we consider even larger dimensionalities. For case where the dimension is 100, we can see that the CSFLR will have some outliers already, and increasing the dimension even further to 200 we see that the CSFLR breaks down comically.

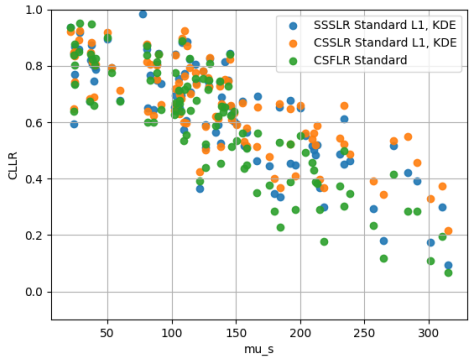


(a) C_{llr} for systems where the dimension has been further increased to 100

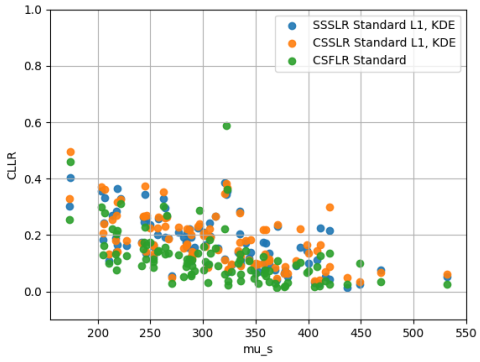


(b) C_{llr} for systems where the dimension has been further increased to 200

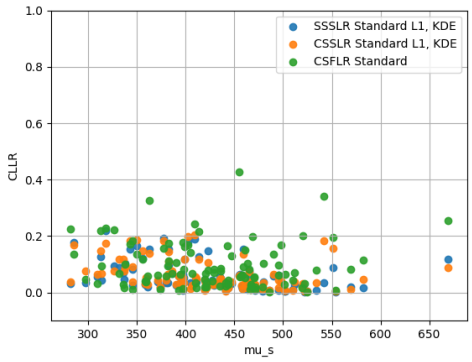
9.2. High within-variability



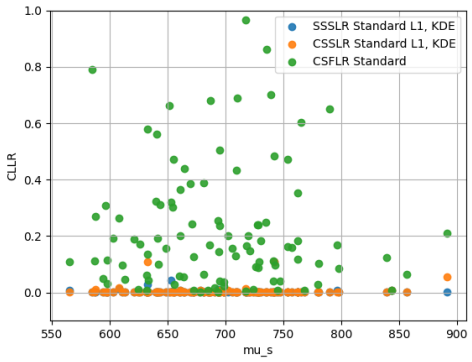
(a) Dimension is 2



(b) Dimension is 10



(c) Dimension is 20



(d) Dimension is 50

Figure 9.4: Performance of the likelihood ratio system as the dimension increases for a higher within-variability of $\sigma_w = 50$

In figure 9.4 we see what happens when we increase the dimension for a within-deviation of $\sigma_w = 50$.

Similar things happen as in the case where we had $\sigma_w = 25$. Initially, all models improve as the dimension grows, but at some point, the feature-based method fails to improve while the score-based methods perform well. Since we have a fixed size of our dataset, it seems that for lower dimensions we have a sufficient amount of data to get a good feature-based model, but as the dimension increases the amount of data becomes insufficient.

The breakdown for the CSFLR now happens at a lower dimension than for a smaller within deviation. We see that for a dimension $d = 20$ the feature-based method performs worse than the score-based methods which extends further for $d = 50$.

9.3. Conclusions

We see that for higher dimensional LR systems, depending on the difficulty of the underlying data set, the performance generally breaks down at some point for feature-based methods. High dimensionality can be advantageous for score-based models if the underlying data is sufficiently easy.

10

Percentile rank

As mentioned in section 2.5 a problem present in score-based methods is that these models do not account for the typicality of the features. In section 4.1 a preprocessor based on percentile ranking was proposed by Matzen et al. [9] as a way of incorporating typicality.

In this chapter, we will apply this preprocessing method in our score-based models SSSLR and CSSLR to see if these models incorporate typicality in their evaluation of measurements while also researching if these models do not lose performance compared to using Z-score normalization.

To reiterate, using percentile rank as a preprocessor replaces the features of our vector by their empirical distribution. For feature $k \in \{1, \dots, d\}$ it takes from each measurement its outcome in the k-th feature $\{X_{(1,k)}, \dots, X_{(n,k)}\}$ and computes the empirical distribution function for that feature

$$F_{k,n}(t) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}\{X_{(j,k)} \leq t\} = \frac{\text{amount of samples in training set with feature} \leq t}{n}$$

We still split off a training and validation set. We compute for each feature $k \in \{1, \dots, d\}$ the empirical distribution function $F_k(t)$ based on the training data. Then we take all our measurement vectors in the training set and replace the features by their evaluation in the empirical distribution function for that feature, and so we transform $X = (x_1, \dots, x_d)$ to $X^* = (F_1(x_1), \dots, F_d(x_d))$. After that, we still make pairs according to the system we are building and apply our scoring function of choice. We will also use L_1 norm as our scoring function and use a KDE as the calibrator of choice. In particular, we now get our scores:

$$\delta(X, Y) = \sum_{k=1}^{d} |F_{k,n}(x_k) - F_{k,n}(y_k)|$$

Here $F_{i,n}$ is the empirical distribution function based on our dataset for feature k. If X and Y truly originate from the same source, their measurements will also be close to each other in the empirical distribution function after transformation, resulting in low H_1 scores. If X and Y do not originate from the same source then their transformed measurements are more likely to be further away, resulting in higher scores for H_2-true measurements.

These properties however also hold for regular Z-score normalization. However percentile rank aims to account for typicality in the following way. Consider a one-dimensional feature, and we observe measurements X and Y. Without loss of generality, we can assume $X > Y$, since swapping X and Y

does not matter as we take the absolute value. As the score we now get:

$$\begin{aligned}\delta(X,Y) &= |F(X) - F(Y)| \\ &= \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{X_j \leq X\} - \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{X_j \leq Y\} \\ &= \frac{1}{n}\sum_{j=1}^{n} \mathbb{1}\{Y \leq X_j \leq X\} \\ &= \frac{\text{amount of samples in between X and Y}}{n}\end{aligned}$$

If we go back to our example of using lengths: In case 1 we measure a trace length of 182 cm and a reference length of 180 cm. In case 2 we observe a trace length of 222 cm and a reference length of 220 cm. Using Z-score normalization would lead to equal scores and thus equal LRs. Using a percentile rank the following happens: We still plug in our measurements but now in our training set we have some samples whose length is between 180 and 182, and the more samples we have that lie in between the larger the score. For a common length like 180, there probably will be more measurements in our dataset that have a length between 180 and 182. Therefore, this score will be relatively larger and we get a relatively lower LR for these measurements.

However in case 2 if we have a measured trace of length 222 and a reference of 220 cm, it is far less likely that we have measurements in our dataset that have a length between these extreme heights. Therefore these input measurements will have a lower score and it becomes more likely that the trace measurement is from the same source as the reference. In this way, percentile rank can take into account typicality.

10.1. Typicality for the SSSLR

Percentile rank as a method aims to account for the typicality of the observations, where if we have less typical measurements we expect to get a higher LR. However, in a specific source score-based model, we have the model accounts for the typicality.

This is again demonstrated better by an example: Assume we again have a case in which we have a trace person of length 182 and a suspect (the specific ”source” of the length) which is 180 cm. Our score is again the absolute difference of 2 cm. Recall the definition for the SSSLR and CSSLR:

$$\text{SSSLR} = \frac{f(\delta(X,Y)|H_{1,s})}{f(\delta(X,Y)|H_{2,s})}$$
$$\text{CSSLR} = \frac{f(\delta(X,Y)|H_{1,s})}{f(\delta(X,Y)|H_{2,s})}$$

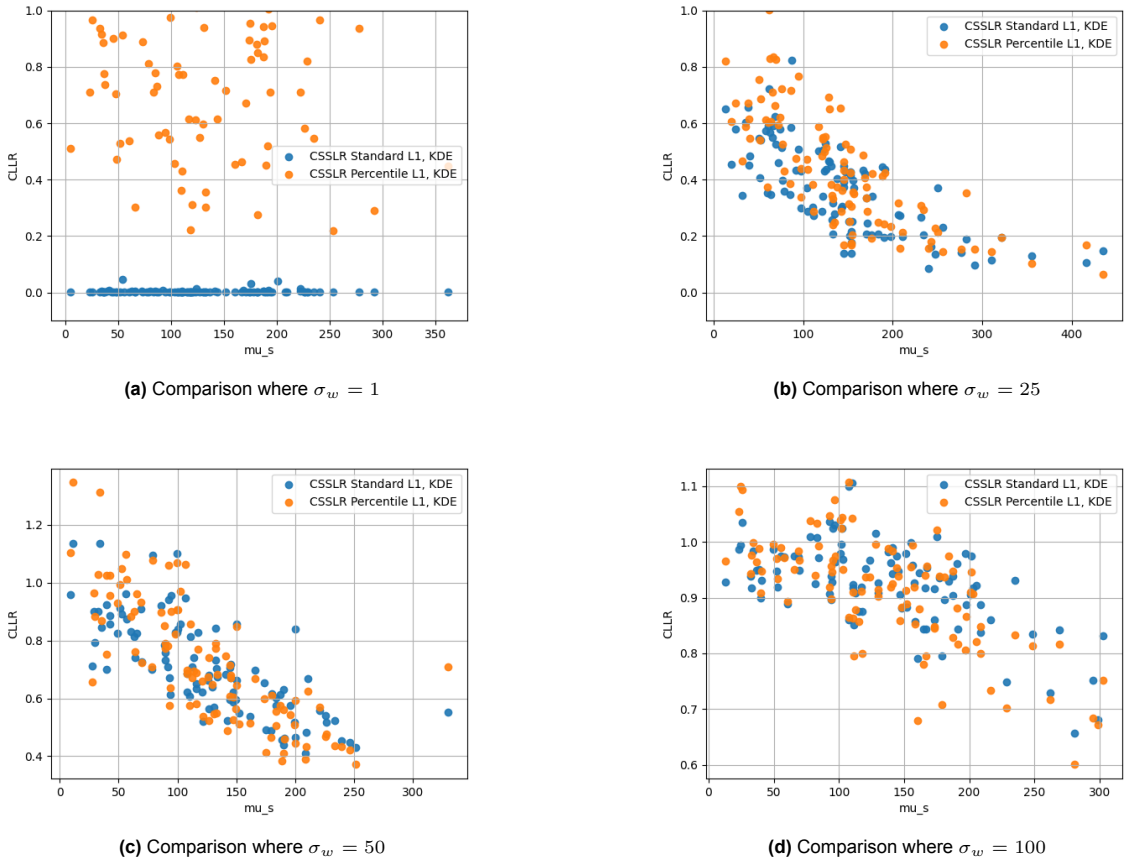
If we build a specific source model we would have a ratio of how often we expect to measure a score of 2 assuming these measurements are from the same known suspect or different people. If we have a suspect whose length is typical, we expect a medium LR since many other people differ from the specific source by a distance of 2, namely all the people with a length of 182 and 178 cm.

The story changes now if we have a suspect with atypical length. Assume we have a case where we have a trace of length 222 and a suspect of length 220. The denominator now looks at how many other people in our dataset differ by a distance of 2. Since our suspect has a length of 220, we count the other people with a length of 218 and 222, of which there are substantially fewer in our population. Therefore, it has become more likely that the trace measurement is from the tall suspect, without altering our score function.

In the common source model, nothing changed, as it still makes a model for the scores of the whole population. It still sees a score of 2 and outputs the same LR as for case 1. The common source model therefore has no inbuilt mechanism to account for typicality, while the specific source already does this. It makes more sense to investigate a percentile rank approach for common source methods since the typicality of the measurements is not already accounted for in the model itself.

10.2. CSSLR comparison

In this section, we study the applicability of the percentile rank preprocessor only to the CSSLR method. We again take $\sigma_b = 100$ and simulate two-dimensional data where each source has 5 repeats.



(a) Comparison where $\sigma_w = 1$

(b) Comparison where $\sigma_w = 25$

(c) Comparison where $\sigma_w = 50$

(d) Comparison where $\sigma_w = 100$

Figure 10.1: The C_{llr} for the CSSLR using a standard or a percentile rank preprocessor for various within deviations, given a larger background population

In figure 10.1 we can see the effects. We now see differences compared to the case where we had a small background population. Already for $\sigma_w = 25$ the CSSLR-PR seems to approach the standard CSSLR in performance and for the case where $\sigma_w = 50$ we see that they perform about equally as good. Note that all models still lose performance as the within deviation increases. For σ_w above 50, no model gets a C_{llr} lower than about 0.4. However relative to the standard CSSLR, the CSSLR-PR improves a lot and is in the same region of performance quicker than for a smaller population.

10.2.1. Conclusions

We conclude that for the CSSLR-PR the most important factors for performance is a combination of difficulty of the underlying population and the sample size. In contrast to the regular CSSLR, having a low within deviation per source leads to big outliers for the CSSLR-PR, especially when you have a small background dataset.

As the within deviation becomes larger, the percentile rank method also loses slight performance but not as heavily as the standard CSSLR. Relative to the standard CSSLR, the CSSLR-PR is actually improving.

The whole aim of researching PR was to see if using this preprocessor could mean if we can take into account typicality for score-based models.

We saw that on average models perform slightly worse using percentile rank method.

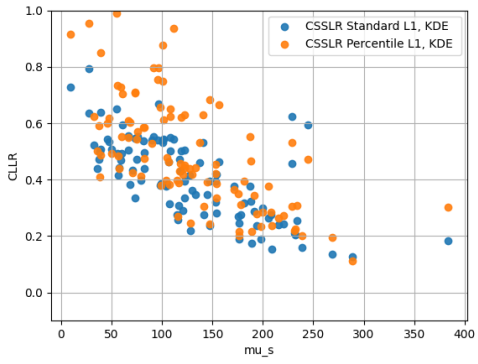
10.3. Large sample sizes

In previous sections, we saw the CSSLR-PR benefit from having a larger background population. This property is not unique to the percentile rank method. We also saw in chapter 7 that the standard score-based methods improved as the size of the background population increased which is a pretty logical notion.

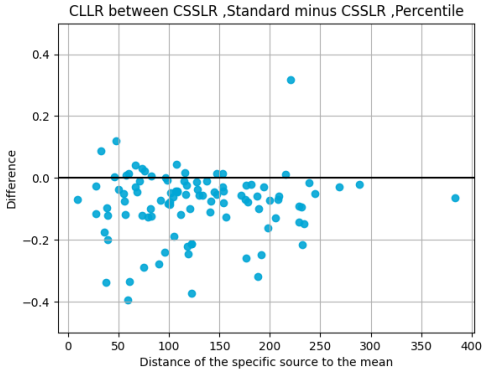
However, for the CSSLR-PR, the improvement was pretty significant as it not only improved in absolute C_{llr} but also relative to the standard CSSLR. In this section, we look at what happens if we increase the size of the background population even further, and if by increasing the sample sizes the percentile rank could outperform the standard method.

We distinguish between cases where we have a specific source from the background population and a case where we artificially put the specific source at the boundary of the population and further. The simulations are done with 50 specific source repeats, 5 repeats per alternative source, and a between variation $\sigma_b = 100$. Each source has an identical within variation of $\sigma_w = 25$, including the specific source.

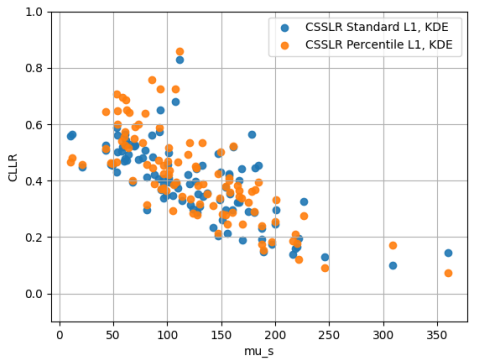
Standard specific source



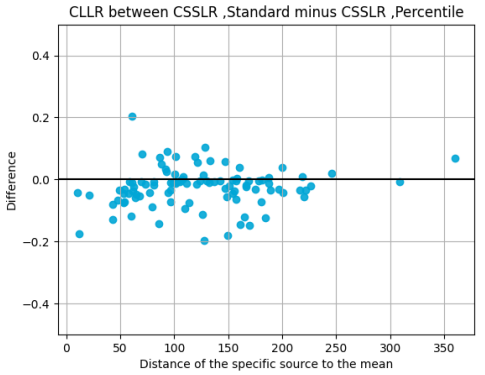
(a) $N = 200$



(b) Difference between standard and percentile rank preprocessor for $N = 200$



(c) $N = 400$



(d) Difference between standard and percentile rank preprocessor for $N = 400$

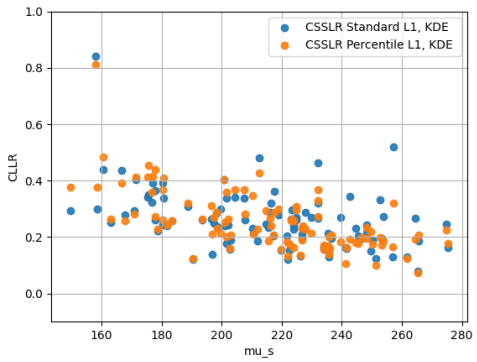
Figure 10.2: Performances for the CSSLR and CSSLR-PR for larger sample sizes, with a specific source within the boundaries of the population

In figure 10.2 we see for various background sizes the differences between the CSSLR-PR and the standard CSSLR. For all different sizes, the differences seem to be below 0 on average, signaling that the standard CSSLR still outperforms the CSSLR-PR.

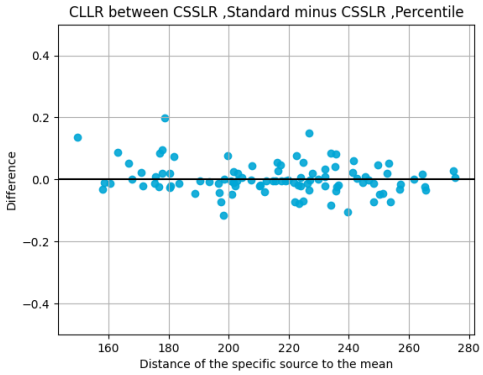
For the largest sample size however where $N = 400$, the difference between the CSSLR and the

CSSLR-PR seems to be the smallest, even though it looks like the regular CSSLR still outperforms the CSSLR-PR.

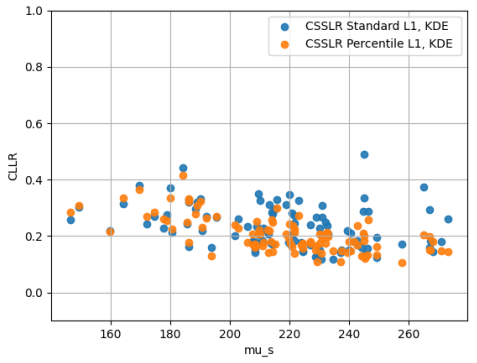
Increasing rarity



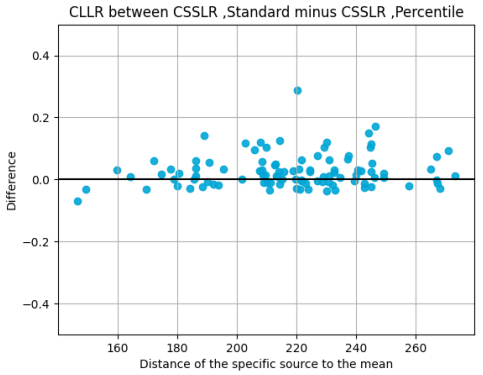
(a) $N = 200$



(b) Difference between standard and percentile rank preprocessor for $N = 200$



(c) $N = 400$



(d) Difference between standard and percentile rank preprocessor for $N = 400$

Figure 10.3: Test

Now we consider the case where we increase the specific source rarity even further. In figure 10.3 we see for various background sizes the differences between the CSSLR-PR and the standard CSSLR.

When we extend the distance from the specific source to the mean even further the CSSLR-PR improves performance-wise for larger sample sizes. For the size $N = 200$, the CSSLR and CSSLR-PR perform about equally, but when we increase the sample size even further the CSSLR-PR overtakes the regular CSSLR.

10.3.1. Application to simulated and glass data

In addition, to make the above simulations more tangible, we give a simulated example and an example from our glass set.

Simulated example

We simulate a dataset of 400 identical sources, each with 5 repeats and a specific source at the boundary with 50 repeats. The between-source deviation $\sigma_b = 100$, and $\sigma_w = 25$. Our source resides at the boundary:

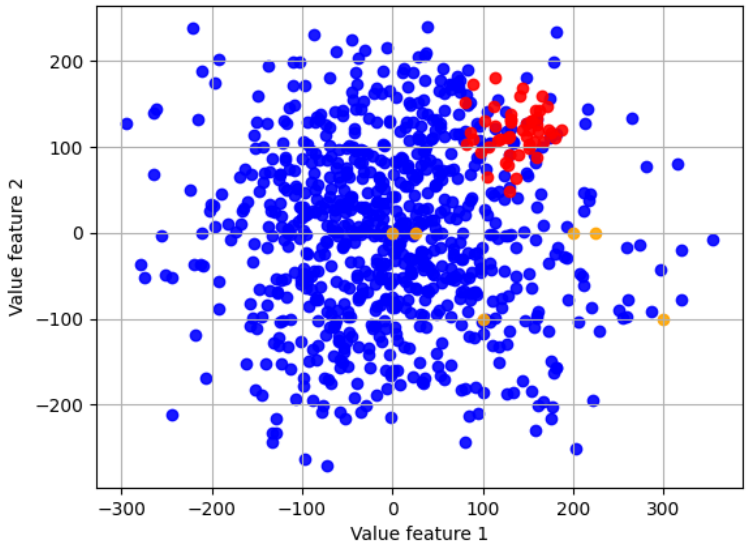


Figure 10.4: Simulated data with a specific source at the boundary in red, with 6 trace measurements located in our datset

From the simulated data, we compute the C_{llr} for a CSSLR and a CSSLR-PR. But we also want to test if our LR systems account for the typicality of new measurements. To this end, we define six trace measurements X_i:

$$X_1 = [0, 0], X_2 = [25, 0], X_3 = [100, -100]$$
$$X_4 = [200, 0], X_5 = [200, 225], X_6 = [300, -100]$$

We will take (X_1, X_2) and (X_4, X_5) as our H_1 true pairs, but one pair has measurements from the middle of our population while the other lies at the boundary. Note that we took the traces in such a way that their true distance is the same, so using a CSSLR will now get the same score for the pairs (X_1, X_2) and (X_4, X_5) despite the latter being less typical.

We take (X_1, X_3) as our 'typical' H_2 pair, and (X_4, X_6) as our H_2 pair that is non-typical. In table 10.1 we can see C_{llr} and LR's for these input values. The CSSLR-PR outputs a larger LR for the H_1 pair at the boundary as intended, and has a lower C_{llr} .

Name	C_{llr}	Center H_1 pair	Center H_2 pair	Boundary H_1 pair	Boundary H_2 pair
CSSLR	0.268	15.15	0.006	15.15	0.006
CSSLR-PR	0.199	13.86	0.005	23.33	0.67

Table 10.1: C_{llr} and LRs for the CSSLR and CSSLR-PR

Glass

We again take the 320 sources, each having 5 repeats, and construct our likelihood ratio systems before, but now we include the CSSSL-PR. Just like in chapter 7, we compute an LR on specific source measurements that our training data has not seen before:

Name	C_{llr}	LR for H_1	LR for H_2
CSSLR Standard L1, KDE	0.059	11.9	0.00041
CSSLR Percentile L1, KDE	0.017	113.83	0.00040

Table 10.2: C_{llr} and example LR for a H_1 and H_2 true case

We see that the LR became greater for H_1, and (barely) lower for H_2. It is not possible to show a ten-dimensional figure, but for the first two features, K39 and Ti49, we can see that the specific source lies at the boundary:

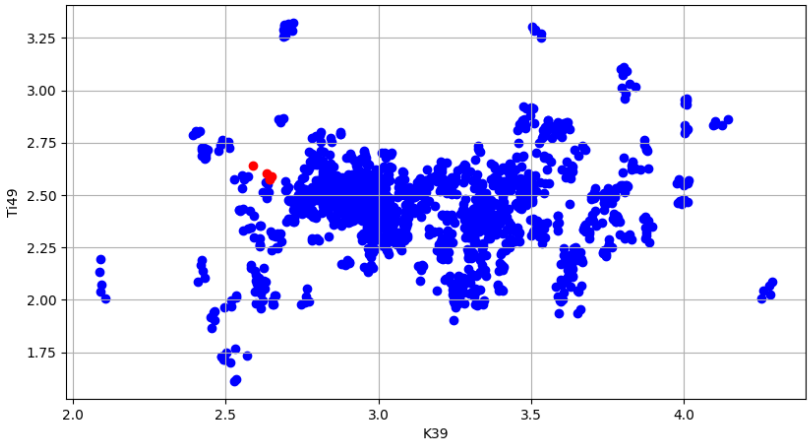


Figure 10.5: Visualization of the glass data for the features K39 and Ti49

10.3.2. Conclusions

We conclude that for larger sample sizes and a non-rare specific source, CSSLR-PR performs about equally as well as the regular CSSLR. If we have a case in which we have a specific source at the boundary then CSSLR-PR can even outperform a standard CSSLR. Therefore, for larger datasets, a CSSLR-PR is a better choice.

11

Conclusion and recommendations

In this research, we studied which practical factors play a role in the performance of likelihood ratio systems when considering evidence from an unknown trace and a known reference source. We considered three different likelihood ratio systems used to give the weight of evidence in court cases. These models are a feature-based CSFLR, a common source score-based model CSSLR, and a specific source score-based model SSSLR.

By simulating datasets that varied in several factors such as sample sizes, source variability, and amount of specific source measurements, we aimed to distill the influences of each factor on the performance of likelihood ratio systems.

The data simulated follows a classical framework called the two-level normal-normal model often used for continuous evidence. Here each source mean is simulated from a first-level multivariate normal distribution which captures the between-source variability. Using this simulated source mean, we generate measurements for our sources following a within-source distribution which is also a multivariate normal distribution.

For a two-dimensional scenario in which every source has an assumed equal within variation, we conclude that the most important factor for the performance of likelihood ratio systems is the relation between the between-source variation and within-source variation.

Another finding was that all models improved in terms of C_{llr} when the specific source moved further from the mean. For the CSFLR this was already incorporated in the LR. For the SSSLR this was caused by the observation that the H_2 scores grow larger while the distribution for H_1 stays the same. If we have a rare specific source then on average the other sources are further away so the H_2 scores are larger. For the CSSLR this does not affect the training scores but when validating with specific source data the rarity of the specific source still came into effect.

In datasets with little within-source variation, all models perform well in C_{llr} , of which the feature-based method performs best. If there is too little specific source data available, we can see that the CSSLR can even outperform the SSSLR in terms of C_{llr} due to having wider ELUB bounds. For other cases, the CSSLR performed equally as well as the SSSLR due to assuming equal variances.

Furthermore, we considered an extension where each source has its particular within-variation. We referred to these as non-identical sources and considered cases where we had a specific source that was lowly, averagely, or highly variable.

We showed that whenever alternative sources have a different variation, using a common source method in a trace-reference setting can yield poor performance. In particular; the CSSLR always performed worse than the SSSLR and the feature-based CSFLR only performed well if its approximation of the estimated mean within-covariance matrix was close to the true within covariance of the specific source.

Thirdly we considered high-dimensional likelihood ratio systems. We observed that regardless of the initial complexity of the dataset, the feature-based approach eventually breaks when dealing with high dimensional data.. Score-based models on the other hand can benefit from higher and do not suffer from high dimensionality.

Lastly, we considered a type of preprocessor called percentile rank that aims to incorporate typicality in score-based likelihood ratio systems. We saw that for smaller sample sizes using a percentile-rank preprocessor often yielded worse performance than using a regular Z-score transformation. However, when we increased the sample sizes even further we saw that the CSSLR with percentile rank improved in terms of raw C_{llr} , while also giving likelihood ratios that incorporate the typicality of the measurements.

Future work

From the subjects covered in this thesis, I would propose the following future work and recommendations:

- In this research the only model for data generation considered was the two-level normal-normal model which was a classical model in forensic evidence modeling. In practice, it is unlikely that real-life data follows a between-source normal variation. These experiments could be repeated for data that follows a different distribution.
- Furthermore we assumed that the simulated features and sources were independent which is a strong assumption. It is reasonable to think that, especially locally, certain characteristics in populations are more prevalent than in other locations.
- This thesis focused only on the L_1-norm as a score function. In practice, it is often not obvious which score function could be optimal for the data and different score functions could be researched, such as cosine similarity or Pearson correlation. The same holds for the calibrator of choice. A desirable property of a likelihood ratio system is that if the dissimilarity score becomes higher then on average the $\log_1 0(LR)$ should decrease. This might not hold for the KDE. This could be solved using different bandwidths, or looking at other calibrators.
- Increasing the dimensionality of the data resulted in the feature-based CSFLR breaking down if the amount of data was insufficient. Interesting future work could be to study the behavior of the CSFLR if we grow the sample size of the background population simultaneously to study the rate at which one would need to collect data to get a feature-based likelihood ratio system that performs well for higher dimensions.
- From the paper by Vergeer [21] an extension to score-based likelihood ratio is mentioned, so-called anchored score-based likelihood ratio systems. These are likelihood ratio systems where you condition on the outcome of the measurement of the trace or reference beforehand. It is proven that anchoring only improves performance for the limiting case, but it would be interesting to see how the performances of anchored likelihood ratio systems are for the same situations we simulated.
- In this thesis we considered a percentile rank preprocessor as a way of capturing typicality in score-based likelihood ratio systems. A topic of interest could be whether a machine learning approach can learn the typicality of the measurements and incorporate this into the likelihood ratio. Another idea could be to combine percentile rank in common source score-based likelihood ratio systems with high-dimensional data. We only simulated two-dimensional data, but if we have more dimensions for measurements to differ in, this could be beneficial for percentile rank.

References

[1] Annabel Bolck and Ivo Alberink. “Variation in likelihood ratios for forensic evidence evaluation of XTC tablets comparison”. In: *Journal of Chemometrics* 25.1 (2011), pp. 41–49. DOI: https://doi.org/10.1002/cem.1361. eprint: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/pdf/10.1002/cem.1361. URL: https://analyticalsciencejournals.onlinelibrary.wiley.com/doi/abs/10.1002/cem.1361.

[2] Annabel Bolck et al. “Different likelihood ratio approaches to evaluate the strength of evidence of MDMA tablet comparisons”. In: *Forensic science international* 191 (Aug. 2009), pp. 42–51. DOI: 10.1016/j.forsciint.2009.06.006.

[3] Wauter Bosma et al. “Establishing phone-pair co-usage by comparing mobility patterns”. In: *Science and Justice* 60.2 (2020), pp. 180–190. ISSN: 1355-0306. DOI: https://doi.org/10.1016/j.scijus.2019.10.005. URL: https://www.sciencedirect.com/science/article/pii/S1355030619300942.

[4] Niko Brümmer and Johan du Preez. “Application-independent evaluation of speaker detection”. In: *Computer Speech and Language* 20.2 (2006). Odyssey 2004: The speaker and Language Recognition Workshop, pp. 230–275. ISSN: 0885-2308. DOI: https://doi.org/10.1016/j.csl.2005.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0885230805000483.

[5] Niko Brümmer and Albert Swart. *Bayesian calibration for forensic evidence reporting*. 2014. arXiv: 1403.5997 [stat.ML].

[6] Franco Taroni Colin Aitken and Silvia Bozza, eds. *Statistics and the Evaluation of Evidence for Forensic Scientists, Third Edition*. Wiley, 2004. ISBN: 9781119245254.

[7] Joaquin Gonzalez-Rodriguez et al. “Emulating DNA: Rigorous Quantification of Evidential Weight in Transparent and Testable Forensic Speaker Recognition”. In: *IEEE Transactions on Audio, Speech, and Language Processing* 15.7 (2007), pp. 2104–2115. DOI: 10.1109/TASL.2007.902747.

[8] Anna Jeannette Leegwater et al. “From data to a validated score-based LR system: A practitioner’s guide”. In: *Forensic Science International* 357 (2024), p. 111994. ISSN: 0379-0738. DOI: https://doi.org/10.1016/j.forsciint.2024.111994. URL: https://www.sciencedirect.com/science/article/pii/S0379073824000756.

[9] Timo Matzen et al. “Objectifying evidence evaluation for gunshot residue comparisons using machine learning on criminal case data”. In: *Forensic Science International* 335 (2022), p. 111293. ISSN: 0379-0738. DOI: https://doi.org/10.1016/j.forsciint.2022.111293. URL: https://www.sciencedirect.com/science/article/pii/S0379073822001232.

[10] Didier Meuwly, Daniel Ramos, and Rudolf Haraksim. “A guideline for the validation of likelihood ratio methods used for forensic evidence evaluation”. In: *Forensic Science International* 276 (2017), pp. 142–153. ISSN: 0379-0738. DOI: https://doi.org/10.1016/j.forsciint.2016.03.048. URL: https://www.sciencedirect.com/science/article/pii/S0379073816301359.

[11] 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”. In: *Science & Justice* 58.1 (2018), pp. 47–58. ISSN: 1355-0306. DOI: https://doi.org/10.1016/j.scijus.2017.06.005. URL: https://www.sciencedirect.com/science/article/pii/S1355030617300849.

[12] Geoffrey Stewart Morrison, Jonas Lindh, and James M Curran. “Likelihood ratio calculation for a disputed-utterance analysis with limited available data”. In: *Speech Communication* 58 (2014), pp. 81–90. ISSN: 0167-6393. DOI: https://doi.org/10.1016/j.specom.2013.11.004. URL: https://www.sciencedirect.com/science/article/pii/S0167639313001635.

[13] Cedric Neumann and Madeline Ausdemore. “Defence against the modern arts: the curse of statistics—Part II: ‘Score-based likelihood ratios’”. In: *Law, Probability and Risk* 19.1 (Apr. 2020), pp. 21–42. ISSN: 1470-8396. DOI: 10.1093/lpr/mgaa006. eprint: https://academic.oup.com/lpr/article-pdf/19/1/21/33390883/mgaa006.pdf. URL: https://doi.org/10.1093/lpr/mgaa006.

[14] Danica M Ommen and Christopher P Saunders. “Building a unified statistical framework for the forensic identification of source problems”. In: *Law, Probability and Risk* 17.2 (May 2018), pp. 179–197. ISSN: 1470-8396. DOI: 10.1093/lpr/mgy008. eprint: https://academic.oup.com/lpr/article-pdf/17/2/179/25076590/mgy008.pdf. URL: https://doi.org/10.1093/lpr/mgy008.

[15] Daniel Ramos et al. “Information-Theoretical Assessment of the Performance of Likelihood Ratio Computation Methods”. In: *Journal of Forensic Sciences* 58.6 (2013), pp. 1503–1518. DOI: https://doi.org/10.1111/1556-4029.12233. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1556-4029.12233. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/1556-4029.12233.

[16] *Ruim 6500 verdachten opgepakt na hack berichtendienst EncroChat*. 2023. URL: https://nos.nl/artikel/2480473-ruim-6500-verdachten-opgepakt-na-hack-berichtendienst-encrochat (visited on 06/27/2023).

[17] Eleni-Konstantina Sergidou et al. “Frequent-words analysis for forensic speaker comparison”. In: *Speech Communication* 150 (Apr. 2023). DOI: 10.1016/j.specom.2023.03.010.

[18] B.W. Silverman. “Density Estimation for Statistics and Data Analysis”. In: *Chapman & Hall, London* (1986).

[19] Michail Tsagris, Christina Beneki, and Hossein Hassani. “On the Folded Normal Distribution”. In: *Mathematics* 2.1 (2014), pp. 12–28. ISSN: 2227-7390. DOI: 10.3390/math2010012. URL: https://www.mdpi.com/2227-7390/2/1/12.

[20] Andrew van Es et al. “Implementation and assessment of a likelihood ratio approach for the evaluation of LA-ICP-MS evidence in forensic glass analysis”. In: *Science and Justice* 57.3 (2017), pp. 181–192. ISSN: 1355-0306. DOI: https://doi.org/10.1016/j.scijus.2017.03.002. URL: https://www.sciencedirect.com/science/article/pii/S1355030617300266.

[21] Peter Vergeer. “From specific-source feature-based to common-source score-based likelihood-ratio systems: ranking the stars”. In: *Law, Probability and Risk* 22.1 (May 2023), mgad005. ISSN: 1470-8396. DOI: 10.1093/lpr/mgad005. eprint: https://academic.oup.com/lpr/article-pdf/22/1/mgad005/50984594/mgad005.pdf. URL: https://doi.org/10.1093/lpr/mgad005.

[22] Peter Vergeer et al. “Numerical likelihood ratios outputted by LR systems are often based on extrapolation: When to stop extrapolating?” In: *Science & Justice* 56 (June 2016). DOI: 10.1016/j.scijus.2016.06.003.

A

Mathematical derivations

A.1. Derivation of the joint normal distributions

In this appendix section we show the derivation for the joint multivariate normal distributions for the trace-trace scenario and trace-reference scenario assuming same or different source scenario. For this we use a well known characterization of multivariate normal distributions.

Theorem 1. *Let* $\mathbf{X} \in \mathbb{R}^d$ *be multivariately distributed with mean* μ *and a positive definite* $d \times d$ *covariance matrix* Σ*. Then there exists a matrix* $\mathbf{A} \in \mathbb{R}^{d \times k}$ *that satisfies* $\mathbf{A}\mathbf{A}^T = \Sigma$*, and we can write*

$$X = \mu + \mathbf{A}\mathbf{Z}$$

Where $\mathbf{Z} \in \mathbb{R}^k$ *is itself a* k*-dimensional vector, multivariate normally distributed,* $\mathbf{Z} \sim MVN(0, I_k)$ *We can also write* $\mathbf{Z}$ *as* $\mathbf{Z} = (Z_1, \ldots, Z_k)$ *with each* $Z_i \sim N(0, 1)$ *being independent standard normals.*

Joint pair distribution in a common source scenario

In the two level model, we have that each measurement X in our dataset is distributed as a multivariate normal, conditional on the source mean μ from which the measurement is drawn and within source covariance matrix $\Sigma_w = \sigma_w^2 I$. In particular, we can rewrite X using the theory and noting that we have matrix $\mathbf{A} = \sigma_w I$ that satisfies $\mathbf{A}\mathbf{A}^T = \Sigma_w$. We can thus write the conditional distribution for X in terms of μ and σ_w:

$$X|\mu \sim MVN(\mu, \Sigma_w)$$
$$X|\mu \overset{d}{=} \mu + \sigma_w I \mathbf{Z}_{\mathbf{w},\mathbf{X}} = \mu + \sigma_w \mathbf{Z}_{\mathbf{w},\mathbf{X}}$$

Here we denote $\mathbf{Z}_{\mathbf{w},\mathbf{X}}$ as the random vector that determine the randomness in the within source deviation for measurement X. Similarly, we denote $Z_{b,X}$ to denote the random vector that contains the standard normals that capture the between source deviation. Note that if we assume a trace X and another trace Y to have the same source, we have $\mathbf{Z}_{\mathbf{b},\mathbf{X}} = \mathbf{Z}_{\mathbf{b},\mathbf{Y}}$.

Note we also have that the source mean follows its own distribution: $\mu \sim MVN(\mathbf{0}, \Sigma_b)$, with $\Sigma_b = \sigma_b I$. Using the theorem on the fact that μ is also MVN, and noting that a matrix A that satisfies the theorem is $\sigma_b I$, we can rewrite:

$$\mu = \mathbf{0} + \sigma_b I \mathbf{Z}_{\mathbf{b}} = \sigma_b \mathbf{Z}_{\mathbf{b}}$$

From this fact we can write down the singular distribution for a trace evidence X and other trace evidence Y, without a reference to the source mean:

$$X = \sigma_b \mathbf{Z}_{\mathbf{b},\mathbf{X}} + \sigma_w \mathbf{Z}_{\mathbf{w},\mathbf{X}}$$
$$Y = \sigma_b \mathbf{Z}_{\mathbf{b},\mathbf{Y}} + \sigma_w \mathbf{Z}_{\mathbf{w},\mathbf{Y}}$$

Now to derive the corresponding means and covariance matrices as written down in 3.1. It is clear that the mean for both X and Y is the zero vector, independent of whether or not they originate from the same source or not.

To derive the covariance matrix for X, we compute between each of its elements the covariance $Cov(X_i, X_j), i, j = 1, \ldots, d$. We have

$$\begin{aligned} Cov(X_i, X_j) &= Cov(\sigma_b Z_{b,X,i} + \sigma_w Z_{w,X,i}, \quad \sigma_b Z_{b,X,j} + \sigma_w Z_{w,X,j}) \\ &= \begin{cases} \sigma_b^2 + \sigma_w^2 & , i = j \\ 0 & , i \neq j \end{cases} \end{aligned}$$

We thus get that the covariance matrix of X equals $(\sigma_b^2 + \sigma_w^2)I$, or in particular $Var(\mathbf{X}) = \Sigma_b + \Sigma_w$. Since Y has exactly the same distribution, we get the same covariance matrix for Y.

The only difference in the two distributions in 3.1 is in the covariance matrix between X and Y. That depends on the assumption if X and Y are from the same source. If they are from the same source, then we have that $Z_{b,X} = Z_{b,Y}$, and there will be non-zero covariance. In that case we have for $Cov(\mathbf{X}, \mathbf{Y})$ as elements:

$$\begin{aligned} Cov(X_i, Y_j) &= Cov(\sigma_b Z_{b,X,i} + \sigma_w Z_{w,X,i}, \quad \sigma_b Z_{b,Y,j} + \sigma_w Z_{w,Y,j}) \\ &= Cov(\sigma_b Z_{b,X,i} + \sigma_w Z_{w,X,i}, \quad \sigma_b Z_{b,X,j} + \sigma_w Z_{w,Y,j}), \text{using } Z_{b,X} = Z_{b,Y} \\ &= \begin{cases} \sigma_b^2 & , i = j \\ 0 & , i \neq j \end{cases} \end{aligned}$$

In that case we end up with $\sigma_b^2 I = \Sigma_b$ for $Cov(\mathbf{X}, \mathbf{Y})$. This yields the joint distribution for the H_1-case for the common source scenario:

$$(\mathbf{X}, \mathbf{Y})|H_{1,c} \sim MVN(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \Sigma_b + \Sigma_w & \Sigma_b \\ \Sigma_b & \Sigma_b + \Sigma_w \end{bmatrix})$$

If they are from different sources, then $Z_{b,X}$ and $Z_{b,Y}$ are independent and will there will be a zero covariance matrix. The only covariance between the elements of X and Y comes from whether or not they are from the same source. We get then for the H_2-case the distribution:

$$(\mathbf{X}, \mathbf{Y})|H_{2,c} \sim MVN(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \Sigma_b + \Sigma_w & 0 \\ 0 & \Sigma_b + \Sigma_w \end{bmatrix})$$

Joint pair distribution in a specific source scenario

In this scenario the mean for the now known reference Y is fixed. The distribution of trace X now depends on the case if we assume the trace to come from the same source as reference Y. In the H_1-case we now have for our trace X and Y the following distributions:

$$\begin{aligned} X &= \mu_{\mathbf{s}} \mathbf{s} + \sigma_w \mathbf{Z_{w,X}} \\ Y &= \mu_{\mathbf{s}} \mathbf{s} + \sigma_w \mathbf{Z_{w,Y}} \end{aligned}$$

We then get as covariance matrix for X:

$$\begin{aligned} Cov(X_i, X_j) &= Cov(\mu_{\mathbf{s}} \mathbf{s} + \sigma_w Z_{w,X,i}, \quad \mu_{\mathbf{s}} \mathbf{s} + \sigma_w Z_{w,X,j}) \\ &= \begin{cases} \sigma_w^2 & , i = j \\ 0 & , i \neq j \end{cases} \end{aligned}$$

So we get $Var(\mathbf{X}) = \Sigma_w$. Similar reasoning yields $Var(\mathbf{Y}) = \Sigma_w$.

For the covariance matrix between X and Y we have

$$\begin{aligned} Cov(X_i, Y_j) &= Cov(\mu_{\mathbf{ss}} + \sigma_w Z_{w,X,i}, \quad \mu_{\mathbf{ss}} + \sigma_w Z_{w,Y,j}) \\ &= Cov(\sigma_w Z_{w,X,i}, \quad \sigma_w Z_{w,Y,j}) \\ &= 0 \end{aligned}$$

We therefore get as distribution for the H_1 case for specific source scenario:

$$(\mathbf{X}, \mathbf{Y})|H_{1,s} \sim MVN(\begin{bmatrix} \mu_{\mathbf{X}} \\ \mu_{\mathbf{Y}} \end{bmatrix}, \begin{bmatrix} \Sigma_w & 0 \\ 0 & \Sigma_w \end{bmatrix})$$

In the H_2-case we now have for our trace X and Y the following distributions:

$$\begin{aligned} X &= \sigma_b \mathbf{Z_{b,X}} + \sigma_w \mathbf{Z_{w,X}} \\ Y &= \mu_{\mathbf{ss}} + \sigma_w \mathbf{Z_{w,Y}} \end{aligned}$$

We then get instead a different covariance matrix for X, we get just like for the common scenario a covariance matrix $Var(\mathbf{X}) = \Sigma_b + \Sigma_w$. We therefore get as distribution for the H_2 case for specific source scenario:

$$(\mathbf{X}, \mathbf{Y})|H_{2,s} \sim MVN(\begin{bmatrix} 0 \\ \mu_{\mathbf{ss}} \end{bmatrix}, \begin{bmatrix} \Sigma_b + \Sigma_w & 0 \\ 0 & \Sigma_w \end{bmatrix})$$

A.2. Derivations of the distribution of the scores

Common source

Under H_1 we have that measurements X and Y are from the same source. They follow a multivariate normal distribution with the same mean μ_X, where μ_X is the mean generated for that source. In particular, their marginals X_i and Y_i are distributed according to $N(\mu_{X,i}, \sigma_w^2)$.

Where therefore have that $X_i - Y_i \sim N(0, 2\sigma_w^2)$, and so we get that $|X_i - Y_i|$ follows a half normal distribution, which is a special case of the folded normal distribution with mean and variance [19]:

$$\begin{aligned} \mathbb{E}[|X_i - Y_i||H_1] &= \sqrt{\frac{2}{\pi}} \cdot \sigma_w \sqrt{2} = \frac{2\sigma_w}{\sqrt{\pi}} \\ Var(|X_i - Y_i||H_1) &= 2\sigma_w^2 - \frac{4\sigma_w^2}{\pi} = \sigma_w^2 \left(2 - \frac{4}{\pi}\right) \end{aligned}$$

We can use linearity of expectation and variance (for independent random variables) to get the moments of our score for H_1:

$$\begin{aligned} \mathbb{E}[\delta(X,Y)|H_{1,c}] &= d \cdot \frac{2\sigma_w}{\sqrt{\pi}} \\ Var[\delta(X,Y)|H_{1,c}] &= d \cdot \sigma_w^2 \left(2 - \frac{4}{\pi}\right) \end{aligned}$$

Under H_2 we have that measurements X and Y are from a different source. But now we can write $X_i \sim N(0, \sigma_b^2 + \sigma_w^2)$, and we also have the same distribution for Y_i. Then we get instead that $X_i - Y_i \sim N(0, 2(\sigma_b^2 + \sigma_w^2))$. Redoing the analysis gives us the expectation and variance for the score under H_2

$$\begin{aligned} \mathbb{E}[\delta(X,Y)|H_{2,c}] &= d \cdot \frac{2\sqrt{\sigma_b^2 + \sigma_w^2}}{\sqrt{\pi}} \\ Var[\delta(X,Y)|H_{2,c}] &= d \cdot (\sigma_b^2 + \sigma_w^2) \left(2 - \frac{4}{\pi}\right) \end{aligned}$$

An explicit formula for the density of the scores would require to do a convolution of the densities of the folded normal distribution and is generally quite complex to write down.

Key thing to note however is that this described the distribution of the scores if we would not preprocess using a Z-score normalisation. However since we preprocess the actual used marginals differ a bit. They still follow a folded normal distribution, but are rescaled a bit since we transform each measurement to have mean 0 and unit variance.

However the same general principles apply after preprocessing, which are that the scores depend directly on the between and within deviation. Another important takeaway is here that both the expectation and variance of the score scale linearly in the dimension due to assuming equal σ_w for each source.

Specific source

Under H_1 we have that measurements X and Y are from the same specific source. The measurements again follow a multivariate normal distribution with the same, but now specific, source mean μ_{ss}. In particular, their marginals X_i and Y_i are distributed according to $N(\mu_{ss}, \sigma_w^2)$, and so we get again that $X_i - Y_i \sim N(0, 2\sigma_w^2)$, since we assumed equal variance per source. The H_1 score distribution is then the same for the specific source case as for the common source case, and so we get

$$\mathbb{E}[\delta(X,Y)|H_{1,s}] = d \cdot \frac{2\sigma_w}{\sqrt{\pi}}$$
$$Var[\delta(X,Y)|H_{1,s}] = d \cdot \sigma_w^2 \left(2 - \frac{4}{\pi}\right)$$

For H_2 we now have that the scores depend on the value of the mean of the specific source. Our trace features X_i follow the same distribution, $X_i \sim N(0, \sigma_b^2 + \sigma_w^2)$, but for the reference Y we now have $Y_i \sim N(\mu_{ss,i}, \sigma_w^2)$. Therefore we have that $X_i - Y_i \sim N(-\mu_{ss}, \sigma_b^2 + 2\sigma_w^2)$, and so $|X_i - Y_i|$ also follows a folded normal distribution [19] but with a mean and variance that depend on the location of the specific source:

$$\mathbb{E}[|X_i - Y_i||H_{2,s}] = \sum_{i=1}^{d} \sqrt{\frac{2(\sigma_b^2 + 2\sigma_w^2)}{\pi}} e^{-\frac{-\mu_{ss,i}^2}{2(\sigma_b^2 + 2\sigma_w^2)}} + \mu_{ss}\left[1 - 2\Phi\left(-\frac{\mu_{ss,i}}{2(\sigma_b^2 + 2\sigma_w^2)}\right)\right] : \mu_{\delta,H_{2,s}}$$
$$Var(|X_i - Y_i||H_{2,s}) = \mu_{ss}^2 + 2(\sigma_b^2 + 2\sigma_w^2) - \mu_{\delta,H_{2,s}}^2 \qquad := \sigma_{\delta,H_{2,s}}^2$$

Our expectation and variance for the score then becomes:

$$\mathbb{E}[\delta(X,Y)|H_{2,s}] = d \cdot \mu_{\delta,H_{2,s}}$$
$$Var[\delta(X,Y)|H_{2,s}] = d \cdot \sigma_{\delta,H_{2,s}}^2$$

We note that now the expectation of the distance depends on the location of the specific source. If the mean of the specific source is nearer to 0, we get something similar to the common source distribution. We only have one level of randomness less since we do not have the added variation of σ_b in the specific source case since we have a fixed specific source.

If our specific source mean is larger however, then the exponential term quickly goes to 0 while the linear factor stays. This makes intuitive sense as well: If we have a specific source mean $\mu_s s$ that increases away from 0, then the average distance between an alternative source and the specific source should increase. Therefore we expect higher scores for H_2 if we have a rarer specific source.

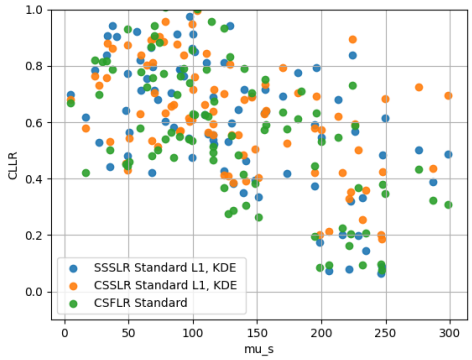
B

Additional figures

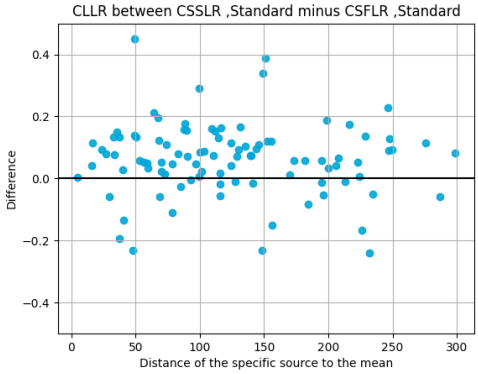
B.1. Identical sources

B.1.1. Little specific source data, small background population

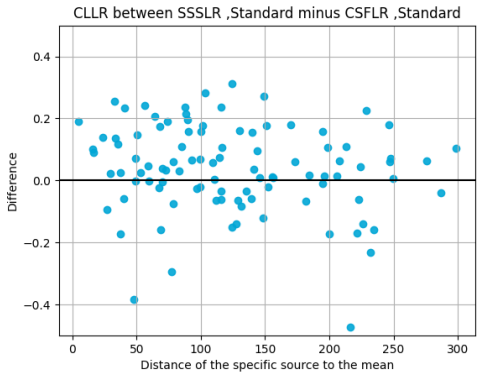
$\sigma_w = 50$



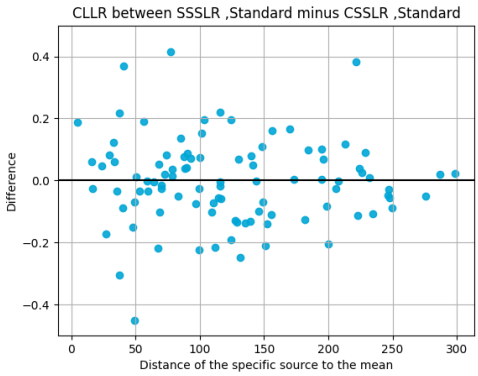
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



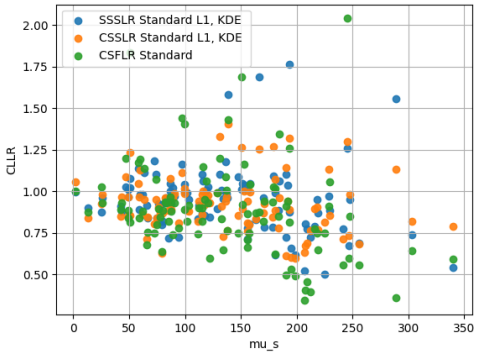
(c) difference between SSSLR and CSFLR



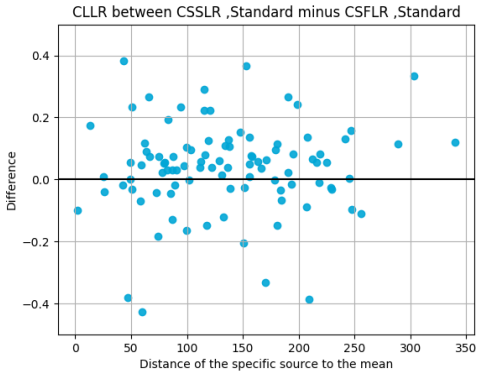
(d) difference between SSSLR and CSSLR

Figure B.1: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 50$

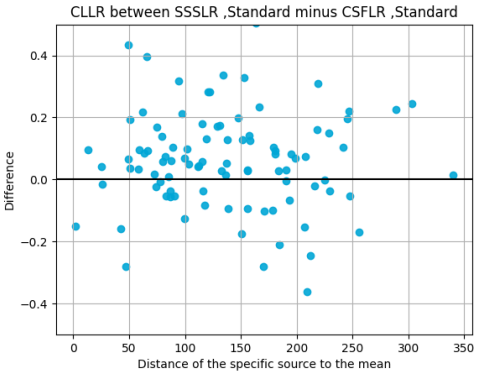
$\sigma_w = 100$



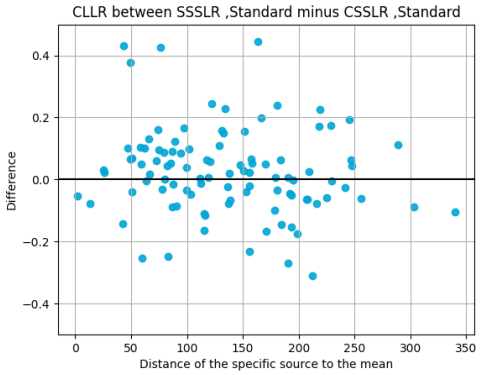
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR

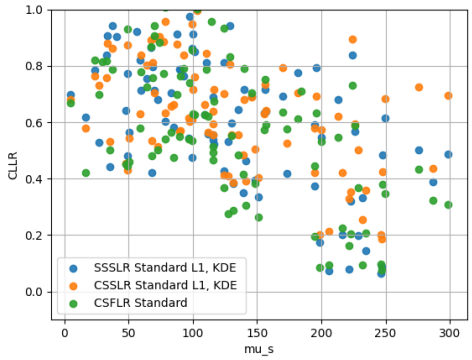


(d) difference between SSSLR and CSSLR

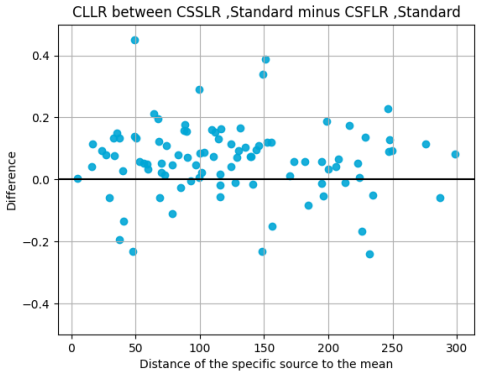
Figure B.2: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 100$

B.1.2. Little specific source data, large background population

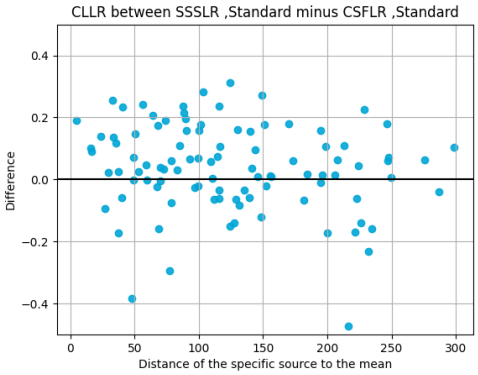
$\sigma_w = 50$



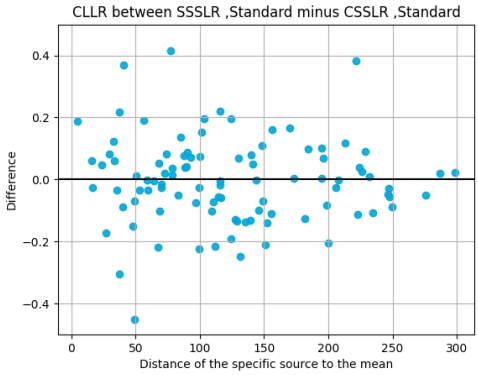
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



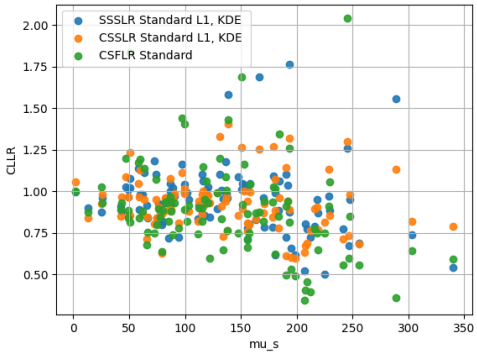
(c) difference between SSSLR and CSFLR



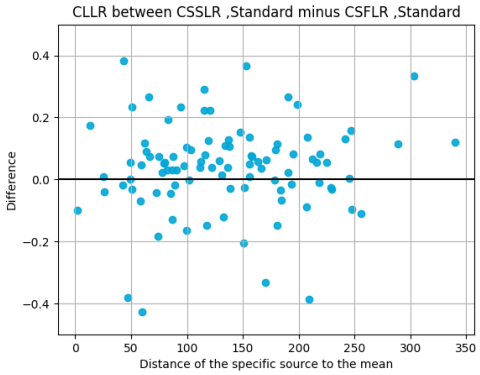
(d) difference between SSSLR and CSSLR

Figure B.3: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 50$

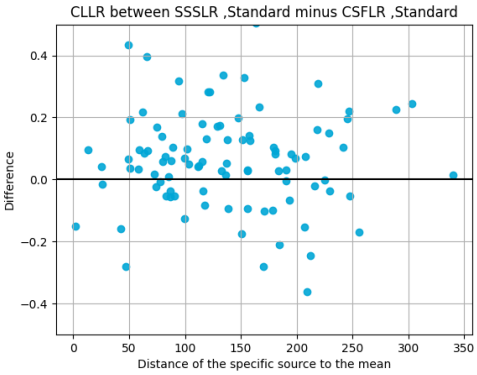
$\sigma_w = 100$



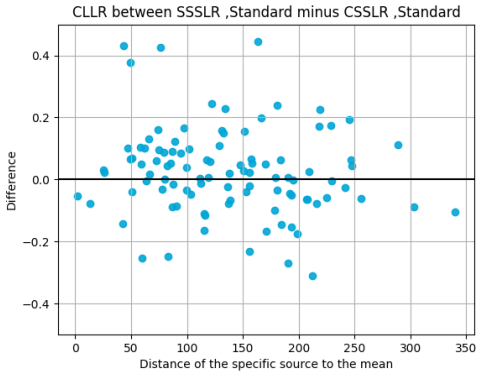
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR

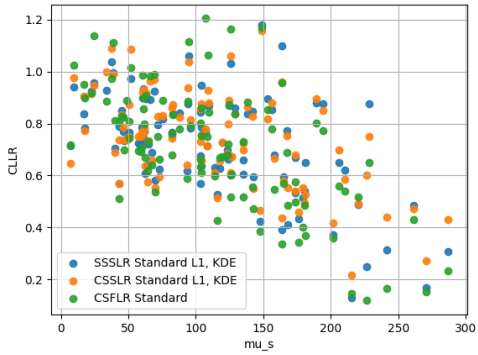


(d) difference between SSSLR and CSSLR

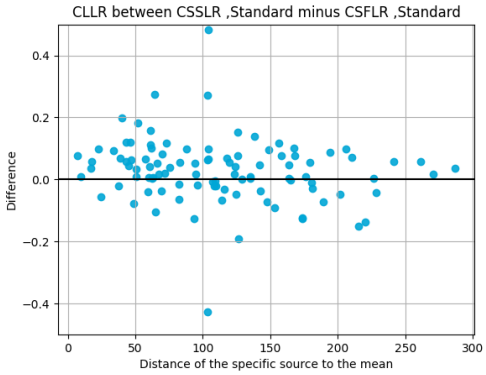
Figure B.4: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 100$

B.1.3. Sufficient specific source data, small background population

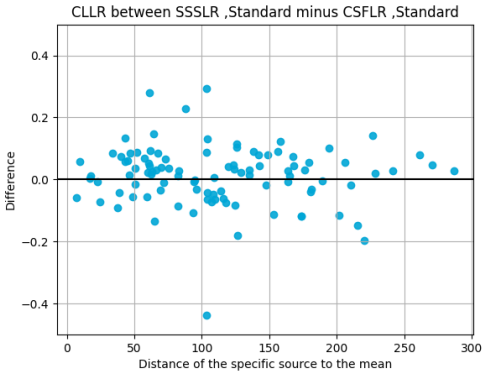
$\sigma_w = 50$



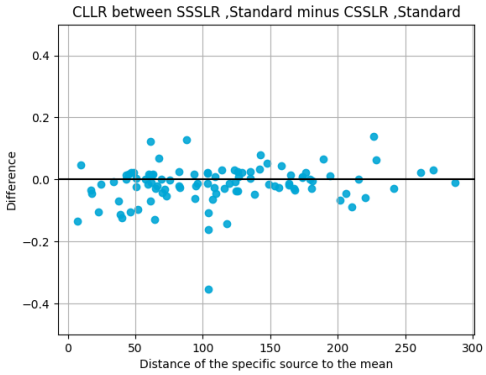
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



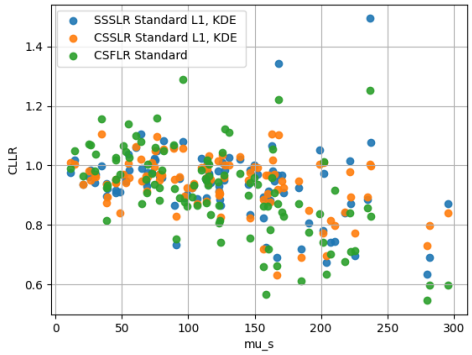
(c) difference between SSSLR and CSFLR



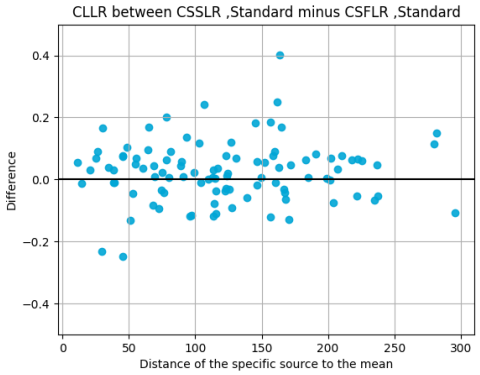
(d) difference between SSSLR and CSSLR

Figure B.5: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 50$

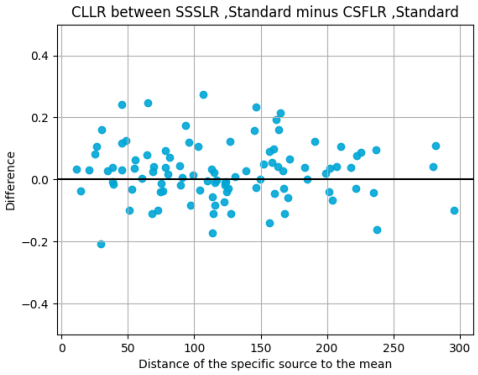
$\sigma_w = 100$



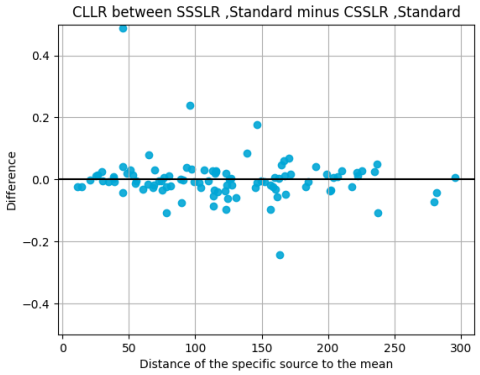
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR

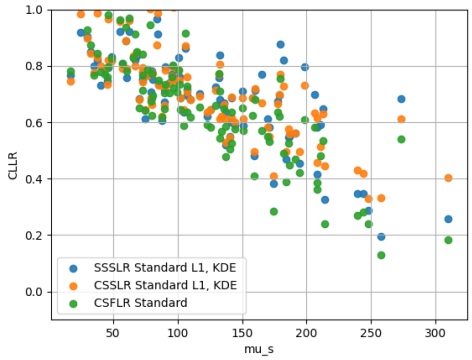


(d) difference between SSSLR and CSSLR

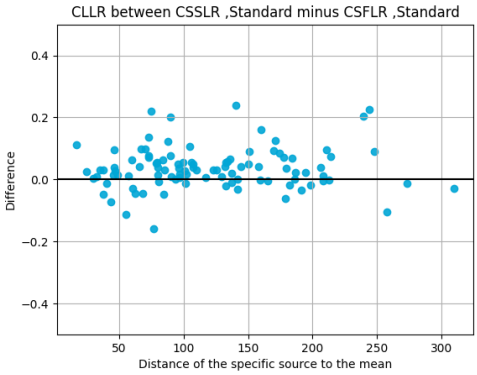
Figure B.6: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 100$

B.1.4. Sufficient specific source data, large background population

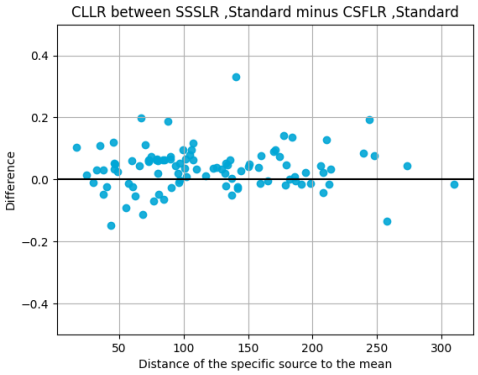
$\sigma_w = 50$



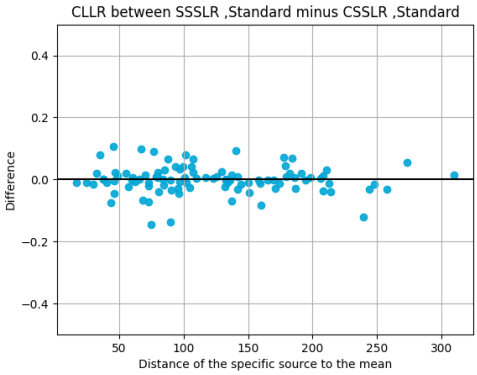
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



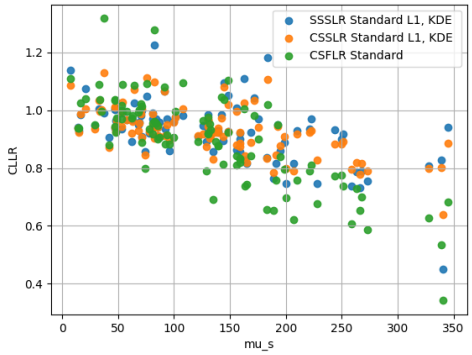
(c) difference between SSSLR and CSFLR



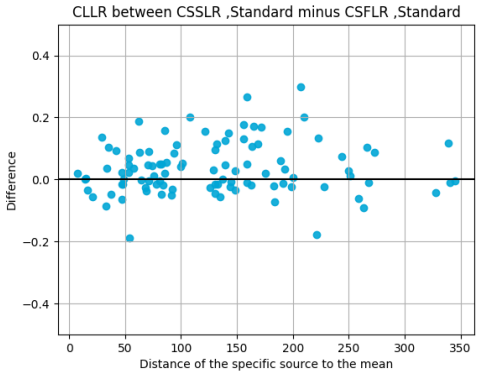
(d) difference between SSSLR and CSSLR

Figure B.7: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 50$

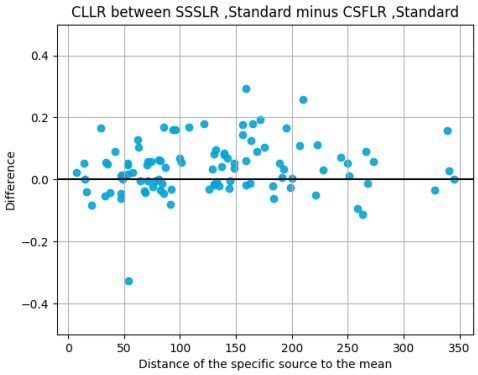
$\sigma_w = 100$



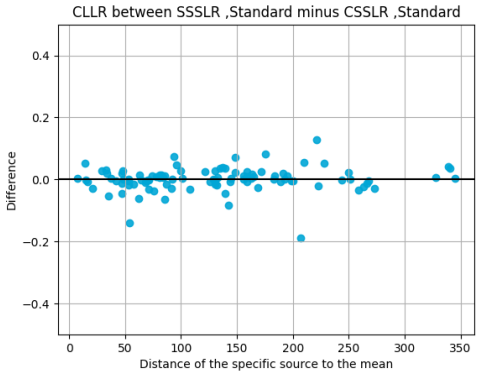
(a) Scatterplot of CLLRs



(b) difference between CSSLR and CSFLR



(c) difference between SSSLR and CSFLR

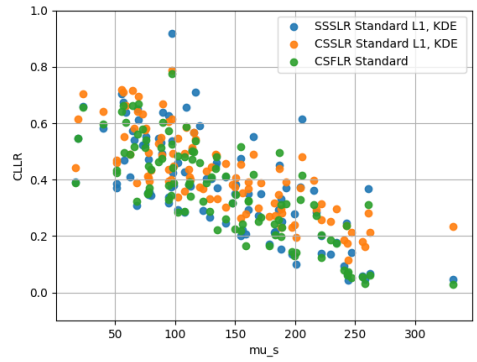


(d) difference between SSSLR and CSSLR

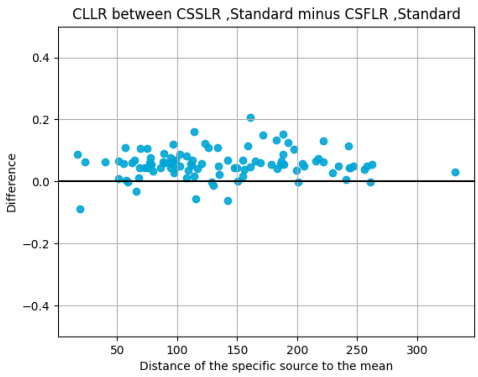
Figure B.8: CLLRS and differences between systems when each source is identical and has a $\sigma_w = 100$

B.2. Non identical sources

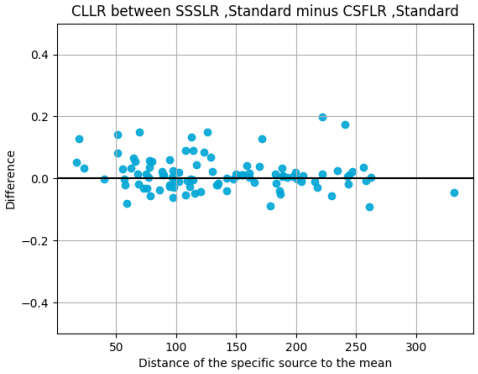
B.2.1. Average specific source deviation



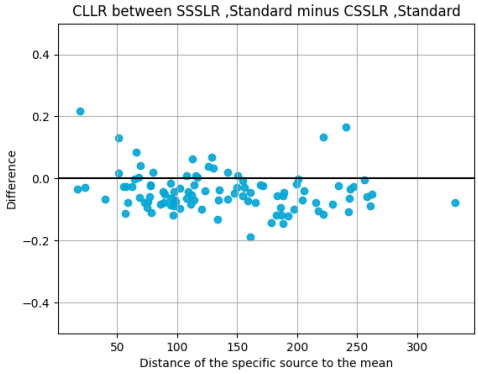
(a) Scatterplot of CLLRs



(b) Difference between CSSLR and CSFLR

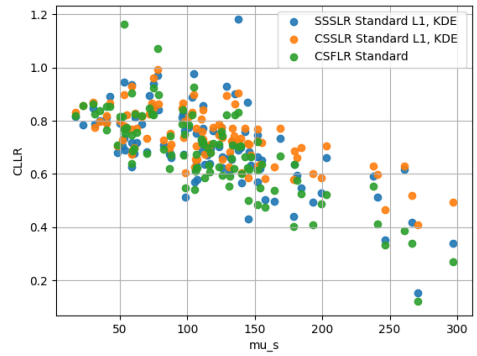


(c) Difference between SSSLR and CSFLR

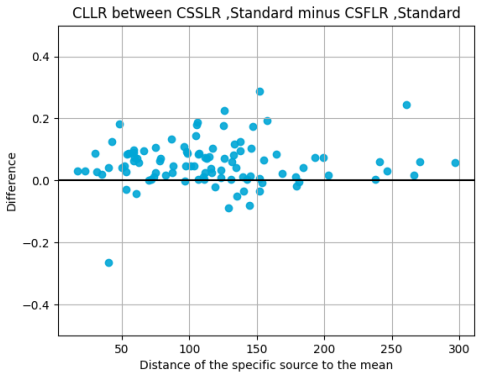


(d) Difference between SSSLR and CSSLR

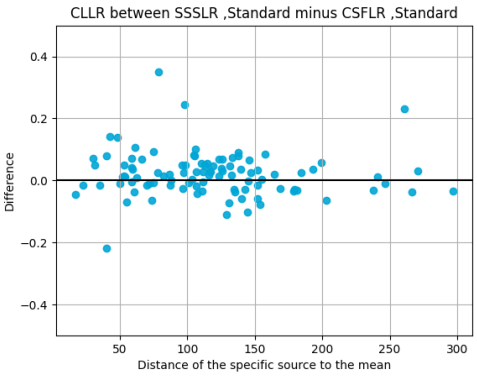
Figure B.9: CLLRS and difference between systems when for each alternative source we have in the population we have $\sigma_w \sim U[1, 50]$, but $\sigma_{w,ss}$ is equal to the midpoint of the interval



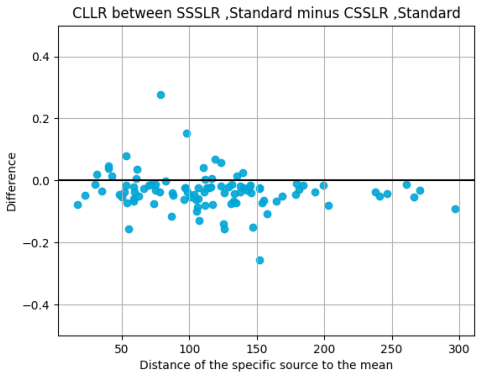
(a) Scatterplot of CLLRs



(b) Difference between CSSLR and CSFLR



(c) Difference between SSSLR and CSFLR

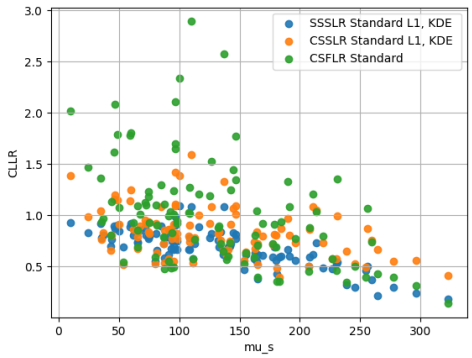


(d) Difference between SSSLR and CSSLR

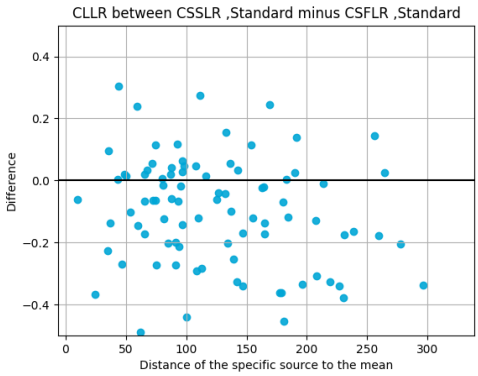
Figure B.10: CLLRS and difference between systems when for each alternative source we have in the population we have $\sigma_w \sim U[1, 100]$, but $\sigma_{w,ss}$ is equal to the midpoint of the interval

B.2.2. Large specific source deviation

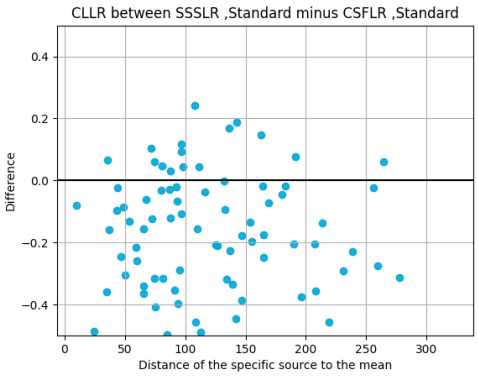
$\sigma_w \sim U[1, 50], \sigma_{w,ss} = 50$



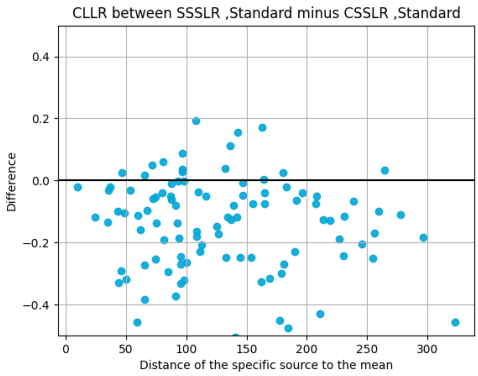
(a) Scatterplot of CLLRs



(b) Difference between CSSLR and CSFLR



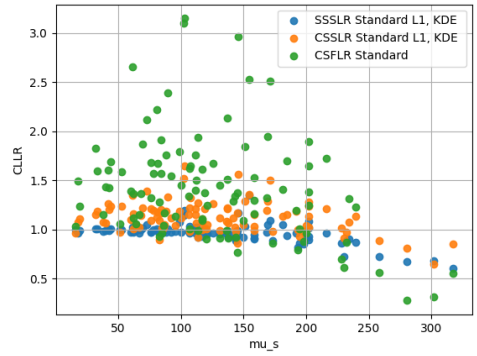
(c) Difference between SSSLR and CSFLR



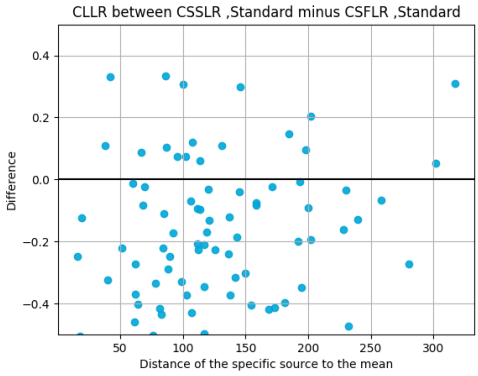
(d) Difference between SSSLR and CSSLR

Figure B.11: CLLRS and difference between systems when for each alternative source we have in the population we have $\sigma_w \sim U[1, 50]$, but $\sigma_{w,ss}$ is equal to the endpoint of the interval, $\sigma_{w,ss} = 50$

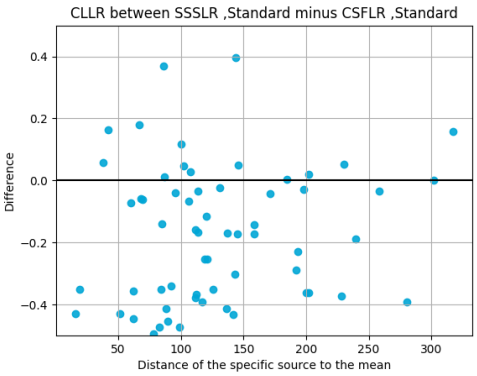
$\sigma_w \sim U[1, 100], \sigma_{w,ss} = 100$



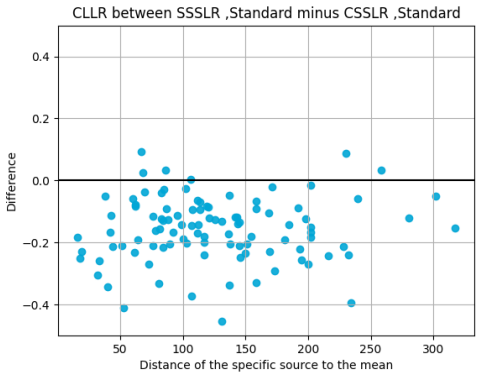
(a) Scatterplot of CLLRs



(b) Difference between CSSLR and CSFLR



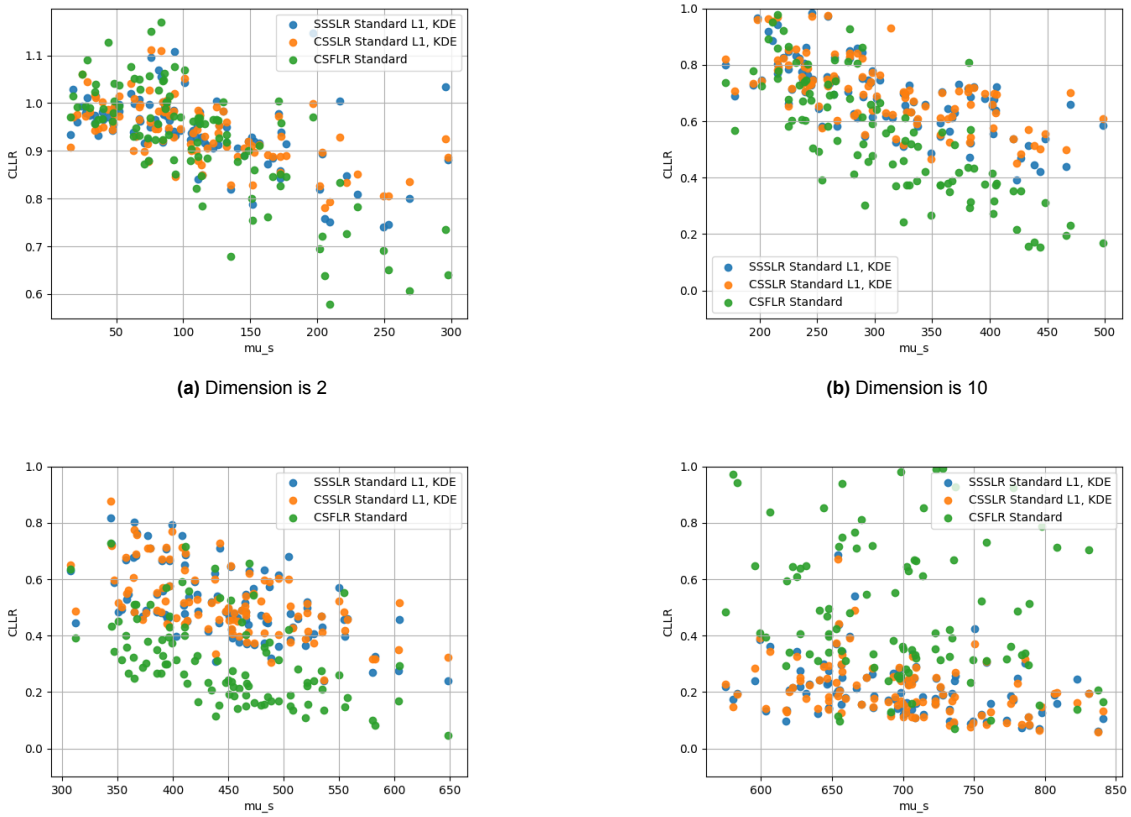
(c) Difference between SSSLR and CSFLR



(d) Difference between SSSLR and CSSLR

Figure B.12: CLLRS and difference between systems when for each alternative source we have in the population we have $\sigma_w \sim U[1, 100]$, but $\sigma_{w,ss}$ is equal to the endpoint of the interval, $\sigma_{w,ss} = 100$

B.3. High dimensional LR systems
B.4. Highest within-variability



(a) Dimension is 2

(b) Dimension is 10

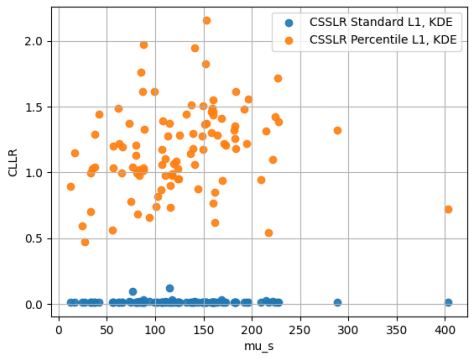
(c) Dimension is 20

(d) Dimension is 50

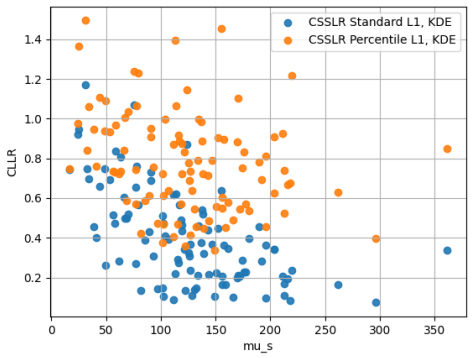
Figure B.13: Performance of the likelihood ratio system as the dimension increases for a higher within-variability of $\sigma_w = 100$

B.5. Percentile rank

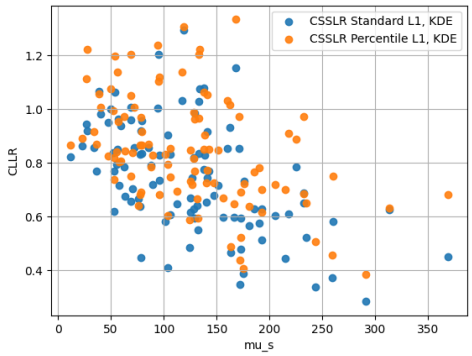
CSSLR performances for a small background population



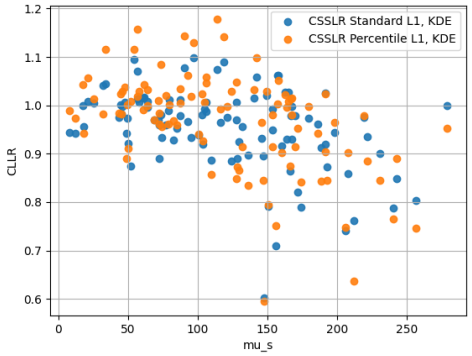
(a) Comparison where $\sigma_w = 1$



(b) Comparison where $\sigma_w = 25$



(c) Comparison where $\sigma_w = 50$



(d) Comparison where $\sigma_w = 100$

Figure B.14: The C_{llr} for the CSSLR using a standard or a percentile rank preprocessor for various within deviations, given a small background population