A Study on Optimization of Security and Convenience in Biometric Identification

（ID レス生体認証における安全性と利便性の最適化に関する研究）

by

Takao Murakami

村上 隆夫

A Doctor Thesis

博士論文

Thesis Supervisor: Kanta Matsuura

指導教員：松浦 幹太

December 13, 2013

Acknowledgement

First of all, I would like to express my gratitude to my supervisor, Associate Professor Kanta Matsuura, for his guidance and encouragement during my doctoral course. He taught me a great deal of things about the field of information security. He also taught me the attitude towards research from his meaningful and thoughtful words. I would also like to thank my advisors, Professor Nobuaki Minematsu and Professor Jun Adachi. They gave me a number of valuable comments and feedbacks on my research results in this thesis and the presentation of them.

I would like to express my gratitude to Kenta Takahashi, my senior colleague in Yokohama Research Laboratory, Hitachi, Ltd. He taught me numerous things about biometrics, and supported my work. Many of the ideas in this thesis were derived from frank and frequent discussions with him. I am also really grateful to my boss Masahiro Mimura. He introduced me to the field of biometrics, and supported me in a variety of ways. I would also like to thank Susumu Serita and Yasuhiro Fujii for their discussions on some part of my work.

I would like to thank the present and past Matsuura-lab members (including external meeting visitors): Takuro Hosoi, Bongkot Jenjarrussakul, Satsuya Ohata, Masahito Ishizaka, Kenichi Yokote, Feng Fei, Teppei Takagi, Toshinori Usui, Takashi Kitagawa, Ken Ichikawa, Takayoshi Hojo and Andreas Gutmann, for many discussions on my research results. Their presentations at the Matsuura-lab meeting were also very informative for me.

I have also benefited from the advice, comments, and encouragement of many other people, including but not limited to: Kazuo Takaragi, Seiichi Susaki, Yoshiaki Isobe, Osamu Takata, Shinji Hirata, Hideitsu Hino, Yosuke Kaga, Yukiko Matsubara, Camille Vuillaume, Jun Nakanishi, Naoya Kanda, Naoto Miura, Tetsushi Ohki, and Seira Hidano. I really appreciate the support of them.

Finally, I would like to thank my family for their continuous encouragement and support.

Abstract

Biometric identification, which recognizes an individual based only on physiological or behavioral characteristics, is nowadays used for commercial applications such as computer login, physical access control, and time and attendance management. It has a potential to provide the best authentication solution with regard to security and convenience because it does not require a user ID, password, nor card but recognizes a user based on his/her biometrics (i.e. something you are). In biometric identification, however, the following factors cause problems with security and convenience, and become serious as the number of enrollees increases: (1) false accepts; (2) wolves and lambs who cause false accepts against many others; (3) false rejects; (4) the number of biometric inputs; (5) response time. False accepts, wolves, and lambs are factors which affect security, while false rejects, the number of inputs, and response time are factors which affect convenience. They are related to each other, and have prevented biometric identification from being applied to large-scale applications. The goal of this study is to optimize security and convenience in biometric identification in terms of these factors.

Firstly, we make an attempt to optimize false accepts, false rejects, and the number of inputs. We focus on MSPRT's (Multi-hypothesis Sequential Probability Ratio Tests), multi-hypothesis tests which can minimize the average number of observations, and propose two sequential fusion schemes in identification: the PPSI (Posterior Probability-based Sequential Identification) scheme and the LRSI (Likelihood Ratio-based Sequential Identification) scheme. The PPSI scheme is based on MSPRT and can minimize the average number of inputs (referred to as ANI), while the LRSI scheme is a simpler one which can be carried out quickly. Then we prove that the LRSI scheme can also minimize ANI by proving that this scheme is equivalent to MSPRT. We also discuss the conditions to achieve the optimality, and show the effectiveness of the two schemes through experimental evaluation using the NIST BSSR1 Set1, a multi-modal score dataset (one face and two fingerprints).

Secondly, we make an attempt to further optimize response time. To this end, we turn our attention to metric space indexing methods which have been developed in the area of similarity search, and focus on pseudo-score based indexing schemes which compute, for each object in the database, a pseudo-score which is easily computed and highly relevant to a score (distance or similarity), and compute scores in order of the pseudo-score. We first propose the PPS (Posterior Probability-based Search) scheme which normalizes each pseudo-score to the posterior probability of being in the answer to the range query. We proved that the PPS scheme has an optimal property with regard to the number of score computations and the expected number of retrieval errors. We also showed that it outperforms the two state-of-the-art schemes: the standard pivot-based indexing scheme and the permutation-based indexing scheme, through experimental evaluation using various kinds of datasets from the Metric Space Library. We then make an attempt to combine metric space indexing and sequential fusion, and propose the PPSS (Posterior Probability-based Sequential Search) scheme, a modification of the PPS scheme to use not only pseudo-scores at the current input but past pseudo-scores and scores as information sources. We also propose a technique which optimizes the number of pivots (biometric templates selected from the database to compute pseudo-scores) with regard to retrieval errors. We demonstrate that our proposals significantly reduce the number of score computations of the PPSI scheme while keeping false accepts, false rejects, and the number of inputs, using a large-scale multi-modal dataset (1800 enrollees; one face and two fingerprints) obtained by combining the NIST BSSR1 Set3 and the CASIA-FingerprintV5.

Finally, we attempt to optimize the trade-off between security against wolves and lambs and convenience in terms of the number of inputs and false rejects. To clarify our target, we first introduce a taxonomy which classifies wolves into three categories (zero-effort wolves, non-adaptive spoofing wolves, and adaptive spoofing wolves) and lambs into two categories (zero-effort lambs and spoofing lambs). Then, we propose the MLRSV (Minimum Likelihood Ratio-based Sequential Verification) scheme as a sequential fusion scheme in verification. We prove that this scheme has security against wolves and lambs, except for adaptive spoofing wolves, and minimizes ANI and false rejects under some conditions. We also discuss the conditions to achieve the security and optimality, and propose an input order decision scheme based on the KL (Kullback-Leibler) divergence to further reduce ANI in the case where the KL divergence differs from one modality to another. We finally demonstrate the effectiveness of our proposals using a multi-modal (one face and eight fingerprints) dataset obtained by combining the NIST BSSR1 Set3 and the CASIA-FingerprintV5.

Contents

Chapter 1

Introduction

Contents

1.1 Background

The information society has grown rapidly over the past two decades. Since 1995, the number of Internet users has increased dramatically and reached 2.7 billion in 2013 [46]. Nowadays various devices such as PCs, smartphones, tablets, and sensors are connected to information networks, and various services such as healthcare, education, games, music, banking, and payments are provided over the networks. As the physical world is more and more connected to the digital world, personal authentication, which associates an identity with an individual, plays an increasingly crucial role in protecting information or physical assets from unauthorized access.

The personal authentication methods recognize an individual using some information which can be classified into the following three factors:

- **Knowledge (i.e. something you know):** secret information such as passwords, passphrases, and personal identification numbers (PINs).
- **Possession (i.e. something you have):** physical possessions such as ID cards, smart cards, physical keys, mobile phones, and passports.
- **Inherence (i.e. something you are):** physiological characteristics (such as fingerprint, face, iris, vein, retina, palmprint, and DNA) or behavioral characteristics (such as voice, signature, keystroke, and gait). They are also known as *biometrics*.

The first and second factors are now widely used (e.g. password to login to a computer, ID card to enter a room). However, the authentication methods based on these factors have problems with security and convenience. The first factor such as passwords can be guessed or cracked. To prevent the guessing attack, a user needs to use the password which is hard to remember and change it periodically. The second factor such as cards can be stolen, misused, forgotten, or lost. The essential problem common to these factors is that these methods cannot distinguish a genuine user from an impostor who illegally acquired the knowledge or possession.

We note here that security and convenience are related to each other, and a security solution which lacks convenience can eventually fail to strengthen, or even weaken, the security of the system. This contradiction is known as the *security vs. convenience dilemma* [18]. For example, password expiration mechanisms which require users to change passwords frequently often induce a counterproductive behavior such as writing down passwords on a piece of paper [1]. Many bank customers also write their PINs on their cards, and thus jeopardize two-factor authentication which uses both knowledge and possession [99]. This kind of problem has long been discussed in the area of usable security [9, 83]. The important thing is that we need to design a system which is simultaneously secure and convenient.

Biometric authentication associates an identity with inherence (i.e. something you are), the third factor. Since biometrics is not forgotten unlike passwords and is much harder to steal than cards, a biometric authentication system can recognize an individual in a more convenient and secure manner. From an algorithm standpoint, the biometric system recognizes an individual using a similarity or distance, called *score*, between his/her biometric sample (referred to as a *query sample*) and a biometric feature enrolled in advance (referred to as a *template*). For example, the system computes the Hamming distance between iriscodes [27], the Euclidean distance between eigenfaces [114], or the percentage of matched minutiae [52] as a score, and makes a decision whether the individual is genuine or not by comparing the score to a predetermined threshold. Throughout this dissertation, we refer to a person who inputs a query sample as a *claimant*, and a person who enrolls a template as an *enrollee*.

Depending on the application context, a biometric authentication system can operate in either of the following two modes: *verification* and *identification* [14, 58]. In the verification mode, a claimant claims an identity via an ID number, a user name, or a smart card, and inputs his/her query sample. Then, the system computes a score between the query sample and a template corresponding to the claimed identity, and makes a decision whether the claimant is genuine or not. This type of biometric authentication is now used for various commercial applications such as ATM, payments, and computer login.

In the identification mode, a claimant does not explicitly claim his/her identity, and only inputs a query sample. Then, the system computes scores between the query sample and templates in the database (i.e. one-to-many matching), and outputs a candidate list of people whose templates are similar to the query sample. Biometric identification can be further classified into two categories: *negative identification* and *positive identification* [14, 58]. Negative identification determines whether a person is *not* in the database as he/she (explicitly or implicitly) claims. This type of identification has long been used for criminal investigation. In 1960's, the NIST engineers started a research of automated fingerprint identification at the request of the FBI [74]. The resulting AFISs (Automated Fingerprint Identification Systems) have been used worldwide, with a significant improvement of the fingerprint identification technology.

On the other hand, positive identification determines whether a person is in the database as he/she claims, and is nowadays used for commercial applications such as computer login, physical access control, and time and attendance management. In these applications, the positive identification system typically outputs at most one candidate (i.e. accepts as some enrollee or rejects as a non-enrollee) and provides service appropriate to the identified user. The advantage of positive identification over verification is its convenience: a claimant does not have to input an ID number nor present a card. For example, a finger-vein identification system only requires a claimant to input his/her finger on the sensor. Another example is remote biometric identification (e.g. face at a distance, iris at a distance) [113] which identifies a claimant at a distance. Since it does not require a claimant to perform any action to be recognized, and recognizes him/her based on "something you are" (an inherence factor), it even has a potential to provide the best solution with regard to both security and convenience.

However, positive identification has problems with security and convenience which differ from the ones of the knowledge-based or possession-based authentication and become serious as the number of enrollees increases. The problems have prevented positive identification from being applied to large-scale applications.

In the following, we describe the problems in detail (hereinafter, we will simply use the term "identification" to refer to positive identification):

- **Problems with security:**
  - **False accepts:** The biometric identification system can incorrectly accept a non-enrollee (a person not enrolled) as some enrollee, or accept an enrollee as another enrollee. Since these two types of identification errors are false accepts, they can cause security problems. Although these errors can be reduced by raising a threshold for a similarity (or lowering a threshold for a distance), this can cause the increase of false rejects: the identification errors in which the system incorrectly rejects an enrollee as a non-enrollee. That is, there is a trade-off between false accepts and false rejects, as also described later. The important thing is that false accepts in identification increase as the number of enrollees increases, since the number of those who can be incorrectly identified increases. In Section 3.2.1, we show that they can increase almost in proportion to the number of enrollees.
  - **Wolves and lambs:** It is known that different users have different degrees of accuracy in biometric authentication, and claimants and enrollees who cause false accepts against many others are referred to as *wolves* and *lambs*, respectively [29]. Although they cause security problems in both verification and identification, we consider they are particularly problematic in identification due to the following reasons: (1) In identification, wolves can cause many false accepts even if they do not intend to impersonate others. Since they know that they have such threatening biometrics as a result, the incidents may even encourage them to actively attack many databases to impersonate others. (2) Although lambs are a vulnerability in verification, they can be a threat in identification because they can make the system identify many claimants as them, and lose the availability of the system. As it is often said that the overall security of a system is determined by the *weakest link in the chain* [5], the overall security of the identification system can be determined by these animals. False accepts caused by the animals also increase with the increase of enrollees, and are in a trade-off relationship with false rejects. Recently, an artifact which causes many false accepts is also proposed [116].
- **Problems with convenience:**
  - **False rejects:** False accepts can be reduced by raising a threshold for a similarity (or lowering a threshold for a distance), as described above. However, this can cause the increase of false rejects in which the system incorrectly rejects an enrollee as a

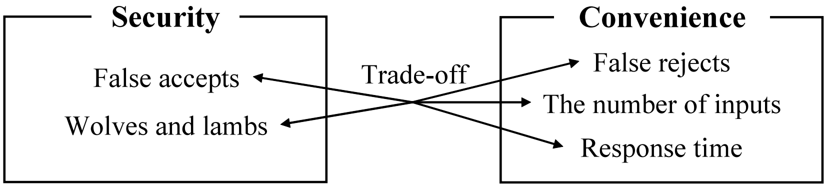


Figure 1.1: Factors which affect security and convenience in positive biometric identification. They become serious with the increase of enrollees.

non-enrollee, and make the system inconvenient. Since there is a trade-off between false accepts and false rejects, both of the problems can eventually become serious as the number of enrollees increases.

- **The number of inputs:** A false reject induces a retry attempt in which a claimant inputs the same biometrics again. Although this attempt may result in acceptance, it takes time and effort for him/her to input additional samples. That is, not only false rejects but the number of inputs are factors which cause inconvenience. To date, a considerable number of multi-modal biometric fusion schemes [57, 96, 97] which combine multiple sources of information (e.g. fingerprint, face and voice; index and middle fingers) have been proposed to reduce identification errors (false accepts and false rejects). However, these schemes can also cause the increase of the number of inputs. That is, there is a trade-off between false accepts, false rejects, and the number of inputs. Thus, the problem of the number of inputs can also become serious as the number of enrollees increases.
- **Response time:** As the number of enrollees increases, the number of score computations increases. Thus, the total response time also increases as a result, which makes the system inconvenient. There is also a trade-off between the response time and the identification accuracy in general. For example, a great number of classification or indexing methods [67] have been proposed to reduce the number of score computations while keeping a genuine score (i.e. a score between the same person), as described in Section 2.2 in detail. However, as the number of score computations decreases in these methods, the errors in which the system fails to compute a genuine score increase, and consequently false rejects increase.

Figure 1.1 shows the factors which affect security and convenience in biometric identification. They are related to each other, and become serious with the increase of enrollees. Although biometric identification has a potential to provide highly convenient and secure way of authentication, the above problems have to be addressed to make it widely used for commercial applications.

1.2 Goals

The goal of this study is to optimize security and convenience in positive biometric identification. Here we include false accepts, wolves, and lambs as factors which affect security, and false rejects, the number of inputs, and response time as factors which affect convenience.

Other than the above problems, there are also problems which are common to both verification and identification. For example, several biometric systems are vulnerable to a spoofing attack where an attacker presents an artifact such as gummy fingerprints [70] and photographs, and a number of liveness detection methods [100] have been proposed to counteract them. Another example is a leakage of biometric information. Since biometric features such as fingerprints and veins are unchangeable throughout a lifetime, they cannot be changed or revoked like passwords even if they are leaked. A number of template protection schemes, which transform the biometric features by a kind of encryption function and match them in the transformed domain, have been proposed to prevent the leakage of the original biometric features [54]. However, liveness detection and template protection are not the focus of this dissertation. As will be described in Chapter 7, our proposals in this dissertation can be combined with both of them to increase the total security of the system.

On the other hand, studies on optimization of false accepts, security against wolves and lambs, false rejects, the number of inputs, and response time in identification seems to be lacking. For example, Nandakumar *el al.* [76] proposed a multi-modal fusion scheme in identification which can minimize the identification error probabilities, as described in Section 2.1.3 in detail. However, since this scheme identifies the claimant after he/she inputs all the query samples, it always requires him/her to input all the query samples, which can cause inconvenience. While most of the conventional fusion schemes make a decision after the claimant inputs all the query samples [57, 96, 97], some studies proposed a fusion scheme in verification which makes a decision each time a claimant inputs a query sample [2, 68, 87, 107]. The former scheme is referred to as a parallel fusion scheme, while the latter as a sequential fusion (or serial fusion) scheme. A sequential fusion scheme can reduce identification errors while keeping down the number of inputs required. No studies, however, have ever proposed a sequential fusion scheme in identification which can optimize the trade-off between identification errors and the number of inputs, to the best of our knowledge. Similarly, no studies have ever tried to optimize response time and security against wolves and lambs, along with the above factors. In this dissertation, we aim at optimizing all of them.

1.3 Organization of This Dissertation

The rest of this dissertation is organized as follows. In Chapter 2, we describe previous work related to this dissertation. Among them, we focus on score level approaches which use scores as information sources because they have several advantages over other approaches. We describe the advantages of score level approaches and clarify challenges which have not been solved in previous work.

In Chapter 3, we make an attempt to develop optimal sequential fusion schemes in identification with regard to identification errors (false accepts and false rejects) and the number of inputs. To this end, we focus our attention on MSPRT's (Multi-hypothesis Sequential Probability Ratio Tests) [30], multi-hypothesis tests which can minimize the average number of observations. Dragalin *et al.* [30] found two such tests and referred to them as *Test* δ_a and *Test* δ_b, respectively. In this chapter, we firstly propose two sequential fusion schemes in identification: the PPSI (Posterior Probability-based Sequential Identification) scheme and the LRSI (Likelihood Ratio-based Sequential Identification) scheme. The PPSI scheme is based on MSPRT (Test δ_a) and can minimize the average number of inputs (referred to as *ANI*) under some conditions, while the LRSI scheme is a simpler one which can be carried out quickly. We secondly prove that the LRSI scheme can also minimize ANI by proving that *this scheme is equivalent to MSPRT (Test* δ_b*)* under some conditions. We also discuss the conditions to achieve the optimality of each scheme. We finally demonstrate the effectiveness of our two schemes through experimental evaluation using the NIST BSSR1 (Biometric Score Set - Release 1) Set1 dataset [77], a multi-modal score dataset (one face and two fingerprints).

In Chapter 4 and Chapter 5, we make an attempt to further optimize response time. To reduce response time using only scores as information sources, we turn our attention to metric space indexing methods (also known as metric access methods, or distance-based indexing methods) [23, 82]. These methods have been developed in the area of similarity search where the system outputs the objects (e.g. images, movies, or documents) in the database which are similar to the query object presented by the user. Although there are a great variety of metric space indexing schemes, we focus on the ones which compute, for each object in the database, some value which is easily computed and highly relevant to a distance (or similarity), and compute scores in order of the value [4, 6, 22, 32]. In this dissertation, we refer to such values as *pseudo-scores*, and such indexing schemes as *pseudo-score based indexing schemes*. The reason we focus on this type of indexing scheme is that it performs very well even if features are in a high dimensional space and includes some state-of-the-art schemes. For example, the permutation-based indexing scheme [4, 22] is one of the most successful metric space indexing schemes, and has received considerable attention in recent years [33, 35, 36, 79, 102].

In Chapter 4, we propose the PPS (Posterior Probability-based Search) scheme which normalizes each pseudo-score to the posterior probability that the corresponding object is within the search radius of the query sample (i.e. answer to the range query). This scheme can enhance the performance of any pseudo-score based scheme and has an optimal property with regard to response time. After describing its algorithm and optimal property, we demonstrate that this scheme outperforms the two state-of-the-art schemes: the standard pivot-based indexing scheme [22] and the permutation-based indexing scheme [4, 22], through experimental evaluation using various kinds of datasets from the Metric Space Library [37].

In Chapter 5, we make an attempt to combine metric space indexing and sequential fusion. We first propose a sequential indexing and fusion framework in identification which is constructed from (I) a pseudo-score based indexing scheme, (II) a sequential search scheme which searches templates using pseudo-scores and scores as a clue, and (III) a sequential fusion scheme. Then we propose the PPSS (Posterior Probability-based Sequential Search) scheme as (II), a modification of the PPS scheme to use *not only pseudo-scores at the current input but past pseudo-scores and scores* as information sources. We also propose a technique which optimizes the number of *pivots*: templates selected from the database which are necessary to compute pseudo-scores. We demonstrate that our proposals significantly reduce the number of score computations of the PPSI scheme while keeping identification errors and ANI, using a large-scale multi-modal dataset ($N = 1800$ enrollees; one face and two fingerprints) obtained by combining the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21].

In Chapter 6, we address an interesting problem of optimization which includes wolves and lambs as factors which affect security. Here we consider the verification mode to simplify the problem, and attempt to optimize the trade-off between security against wolves and lambs and convenience in terms of the number of inputs and false rejects. To clarify our target, we first introduce a taxonomy which classifies wolves into three categories (zero-effort wolves, non-adaptive spoofing wolves, and adaptive spoofing wolves) and lambs into two categories (zero-effort lambs and spoofing lambs). Then, we propose the MLRSV (Minimum Likelihood Ratio-based Sequential Verification) scheme as a sequential fusion scheme in verification. We prove that this scheme has security against wolves and lambs, except for adaptive spoofing wolves, and minimizes ANI and false rejects under some conditions. We also discuss the conditions to achieve the security and optimality, and propose an input order decision scheme based on the KL (Kullback-Leibler) divergence [25] to further reduce ANI in the case where the KL divergence differs from one modality to another. We finally demonstrate the effectiveness of our proposals using a multi-modal (one face and eight fingerprints) dataset obtained by combining the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21].

In Chapter 7, we sum up the contributions of our line of work, and provide a future direction of this study which is necessary to establish a secure and convenient biometric identification system.

Chapter 2

Related Work

Contents

In this chapter, we introduce previous work related to this dissertation. Figure 2.1 shows a typical model of a biometric identification system. At the enrollment phase, an enrollee presents his/her biometric sample and a sensor captures a raw biometric data (e.g. digital image). Then, a feature extractor extracts a biometric feature (e.g. minutiae [52], eigenface [114], iriscode [27]) from the raw biometric data, and enroll it as a template. At the identification phase, a claimant inputs his/her query sample to a sensor, and a feature extractor extracts a biometric feature in the same way. Then, a matcher performs a *one-to-many matching* and outputs scores (similarities or distances) against templates. Finally, a decision module makes a decision who the claimant is (i.e. accepts as an enrollee or rejects as a non-enrollee).

As described in Section 1.1, the problems of false accepts, false rejects, and the number of inputs are related to each other, and become worse as the number of enrollees increases. Response time also increases with the increase of enrollees. Furthermore, the existence of wolves and lambs can cause serious security problems.

We introduce related work to solve these problems. We mainly explain (1) multi-modal biometric fusion for accuracy improvement, (2) classification and indexing for reducing response time, and (3) countermeasures against wolves and lambs as related work. We describe

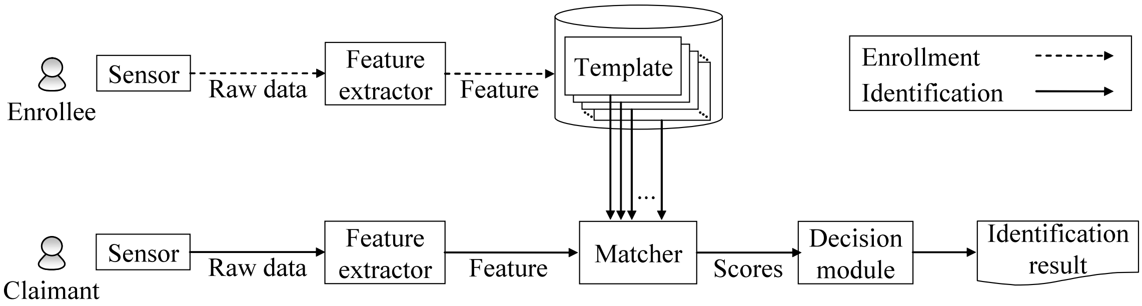


Figure 2.1: Biometric identification system model.

each of them in Section 2.1, 2.2, and 2.3.1, respectively (we also explain related work on the Biometric zoo [29], a classification of users which includes wolves and lambs, from several directions in Section 2.3). Among them, we focus on score level approaches which use scores as information sources in this dissertation. We describe the advantages and challenges of them in Section 2.4 and 2.5, respectively.

2.1 Multi-modal Biometric Fusion

There are several approaches to improving accuracy in biometric authentication. They can be broadly categorized into the following two types: (1) improving a module such as a sensor, feature extractor, and matcher, and (2) using a multi-modal biometric fusion scheme [57, 96, 97]. The former approach is generally specialized in a certain modality such as fingerprint, and all of the sensor, feature extractor, and matcher are often updated together since they are related to each other. The latter approach combines multiple sources of information to achieve high accuracy. This section focuses on the latter and describes its related work.

2.1.1 Fusion Scenarios

There are a variety of scenarios in multi-modal biometric fusion systems depending on the number of biometric traits, instances, samples, and sensors [57, 96, 97]:

1. **Multiple sensors:** The system captures a single biometric trait using multiple sensors (e.g. multiple images of the face) and combines them.

2. **Multiple biometric traits:** The system combines multiple biometric traits of an individual (e.g. fingerprint, face and voice).

3. **Multiple instances:** The system combines a multiple instances of the same body trait (e.g. index and middle fingers). It is also referred to as multiple units.

4. **Multiple samples:** The system acquires multiple samples from the same biometric trait and combines them. It is also referred to as multiple snapshots.

5. **Multiple algorithms:** The system processes the same biometric data using multiple feature extractors or multiple matchers, and combines multiple features or multiple scores. The scenario where multiple features are combined is referred to as multiple representations, and the one where multiple scores are combined as multiple matchers.

In multiple sensors, multiple samples, or multiple algorithms (i.e. 1, 4, 5), the system combines multiple information sources obtained from the same biometric trait. Thus, the information sources to be fused are strongly correlated in these scenarios. On the other hand, in multiple biometric traits or multiple instances (i.e. 2, 3), since the system combines different biometric traits or instances which are independent or weakly dependent, a significant improvement of accuracy can be expected [58]. Therefore, we focus on multiple biometric traits and multiple instances in this dissertation.

2.1.2 Levels of Fusion

According to the type of a module (i.e. sensor, feature extractor, matcher, or decision module) which outputs the information sources to be fused, multi-modal biometric fusion can also be classified as follows [57, 97]:

1. **Sensor level fusion:** This type of fusion combines multiple raw biometric data from the sensor(s). Sensor level fusion can be used in the multiple samples scenario or the multiple sensors scenario. For example, some studies proposed a fingerprint mosaicking scheme which uses multiple samples from the same fingerprint to construct a composite fingerprint template [56, 94]. Another example is a mosaicking scheme in face recognition proposed by Yang *et al.* [125] where 2D face images captured using multiple cameras are used to construct a panoramic face model.

2. **Feature level fusion:** This type of fusion combines multiple features, and can be broadly classified into the following two types: homogeneous feature fusion and heterogeneous feature fusion [57]. Homogeneous feature fusion combines multiple features obtained by applying the same feature extractor to multiple samples of the same biometric trait (i.e. multiple samples or multiple sensors). For example, Jiang and Ser [60] proposed a fusion scheme which estimates a single resultant minutia set as a weighted average over multiple minutia sets from the same fingerprint. Heterogeneous feature fusion combines multiple features from different feature extractors, different biometric traits, or different instances (i.e. multiple representations, multiple biometric traits, or multiple instances). For example, Kumar *et al.* [62] proposed a fusion scheme which combines a palmprint feature and a hand geometry feature. Ross *et al.* [95] proposed a scheme which concatenates feature vectors and then reduce the size of the concatenated vector to solve the *curse of dimensionality problem* where the increase of the vector size can degrade the performance especially when the number of training samples is small.

3. **Score level fusion:** This type of fusion combines multiple scores, and can be classified into the following categories: density-based score fusion, transformation-based score fusion, and classifier-based score fusion [97]. Density-based score fusion estimates a probability density function (pdf) of scores given a class label (e.g. genuine or impostor), and makes a decision using the pdf. For example, Nandakumar *et al.* [75] proposed a likelihood ratio-based fusion scheme in verification using the pdf of genuine scores and that of impostor scores. This scheme is based on the likelihood ratio test, and minimizes the false accept probability for a fixed false reject probability (i.e. optimizes the trade-off between false accepts and false rejects) if the pdfs are perfectly estimated. They modeled each pdf as a finite Gaussian mixture model (GMM). Transformation-based score fusion normalizes scores from the different matchers to make them compatible, and combines

the normalized scores using the combination rule such as the sum, max and min rule [55]. Classifier-based score fusion uses a classifier which learns a decision boundary between genuine scores and impostor scores without estimating a pdf of scores. Examples of classifiers include SVM [11], k-NN classifier [118], logistic regression [117, 118], and random forest [64].

4. **Decision level fusion:** This type of fusion combines multiple decision results. The AND rule and OR rule [14] are examples of decision level fusion. In verification, the AND rule outputs *accept* only if all the matchers output *accept*, while the OR rule outputs *accept* if at least one of the matchers output *accept*. Although they have been originally developed in the verification mode, they can also be extended to the identification mode. One way to do so is to apply the AND/OR rule to each enrollee and output the candidate list [14]. Another example is a majority voting [57, 97] which makes a final decision as follows: if more than half matchers make the same decision, adopt the decision result; otherwise, reject.

In the identification mode where the matcher outputs a ranking of the enrollees, there is also rank level fusion [57, 97] which combines multiple ranking results. However, we leave out the details of this type of fusion in this dissertation.

2.1.3 Multi-modal Biometric Fusion in Identification

Although most of the conventional multi-modal biometric fusion schemes are proposed in the verification scenario, some studies proposed a fusion scheme in the identification mode.

For example, Hong and Jain [43] proposed a score level fusion scheme which integrates faces and fingerprints in identification. This scheme first narrows down enrollees to some candidates using the face matcher, and normalizes a score to the FAR (False Accept Rate) value (the error rate that an impostor is incorrectly accepted in verification) for each candidate and modality. Then, it computes the product of the FAR values (referred to as the *FAR product*) for each candidate, and identifies the claimant by comparing the minimum value of the FAR products to a threshold (if it is smaller than the threshold, accept as the corresponding enrollee; otherwise, reject). Nandakumar *et al.* [76] proposed a scheme which computes, for each enrollee, the posterior probability of being the same person as the claimant using scores, and compares the maximum posterior probability to a threshold (if it is larger than the threshold, accept; otherwise, reject). This scheme is based on the Bayes decision theory (BDT) [31] and minimizes the identification error probabilities if the posterior probability is correctly estimated.

2.1.4 Sequential Fusion

Since we focus on multiple biometric traits or multiple instances as described in Section 2.1.1, we do not consider a scenario, such as multiple sensors and multiple algorithms, where the system combines multiple information sources from a single biometric trait. However, most of the conventional schemes in the multiple biometric traits scenario or multiple instances scenario are parallel fusion schemes which make a decision after the claimant inputs all the query samples. Since the parallel fusion schemes always require the claimant to input all the query samples, they can make the system inconvenient.

Table 2.1: Classification of the score level fusion schemes and decision level fusion schemes introduced in Section 2.1.2, 2.1.3, and 2.1.4.

	parallel	sequential
verification	likelihood ratio (GMM) [75], sum [55], max [55], min [55], SVM [11], k-NN [118], logistic regression [117, 118], random forest [64], AND [14], majority voting [57, 97].	OR [14], LRSV [107].
identification	FAR product [43], Bayes decision theory (BDT) [76], AND [14].	OR [14].

On the other hand, some fusion schemes make a decision each time a claimant inputs a query sample, and are referred to as a sequential fusion (or serial fusion) scheme. This type of fusion scheme can improve accuracy while keeping down the number of inputs required. The OR rule [14] can be regarded as the simplest example of sequential fusion schemes because it can terminate the verification (or identification) process if some modality outputs accept. Some literatures studied other sequential fusion (or serial fusion) schemes in verification [2, 68, 87, 107]. For example, Takahashi *et al.* [107] proposed a sequential fusion scheme in verification which compares a likelihood ratio to a threshold each time the claimant inputs a query sample. This scheme is based on SPRT (Sequential Probability Ratio Test), a statistical hypothesis test which minimizes the average number of observations among all tests with the same error probabilities, if the samples are independent and identically distributed (i.i.d.) [71, 121, 122] (we describe SPRT in Section 3.3 in more detail). In this dissertation, we refer to this scheme as the *LRSV (Likelihood Ratio-based Sequential Verification) scheme.* Since the LRSV scheme applies SPRT to biometric verification, it can minimize ANI (the average number of inputs) among all sequential fusion schemes with the same verification error probabilities. In other words, this scheme can optimize the trade-off between verification errors and the number of inputs. Allano *et al.* [2] evaluated this scheme with respect to the cost which includes the processing time and the financial cost of sensors.

Table 2.1 shows the classification of the score level fusion schemes and decision level fusion schemes we have introduced so far in terms of whether they are in the verification mode or identification mode, and whether they are parallel or sequential.

2.2 Classification and Indexing

There are also several approaches to reducing response time in biometric identification. They can be classified into the following categories: (1) improving a matcher, (2) using a feature level classification or indexing scheme [67], and (3) using a metric level indexing method [23, 82]. The first approach reduces the one-to-one matching time (time to compute a score between one query sample and one template), while the second and third approaches reduce the number of score computations. Although we can also reduce response time by running multiple servers in parallel, we exclude such a solution due to the following reasons: it takes cost to prepare multiple servers; in physical access control systems, the one-to-many matching process is carried out in the client side (i.e. control device) to avoid communication failure.

The first approach is generally specialized in the matcher, and program optimization is its example. In the following, we describe the second and third approaches in detail.

2.2.1 Feature Level Classification/Indexing

The second approach, a feature level classification or indexing scheme, is widely studied in biometrics. Feature level classification divides templates into some predefined classes in advance. At the identification phase, it reduces the number of score computations by assigning a query sample to some class and searching only templates in the class. For example, the Galton-Henry classification scheme [38, 41], which classifies fingerprints into the categories such as arch, tented arch, left loop, right loop, and whorl, has been used by law enforcement agencies worldwide. However, in feature level classification, the number of classes is generally small and features are non-uniformly distributed among the classes. They are problems in effectively reducing the number of score computations.

Feature level indexing, also known as continuous classification, does not assign a query sample to a predefined class but provides a continuous ordering of templates by using a numerical vector which simply represents a feature [67]. For example, Cappelli *et al.* [20] proposed a fingerprint indexing scheme which uses templates corresponding to the above five classes (arch, tented arch, left loop, right loop, and whorl) to create a numerical vector. This scheme computes the cost of the adaptation of each template to a biometric sample and uses a five-dimensional vector composed of the normalized costs for providing a continuous ordering. Tan *et al.* [109] compared the performance of feature level indexing with that of feature level classification, and reported that the former outperformed the latter. Nowadays, feature level indexing is widely studied especially in fingerprints [19, 45, 51, 63, 101].

2.2.2 Metric Space Indexing

The third approach, a metric space indexing method (also known as metric access method, or distance-based indexing method) has been developed in the area of similarity search [23, 82]. Here the goal is generally to find a set of objects (e.g. images, movies, or documents) in the database whose distance (or similarity) to the query object is less than (or more than) the threshold (i.e. answer to the range query) or the k-nearest neighbor objects (i.e. answer to the k-NN query). This method reduces the number of score computations between the query object and the objects in the database using an index which is constructed based on scores. The main feature of this method is that it can be applied as long as a score measure (i.e. a measure of distances or similarities between objects) is defined. In fact, it has a variety of applications which retrieve complex data such as audio, images, videos, documents, and biometrics.

One way to categorize metric space indexing methods is based on whether they guarantee to return the correct answer or not. The former ones are referred to as *exact indexing schemes*, while the latter ones are referred to as *approximate (or inexact) indexing schemes*. Exact indexing schemes make use of the triangle inequality for distances to discard the objects which are not in the correct answer. To date, a considerable number of exact schemes have been proposed: BK-tree [16], AESA [119], LAESA [73], VP-tree [126], GH-tree [115], GNAT [15], etc. The problem of these schemes is that they suffer from the *curse of dimensionality* [23]: the triangle inequality is effective only in the case where features are in a low dimensional space, and they end up computing almost all scores in a high dimensional space. Biometric features are often in a high dimensional space (e.g. Daugman [27] used 2048-bit iriscodes) and score measures in biometrics often do not satisfy the triangle-inequality.

Approximate indexing schemes are designed to quickly find an approximate answer even if features are in a high dimensional space or score measures do not satisfy the triangle-inequality. Examples of them include what we call *pseudo-score based schemes* [4, 6, 22, 32]. They compute, for each object in the database, a *pseudo-score* which is easily computed and highly relevant to a score, and compute scores in ascending (or descending) order of the pseudo-scores. By stopping searching at some halting point, they reduce the number of score computations. The example of pseudo-score based schemes is the standard pivot-based indexing scheme [22], the permutation-based indexing scheme [4, 22], the distance regression-based indexing scheme [32], and BoostMAP [6]. Among them, the permutation-based indexing scheme, which was independently proposed by Chávez *et al.* [22] and Amato and Savino [4], is known to outperform other existing schemes in some cases by a wide margin, and has received considerable attention as one of the most successful metric indexing schemes [33, 35, 36, 79, 102] (we describe the algorithm of this scheme in Section 4.2.3 in detail).

There are other types of approximate indexing schemes: PAC-NN [24], the probabilistic incremental search [17], SASH [44], iAESA [35], PP-Index [33], M-Index [79], etc. Some studies also proposed an approximate indexing scheme for biometric identification [10, 39, 65] (we describe them at the beginning of Chapter 5 in detail).

2.3 Biometric Zoo

It is known that different users have different degrees of accuracy in biometric authentication. Doddington *et al.* [29] classified users in speaker recognition as follows:

- Sheep: those who are easily recognized (default users);
- Goats: those who are particularly difficult to recognize;
- Lambs: those who are particularly easy to imitate;
- Wolves: those who are particularly successful at imitating others.

This concept is known as the *biometric zoo* (or *Doddington's zoo*). Goats cause many false rejects, while wolves and lambs cause many false accepts. Doddington *et al.* also showed the existence of these animals through statistical tests on scores [29].

To date, numerous studies have been made on this issue from several directions. Poh and Kittler [88] examined the potential of some score-based measures as an index characterizing recognizability of a user. They also proposed BMI (Biometric Menagerie Index), a measure to quantify the extent of the biometric zoo [89]. Yager and Dunstone [124] introduced a new class of animals considering a relationship between genuine and impostor scores. Teli *et al.* [110] investigated the consistency of the biometric zoo across algorithms and datasets. Similarly, Paone and Flynn [81] investigated the consistency across algorithms and two irises for a single user.

Although the above work studied the difference of recognizability among individuals, recent studies also proposed a sophisticated attack relevant to the animals. For example, Une *et al.* [116] proposed a *universal wolf* sample, an artifact which has extremely high similarities (or extremely low distances) against all templates. They also proposed WAP (Wolf Attack Probability), the maximum probability of false accepts caused by a query sample, as a security measure for wolves, and showed that the proposed artifact can achieve WAP = 100[%] in the most basic verification system which compares a score to a threshold. It should be noted that if the artifact is enrolled in the database in identification, it can make the system identify all claimants as the enrollee who input the artifact, thus completely losing the availability of the system. We refer to such template as a *universal lamb template*.

An attack to a template update mechanism, which updates a template using query samples presented at the authentication phase, is also proposed by Wang *et al.* [123]. They applied the Frog-Boiling attack to keystroke template update mechanisms to changes the victim's template little by little towards a template of an ill-performing animal such as a lamb. Although this attack assumes that the attacker already has a query sample which can impersonate the victim, we consider this is still threatening in identification because lambs can lose the availability of the system as described above. Therefore, we do not consider template update mechanisms in this dissertation.

In the following, we introduce related work on countermeasures against wolves and lambs.

2.3.1 Countermeasures against Wolves and Lambs

Some studies proposed countermeasures against wolves. For example, Inuma *et al.* [47] proposed a countermeasure which estimates a feature distribution for each of all human beings and determines a verification threshold for each query sample using the feature distributions. This method can keep WAP less than a desired value if the feature distributions are perfectly estimated. Kojima *et al.* [61] proposed another countermeasure in verification using decision results (accept or reject) with biometric samples other than the template to detect wolves. Since the former countermeasure uses feature distributions and the latter one uses decision results, they can be regarded as a feature level approach and a decision level approach, respectively.

As for lambs, we can counteract them before authentication since lamb templates are presented at the enrollment phase. For example, we can detect a lamb template at the enrollment phase, by computing scores against other templates in the database [49], and make the enrollee to re-enroll another biometrics. Although this method can easily detect a universal lamb template mentioned above, it can miss a more *ambiguous* lamb template which has high similarities to some people and low similarities to others. Since human beings can present such an ambiguous lamb template rather than a universal lamb template as shown by Doddington *et al.* [29], countermeasures against such a template is necessary.

We can counteract ambiguous lamb templates at the authentication phase, using a score normalization scheme [85] which uses enrollee-specific parameters or impostor distributions. To date, a number of score normalization schemes have been proposed, and a survey of them is given in [85]. For example, Z-norm [7] attempts to normalize an impostor distribution of each enrollee to a distribution with zero-mean and unit-variance. Since impostor distributions of lambs are also centered around the distribution, this scheme can improve security against lambs. Another example is a selective fusion scheme in verification proposed by Ross *et al.* [98]. This scheme detects weak templates such as lamb templates and goat templates, and invokes fusion only for enrollees who have such weak templates. Since this scheme requires only such enrollees to input multiple biometrics, it can reduce verification errors caused by lambs and goats while keeping down the number of inputs of other enrollees.

We finally note that all of the above countermeasures against lambs (i.e. lamb detection, score normalization, and selective fusion) are score level approaches.

2.4 Advantages of Score Level Approaches

In Section 2.1, 2.2, and 2.3.1, we explained related work on multi-modal biometric fusion, classification and indexing, and countermeasures against wolves and lambs. According to the level of information sources (i.e. raw data, features, scores, or decision results), they can be classified into sensor level approaches, feature level approaches, score level approaches, and decision level approaches. The information available to the system becomes compressed as it is processed from the sensor to the decision module. The raw data contains the richest information, while the decision result contains the poorest information (e.g. a binary sequence representing an enrollee ID or a non-enrollee). Approaches which use richer information have a potential to provide a better performance.

Nonetheless, in this dissertation we focus on score level approaches which use scores as information sources. The reason for this is as follows:

1. Most commercial biometric systems do not provide access to features nor raw data [57, 97], and both of the feature level approaches and sensor level approaches cannot be applied to such systems. On the other hand, score level approaches can be applied to such systems if they provide access to scores. For example, in BioAPI [48], the international standard for the API used for the development of the biometric system, a one-to-one matching function outputs a score.

2. A template protection scheme [54], which transforms features (or raw data) by a kind of encryption function and matches them in the transformed domain, can make feature level approaches (or sensor level approach) difficult to apply. For example, feature level indexing needs to extract a numerical vector from a distorted image [93] or an image which is indistinguishable from a random sequence [106]. On the other hand, score level approaches can be used in conjunction with a template protection scheme which outputs scores [93, 106, 111, 112].

3. The feature level approaches and sensor level approaches generally specialize in a certain modality such as fingerprint, face, and iris. On the other hand, score level approaches can be applied to any kind of modality since they only use scores as information sources.

4. In practice, it is not always the case that feature level approaches or sensor level approaches can provide a better performance than score level approaches. This is because features or raw data are more difficult to handle than scores. For example, the dimension of features or raw data is often very high (e.g. 2048-bit iriscodes [27]), and can be changed from one acquisition to another (e.g. the number of minutiae varies from sample to sample), while the dimension of scores is only one. Thus, the estimation of feature distributions, which is required in the countermeasure against wolves in [47], is generally much more difficult than the estimation of score distributions. There are also experimental results where score level fusion outperformed feature level fusion [62, 95].

5. On the other hand, since scores are easy to handle as described above (e.g. the dimension of scores is one) and contain much richer information than decision results, score level approaches generally provide a much better performance than decision level approaches (in Section 3.6, we also show that the proposed score level fusion schemes significantly outperform the OR rule, one of the decision level fusion schemes).

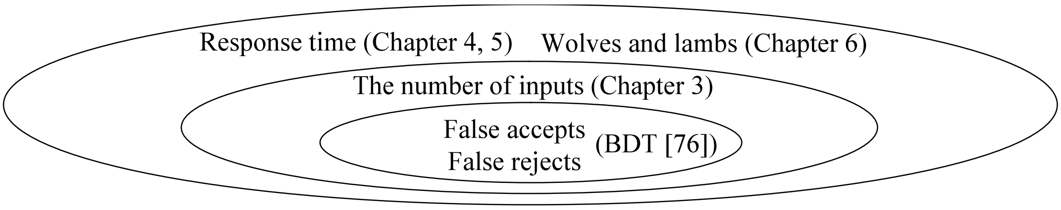

Response time (Chapter 4, 5)
Wolves and lambs (Chapter 6)
The number of inputs (Chapter 3)
False accepts
False rejects
(BDT [76])

Figure 2.2: Challenges we attempt to address in this dissertation. Although Nandakumar *et al.* [76] made an attempt to optimize false accepts and false rejects in identification using the Bayes decision theory (BDT), we attempt to further optimize the number of inputs in Chapter 3. In addition to these factors, we attempt to optimize response time in Chapter 4 and 5, and security against wolves and lambs in Chapter 6.

The improvement of a module such as a sensor, feature extractor, and matcher also has the first and third problems (i.e. generally specializes in a certain modality and cannot be made in the commercial biometric systems). Conversely, in the application where we can make the improvement of the module, score level approaches can be further applied to provide a better performance.

2.5 Challenges

Recall that the goal of this dissertation is to optimize security and convenience in biometric identification, where we include false accepts, wolves, and lambs as factors which affect security, and false rejects, the number of inputs, and response time as factors which affect convenience. Although we have so far introduced a number of related work, there are still some major challenges which need to be addressed:

- Firstly, little attention has been given to the optimization of identification errors and the number of inputs in biometric identification. Most of the conventional multi-modal biometric fusion schemes are proposed in the verification mode or parallel fusion schemes, as described in Section 2.1.3 and 2.1.4. For example, Nandakumar *et al.* [76] proposed a fusion scheme in identification which can minimize the identification error probabilities (i.e. optimize false accepts and false rejects), using the Bayes decision theory (BDT). However, this scheme is a parallel fusion scheme which always requires the claimant to input all the query samples, making the system inconvenient. In Chapter 3, we make an attempt to develop sequential fusion schemes in identification which optimize the trade-off between identification errors and the number of inputs.

- Secondly, no studies have ever tried to optimize response time along with the above factors (identification errors and the number of inputs) in biometric identification, to the best of our knowledge. In this dissertation, we take two steps to achieve this goal. In Chapter 4, we make an attempt to develop metric space indexing schemes which have an optimal property with regard to the number of score computations, and outperform the state-of-the-art metric space indexing schemes such as the permutation-based indexing scheme [4, 22] described in Section 2.2.2. In Chapter 5, we make an attempt to optimize the trade-off between identification errors, the number of inputs, and response time, by combining metric space indexing and sequential fusion in identification.

- Last but not least, no studies have ever tried to optimize security against wolves and lambs and convenience in biometrics. In Chapter 6, we consider the verification mode to simplify the problem, and attempt to optimize the trade-off between security against wolves and lambs and convenience in terms of the number of inputs and false rejects.

Figure 2.2 shows challenges we attempt to address in this dissertation. Although we focus on score level approaches which have several advantages as described in Section 2.4, it should be noted that the fact that the above three challenges have not been solved is true for all of the sensor level, feature level, score level, and decision level approaches.

Chapter 3

Towards Optimal Sequential Fusion in Biometric Identification

Contents

3.1 Introduction

The aim of this chapter is to develop sequential fusion schemes in biometric identification which optimize the trade-off between identification errors and the number of inputs. Takahashi *et al.* [107] proposed a sequential fusion scheme in verification which we call the LRSV (Likelihood Ratio-based Sequential Verification) scheme. This scheme is based on SPRT (Sequential Probability Ratio Test), a statistical hypothesis test which can minimize the average number of observations among all tests with the same error probabilities [71, 121, 122]. Since this scheme applies SPRT to biometric verification, it can minimize ANI (the average number of inputs) among all sequential fusion schemes with the same verification error probabilities.

However, in constructing optimal sequential fusion schemes in identification, we must note that SPRT is a binary test which accepts either a null hypothesis or an alternative hypothesis (i.e. binary classification). In contrast, biometric identification accepts a claimant as one of the enrollees in the database or rejects him/her as a non-enrollee (i.e. multi-class classification). Noda and Kawaguchi [78] proposed a technique which reduces the utterance length in speaker identification by extending SPRT to multi-class classification and using features as information sources. However, it is not proved that this technique can minimize the average utterance length since SPRT is optimal only in the case of binary classification.

3.1.1 Our Contributions

In this chapter, we provide a theoretical basis for optimal sequential fusion in biometric identification. To this end, we focus our attention on MSPRT's (Multi-hypothesis Sequential Probability Ratio Tests) [30], multi-hypothesis tests which can minimize the average number of observations in the case of multi-class classification. More specifically, Dragalin *et al.* [30] found two such tests and referred to them as *Test* δ_a and *Test* δ_b, respectively. Then, the main contributions of this chapter can be written as follows:

1. We firstly propose two sequential fusion schemes in identification: the PPSI (Posterior Probability-based Sequential Identification) scheme and the LRSI (Likelihood Ratio-based Sequential Identification) scheme. The PPSI scheme is based on MSPRT (Test δ_a) and can minimize ANI (the average number of inputs) under some conditions, while the LRSI scheme is a simpler one which can be carried out quickly.

2. We secondly prove that the LRSI scheme can also minimize ANI, by proving that *this scheme is equivalent to MSPRT (Test* δ_b*)* under some conditions. We also discuss the conditions to achieve the optimality.

3. We finally demonstrate the effectiveness of our proposals through experimental evaluation using the NIST BSSR1 (Biometric Score Set - Release 1) Set1 dataset [77], a multi-modal score dataset (one face and two fingerprints).

3.1.2 Organization of This Chapter

This chapter is organized as follows. In Section 3.2, we define accuracy measures for positive biometric identification. In Section 3.3, we describe MSPRT's [30] in detail. In Section 3.4, we propose the PPSI scheme and the LRSI scheme as sequential fusion schemes in identification. In Section 3.5, we prove that the optimality of the LRSI scheme with regard to ANI, and discuss the conditions to achieve the optimality. In Section 3.6, we show the experimental results using the NIST BSSR1 Set1 [77], and discuss the results. Finally, we conclude this chapter in Section 3.7.

3.2 Accuracy Measures for Positive Biometric Identification

In the verification mode, two kinds of error rates are defined: FRR (False Reject Rate) and FAR (False Accept Rate). FRR is the error rate that the system rejects a genuine individual

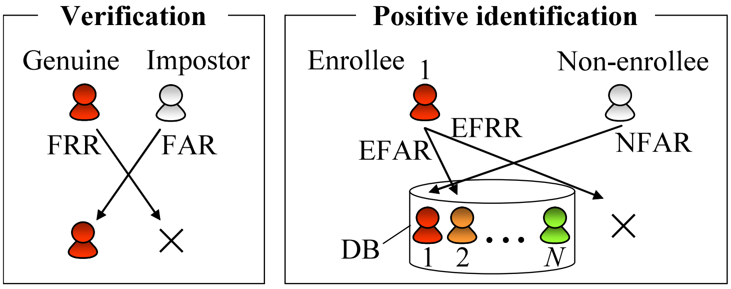


Figure 3.1: Three error rates in positive identification (EFRR/EFAR/NFAR).

as an impostor. FAR is the error rate that the system accepts an impostor as a genuine individual. High FRR causes inconvenience, and high FAR causes security problems.

In the identification mode, FNIR (False Negative Identification Rate) and FPIR (False Positive Identification Rate) are defined as accuracy measures [50]. FNIR is the error rate that the system does not include an enrollee who inputs a query sample in the candidate list. FPIR is the error rate that the system outputs one or more candidates when a non-enrollee inputs a query sample. In criminal investigations, for example, it is essential to decrease both FNIR and FPIR.

Positive identification, however, typically outputs at most one candidate and is used for the applications which provide service appropriate to the identified user, as described in Section 1.1. In such cases, the identification error in the case where the enrollee inputs a query sample can be divided into two types: accepting the enrollee as another enrollee, and rejecting the enrollee as non-enrollee. Although both of the errors cause inconvenience since the enrollee cannot use appropriate service, the former error further causes security problems because the system incorrectly provides service appropriate to another enrollee.

Taking these matters into account, we newly define three kinds of error rates as performance measures for positive identification systems.

EFRR (Enrollee False Reject Rate):
The error rate that the system incorrectly rejects an enrollee as a non-enrollee.
EFAR (Enrollee False Accept Rate):
The error rate that the system incorrectly accepts an enrollee as another enrollee.
NFAR (Non-Enrollee False Accept Rate):
The error rate that the system incorrectly accepts a non-enrollee as an enrollee.

Taking computer login systems for example, EFRR is the error rate that an enrollee fails to login, EFAR is the error rate that an enrollee logins to another account, NFAR is the error rate that a non-enrollee (i.e., an attacker) logins to someone's account. High EFRR causes inconvenience, high EFAR causes both inconvenience and security problems, and high NFAR causes security problems. Figure 3.1 shows the three error rates in positive identification (EFRR/EFAR/NFAR).

3.2.1 Relationship between Identification Errors and the Number of Inputs

In Section 1.1, we described that false accepts increase as the number of enrollees increases in identification. We now derive the relationship between the three kinds of identification errors and the number of enrollees. Here, to differentiate error probabilities from error rates, we define FRP (False Reject Probability), FAP (False Accept Probability), EFRP (Enrollee FRP), EFAP (Enrollee FAP), and NFAP (Non-enrollee FAP) as an *error probability* corresponding to FRR, FAR, EFRR, EFAR, and NFAR, respectively.

Take FRP, FAP, FRR, and FAR for example. Let $d \in \{0, 1\}$ be a variable which takes 1 or 0 if the final decision result is *accept* or *reject*, respectively. Let further W_1 be the event that a genuine user attempts verification against him/herself, and W_0 be the event that an impostor attempts verification against someone else. Then, FRP and FAP can be written as follows:

$$\begin{aligned} FRP &= P(d = 0|W_1) && (3.1) \\ FAP &= P(d = 1|W_0), && (3.2) \end{aligned}$$

where $P()$ is a probability mass function. Since they are theoretical values, in practice FRR and FAR are evaluated, instead of FRP and FAP, using a finite number of biometric samples as follows:

$$\begin{aligned} FRR &= \frac{\text{The number of false rejects}}{\text{The total number of genuine attempts}} && (3.3) \\ FAR &= \frac{\text{The number of false accepts}}{\text{The total number of impostor attempts}}. && (3.4) \end{aligned}$$

Similarly, EFRP/EFAP/NFAP is a theoretical value, while EFRR/EFAR/NFAR is the number of error rates divided by the total number of attempts. We use error probabilities (i.e. FRP, FAP, EFRP, EFAP, NFAP) in a theoretical analysis, and error rates (i.e. FRR, FAR, EFRR, EFAR, NFAR) in an experimental evaluation.

We now derive the relationship between EFRP/EFAP/NFAP and the number of enrollees N. Let $f()$ be a distribution of a genuine score (i.e. score between the same individual) and $g()$ be a distribution of an impostor score (i.e. score between the different individuals). We refer to $f()$ and $g()$ as a *genuine distribution* and *impostor distribution*, respectively. We first consider a verification system which makes a decision by comparing a similarity score s to a verification threshold s_{th} (we assume that a similarity score s is continuous). Then, FRP and FAP can be expressed as follows:

$$\begin{aligned} FRP &= \int_{-\infty}^{s_{th}} f(s)ds && (3.5) \\ FAP &= \int_{s_{th}}^{\infty} g(s)ds && (3.6) \\ &= 1 - G(s_{th}), && (3.7) \end{aligned}$$

where $G()$ is a cumulative distribution function corresponding to $g()$. We then consider an identification system which makes a decision as follows: if one or more similarity scores exceed an identification threshold s_{th}, identify as the enrollee whose score is the highest; otherwise, reject. The probability that a genuine score falls below the threshold and the probability that an impostor score exceeds the threshold can be expressed as FRP and FAP, respectively. Thus, by assuming that all scores are independent, we can obtain the following approximation if $N \times FAP \ll 1$:

$$
\begin{aligned}
EFRP &= FRP \times (1 - FAP)^{N-1} && (3.8)\\
&\approx FRP \times [1 - (N-1) \times FAP] && (3.9)\\
&\approx FRP && (3.10)\\
EFAP &= \int_{-\infty}^{s_{th}} f(s) \left\{1 - [G(s_{th})]^{N-1}\right\} ds + \int_{s_{th}}^{\infty} f(s) \left\{1 - [G(s)]^{N-1}\right\} ds && (3.11)\\
&= \int_{-\infty}^{s_{th}} f(s) ds \times \left\{1 - [1 - (1 - G(s_{th}))]^{N-1}\right\} \\
&\quad + \int_{s_{th}}^{\infty} f(s) \left\{1 - [1 - (1 - G(s))]^{N-1}\right\} ds && (3.12)\\
&\approx \int_{-\infty}^{s_{th}} f(s) ds \times \{1 - [1 - (N-1) \times (1 - G(s_{th}))]\} \\
&\quad + \int_{s_{th}}^{\infty} f(s) \{1 - [1 - (N-1) \times (1 - G(s))]\} ds && (3.13)\\
&= (N-1) \times \left[FRP \times FAP + \int_{s_{th}}^{\infty} f(s) (1 - G(s)) ds\right] && (3.14)\\
NFAP &= 1 - (1 - FAP)^{N} && (3.15)\\
&\approx N \times FAP. && (3.16)
\end{aligned}
$$

That is, EFAP and NFAP increase almost in proportion to the number of enrollees N.

3.3 MSPRT's

This chapter aims at optimizing the trade-off between the identification error probabilities (EFRP/EFAP/NFAP) and the number of inputs. MSPRT's (Multi-hypothesis Sequential Probability Ratio Tests) [30] play a key role in achieving our aim.

Assume that either of the hypotheses $H_0, H_1, \cdots, H_N$ $(N \geq 1)$ is true, and observed data $\boldsymbol{s_1}, \boldsymbol{s_2}, \cdots, \boldsymbol{s_t}$ are distributed according to the true hypothesis ($\boldsymbol{s_t}$ can be either a scalar or a vector, and can be either continuous or discrete; in this dissertation, we assume that $\boldsymbol{s_t}$ is a vector and continuous). We begin with a sequential test of binary hypotheses: a null hypothesis H_0 and an alternative hypothesis H_1 $(N = 1)$. After a set of data $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$ is obtained, SPRT (Sequential Probability Ratio Test) [121] computes a likelihood ratio which is given by

$$
Z_t = \frac{p(\boldsymbol{S_t}|H_1)}{p(\boldsymbol{S_t}|H_0)}, \qquad (3.17)
$$

where $p()$ is a probability density function. Then, SPRT makes the following decision: if Z_t exceeds a threshold A_{high}, accept H_1; if Z_t falls below another threshold A_{low} $(< A_{high})$, accept H_0; otherwise, continue observing data. It is proved that this test minimizes the average number of observations among all binary tests with the same error probabilities if observed data $\boldsymbol{s_1}, \boldsymbol{s_2}, \cdots, \boldsymbol{s_t}$ are independent and identically distributed (i.i.d.) [71, 122].

We now consider a sequential test of multiple hypotheses $H_0, H_1, \cdots, H_N$ $(N \geq 2)$. The probability that the hypothesis H_i $(0 \leq i \leq N)$ is incorrectly accepted is given by

$$\alpha_i = \sum_{j \neq i} P(H_j)\alpha_{ji}, \tag{3.18}$$

where $P(H_j)$ is the prior probability that H_j is true, and α_{ji} is the probability of accepting H_i when H_j is true. Let δ be a sequential test of multiple hypotheses, and consider a set of tests whose α_i does not exceed the required value $\overline{\alpha}_i$ $(0 \leq i \leq N)$:

$$\Delta(\overline{\alpha}) = \{\delta : \alpha_i \leq \overline{\alpha}_i, 0 \leq i \leq N\}, \tag{3.19}$$

where $\overline{\alpha} = (\overline{\alpha}_0, \overline{\alpha}_1, \cdots, \overline{\alpha}_N)$[1]. Then, Dragalin *et al.* [30] proved that, in the asymptotic case where $\overline{\alpha}_{max} = \max_{0 \leq i \leq N} \overline{\alpha}_i$ goes to 0 (the number of observations t goes to ∞), the following tests (MSRPT's) minimize the average number of observations among $\Delta(\overline{\alpha})$:

MSPRT (Test δ_a):

After a set of data $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$ is obtained, compute, for each hypothesis H_i $(0 \leq i \leq N)$, the posterior probability that H_i is true as follows:

$$P(H_i|\boldsymbol{S_t}) = \frac{P(H_i)Z_{ti}}{\sum_{n=0}^{N} P(H_n)Z_{tn}}, \tag{3.20}$$

where Z_{ti} is a likelihood ratio between H_i and H_0 which is given by

$$Z_{ti} = \frac{p(\boldsymbol{S_t}|H_i)}{p(\boldsymbol{S_t}|H_0)}. \tag{3.21}$$

If one or more $P(H_i|\boldsymbol{S_t})$ exceed a threshold A_i, accept one of the corresponding hypotheses. Otherwise, continue observing data.

MSPRT (Test δ_b):

After a set of data $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$ is obtained, compute, for each hypothesis H_i $(0 \leq i \leq N)$, the following value:

$$L_{ti} = \frac{P(H_i)p(\boldsymbol{S_t}|H_i)}{\max_{0 \leq n \leq N, n \neq i} P(H_n)p(\boldsymbol{S_t}|H_n)}. \tag{3.22}$$

L_{ti} is referred to as a *generalized likelihood ratio* between H_i and the remaining hypotheses. If one or more L_{ti} exceed a threshold B_i, accept one of the corresponding hypotheses. Otherwise, continue observing data.

The thresholds A_i and B_i are set to be the following values:

$$A_i = \log(P(H_i)/\overline{\alpha}_i) \tag{3.23}$$

$$B_i = \log(N \cdot P(H_i)/\overline{\alpha}_i). \tag{3.24}$$

Then, both of the tests satisfy the requirements of the identification error probabilities (i.e. $\delta_a, \delta_b \in \Delta(\overline{\alpha})$), and minimize the average number of observations in the asymptotic case where

[1] More specifically, Dragalin *et al.* [30] considered more general case: they also introduced a loss function $W(j,i) \in [0,\infty)$ in the case where H_i is accepted when H_j is true ($W(i,i) = 0$), and considered a set of tests whose risk given by $R_i = \sum_{j \neq i} P(H_j)W(j,i)\alpha_{ji}$ do not exceed the required value $\overline{R}_i$ $(0 \leq i \leq N)$. However, in this dissertation we consider the *zero-one* loss function where $W(j,i) = 1$ $(j \neq i)$, for simplicity.

$\overline{\alpha}_{max}$ ($= \max_{0 \leq i \leq N} \overline{\alpha}_i$) goes to 0 [30]. Note that this asymptotic optimality is proved not only in the i.i.d. case but also in the *non-i.i.d.* case (i.e. the case where observed data are not independent nor identically distributed). For more details, refer to [30].

3.4 Proposed Sequential Fusion Schemes in Identification

We now propose two sequential fusion schemes in biometric identification: the PPSI (Posterior Probability-based Sequential Identification) scheme and the LRSI (Likelihood Ratio-based Sequential Identification) scheme. The former is based on MSPRT (Test δ_a), while the latter is a simpler scheme which can be carried out quickly.

3.4.1 Posterior Probability-based Sequential Identification Scheme

We first explain the overview of the PPSI (Posterior Probability-based Sequential Identification) scheme. Let s_{ti} be a score for the i-th enrollee at the t-th input, and $\boldsymbol{s_t}$ be the following N-dimensional score vector:

$$\boldsymbol{s_t} = (s_{t1}, s_{t2}, \cdots, s_{tN}). \tag{3.25}$$

We use the score vector $\boldsymbol{s_t}$ as observed data, and define the following hypotheses:

H_i: The user is the i-th enrollee ($1 \leq i \leq N$).

H_0: The user is a non-enrollee.

Now we can apply MSPRT (Test δ_a) to biometric identification. After the claimant inputs the t-th query sample, the PPSI scheme identifies him/her as follows:

1. Compute a score vector $\boldsymbol{s_t} = (s_{t1}, s_{t2}, \cdots, s_{tN})$.

2. Compute the posterior probability $P(H_i|\boldsymbol{S_t})$ ($0 \leq i \leq N$) using a set of score vectors $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$ (we describe how to compute $P(H_i|\boldsymbol{S_t})$ in detail in Section 3.4.2).

3. If one or more $P(H_i|\boldsymbol{S_t})$ exceed a threshold A ($= A_0 = A_1 = \cdots = A_N$), accept the hypothesis H_i whose $P(H_i|\boldsymbol{S_t})$ is the highest (if $1 \leq i \leq N$, accept as the i-th enrollee; if $i = 0$, reject). Otherwise, require another biometric input (if the number of inputs has reached an upper limit T, reject).

Here we use an identification threshold A ($= A_0 = A_1 = \cdots = A_N$) which is common to all hypotheses. In Section 3.4.2, we also describe the reason for this. Figure 3.2 shows the overview of the PPSI scheme.

3.4.2 Computation of Posterior Probabilities

We now explain how to compute $P(H_i|\boldsymbol{S_t})$ ($0 \leq i \leq N$) in detail. The PPSI scheme first computes the likelihood ratio $Z_{ti} = p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_0)$ ($0 \leq i \leq N$) using a set of score vectors $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} = (s_{\tau 1}, s_{\tau 2}, \cdots, s_{\tau N}) | 1 \leq \tau \leq t\}$ (see (3.21)), and then normalizes it to the posterior probability $P(H_i|\boldsymbol{S_t})$ ($0 \leq i \leq N$) (see (3.20)).

We begin by explaining how to compute Z_{ti} ($0 \leq i \leq N$). As described in Section 2.1.1, we focus on sequential fusion of multiple biometric traits (e.g. fingerprint, face and iris) or multiple instances (e.g. index and middle fingers) where each modality is independent or

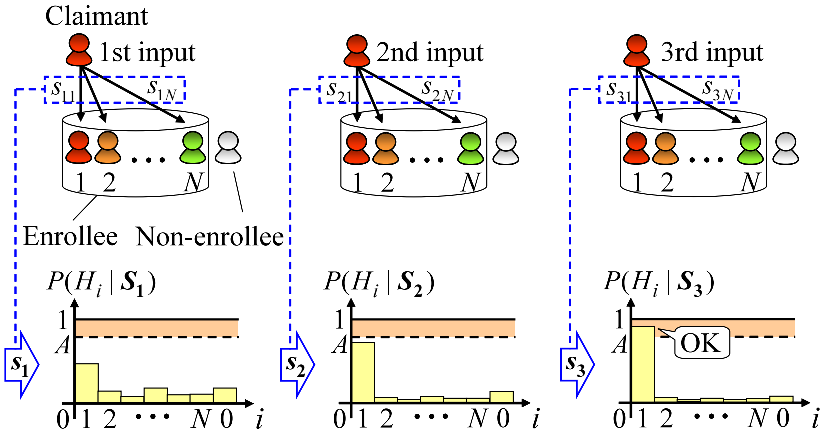


Figure 3.2: Overview of the PPSI (Posterior Probability-based Sequential Identification) scheme.

weakly dependent [58]. Then we can reasonably assume that all scores are independent. We further assume that the likelihood $p(s_{ti}|H_j)$ $(0 \leq i, j \leq N)$ can be written as follows:

$$p(s_{ti}|H_j) = \begin{cases} f^{(t)}(s_{ti}) & \text{(if } i = j\text{)} \\ g_i^{(t)}(s_{ti}) & \text{(if } i \neq j\text{),} \end{cases} \tag{3.26}$$

where $f^{(t)}()$ is a distribution of a genuine score (i.e. genuine distribution), $g_i^{(t)}()$ is a distribution of a score between the i-th enrollee and another person (referred to as an *enrollee-specific impostor distribution*), and t is the input number (i.e. we model these score distributions for each modality). These distributions are depicted in the upper right of Figure 3.3.

The reason we assume an impostor distribution for each enrollee is that it is different from enrollee to enrollee, to be exact (e.g. lambs have high similarity scores against many others). To date, a number of studies have reported that the accuracy was improved by using enrollee-specific impostor distributions [85]. On the other hand, we use a genuine distribution which is common to all enrollees. A genuine distribution is also different from enrollee to enrollee, to be exact (e.g. goats have low similarity scores against themselves). Nevertheless, we assume a genuine distribution common to all enrollees because generally very few genuine scores per enrollee (e.g. 2 or 3 scores) are available as training samples. For example, Poh *et al.* [91] showed that there were low correlation between the standard deviations of the enrollee-specific genuine distributions in the training set and those in the evaluation set. This indicates the difficulty of reliably estimating the enrollee-specific genuine distributions. Furthermore, if each enrollee presents only one biometric sample during enrollment, there are no genuine scores which can be obtained from the sample. Even in such a case, we can train $f^{(t)}$ using genuine scores obtained from other biometric samples which are collected in advance (e.g. biometric samples collected for performance evaluation). We also explain how to train $f^{(t)}$ and $g_i^{(t)}()$ later in detail.

Under the above assumptions, the likelihood ratio $Z_{ti} = p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_0)$ $(0 \leq i \leq N)$

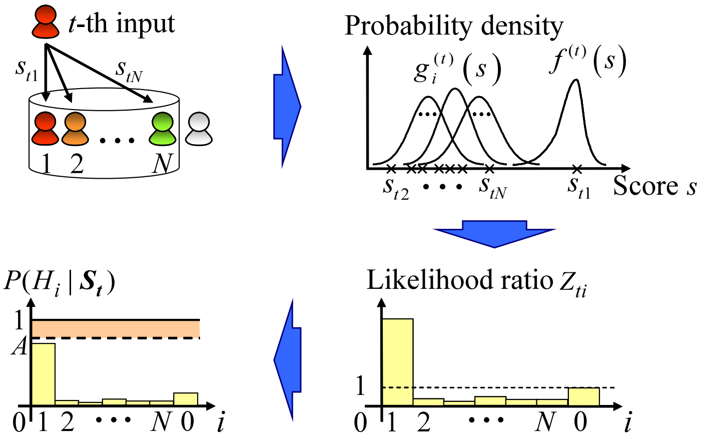


Figure 3.3: Computation of the posterior probability $P(H_i|\boldsymbol{S_t})$ $(0 \leq i \leq N)$ in the PPSI scheme. The genuine distribution $f^{(t)}()$ and the impostor distributions $g_1^{(t)}(), \cdots, g_N^{(t)}()$ (or the likelihood ratio functions $f^{(t)}()/g_1^{(t)}(), \cdots, f^{(t)}()/g_N^{(t)}()$) are trained in advance, using genuine scores and impostor scores, respectively.

is decomposed as follows:

$$
\begin{aligned}
Z_{ti} &= \frac{p(\boldsymbol{S_t}|H_i)}{p(\boldsymbol{S_t}|H_0)} \\
&= \frac{\prod_{\tau=1}^{t} p(s_{\tau i}|H_i) \prod_{n \neq i} p(s_{\tau n}|H_i)}{\prod_{\tau=1}^{t} p(s_{\tau i}|H_0) \prod_{n \neq i} p(s_{\tau n}|H_0)} \\
&= \begin{cases} \prod_{\tau=1}^{t} f^{(\tau)}(s_{\tau i})/g_i^{(\tau)}(s_{\tau i}) & (\text{if } i \neq 0) \\ 1 & (\text{if } i = 0), \end{cases}
\end{aligned}
\tag{3.27}
$$

which can be computed using the genuine distribution $f^{(\tau)}()$ and the impostor distribution $g_i^{(\tau)}()$. These distributions are trained using genuine scores and impostor scores against the i-th enrollee, respectively, both of which are obtained from templates in the database or any other pre-collected biometric samples (e.g. biometric samples collected for performance evaluation). For example, they can be trained by assuming the Gaussian distribution models and using the maximum likelihood estimator. Alternatively, the likelihood ratio function $f^{(\tau)}()/g_i^{(\tau)}()$, from which Z_{ti} is derived (see equation (3.27)), can be directly trained. For example, we can assume the logistic regression model [12] which estimates the log-likelihood ratio function $\log f^{(\tau)}()/g_i^{(\tau)}()$ as follows:

$$
\log f^{(\tau)}(s_{\tau i})/g_i^{(\tau)}(s_{\tau i}) = w_{i1}^{(\tau)} s_{\tau i} + w_{i0}^{(\tau)}, \tag{3.28}
$$

where $w_{i1}^{(\tau)}$ and $w_{i0}^{(\tau)}$ are regression coefficients. Since it does not estimate $f^{(\tau)}()$ and $g_i^{(\tau)}()$ but directly estimate $\log f^{(\tau)}()/g_i^{(\tau)}()$ (i.e. it is not a generative model but a discriminative model [12]), we can expect that it provides good performance. Indeed, some studies showed the validity of the logistic regression model in biometrics [117, 118]. We also show it in our experiment in Section 3.6.

After computing the likelihood ratio Z_{ti} $(0 \leq i \leq N)$, the PPSI scheme normalizes it to the posterior probability $P(H_i|\boldsymbol{S_t})$ $(0 \leq i \leq N)$. The important problem here is that we need to set the prior probability $P(H_i)$ $(0 \leq i \leq N)$ in advance (see (3.20)). Since it is generally very difficult to correctly estimate $P(H_i)$ (i.e. it is very difficult to know who attempts authentication in advance), in this dissertation we assume that $P(H_i)$ is uniform (i.e. $P(H_0) = \cdots = P(H_N) = 1/(N+1)$). This form of prior distribution is known as a *noninformative prior* which is intended to have little influence on the posterior distribution (i.e. to "let the data speak for themselves") [12]. We also set the required identification error probability $\overline{\alpha}_i$ to be uniform (i.e. $\overline{\alpha}_0 = \overline{\alpha}_1 = \cdots = \overline{\alpha}_N$), for simplicity. Then, it follows from (3.23) that we use the identification threshold A $(= A_0 = A_1 = \cdots = A_N)$ which is common to all hypotheses, as described in Section 3.4.1.

To sum up, after a set of score vectors $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$ is obtained, the PPSI scheme computes the posterior probability $P(H_i|\boldsymbol{S_t})$ $(0 \leq i \leq N)$ as follows:

1. Compute the likelihood ratio $Z_{ti} = p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_0)$ $(0 \leq i \leq N)$ using (3.27).

2. Normalize Z_{ti} $(0 \leq i \leq N)$ to the posterior probability $P(H_i|\boldsymbol{S_t})$ $(0 \leq i \leq N)$ using (3.20), where $P(H_i)$ is uniform (i.e. $P(H_0) = \cdots = P(H_N) = 1/(N+1)$).

Figure 3.3 shows the computation of the posterior probabilities in the PPSI scheme.

3.4.3 Likelihood Ratio-based Sequential Identification Scheme

The LRSI (Likelihood Ratio-based Sequential Identification) scheme does not normalize the likelihood ratio Z_{ti} $(0 \leq i \leq N)$ to the posterior probability $P(H_i|\boldsymbol{S_t})$ $(0 \leq i \leq N)$, but compares Z_{ti} $(1 \leq i \leq N)$ to a threshold B (> 1) which is common to all enrollees. Here, since the non-enrollee's likelihood ratio Z_{t0} is always 1 (see (3.27)), the LRSI scheme sets the threshold B larger than 1 and does not compare Z_{t0} to B.

That is, after the claimant inputs the t-th query sample, the LRSI scheme identifies him/her as follows:

1. Compute a score vector $\boldsymbol{s_t} = (s_{t1}, s_{t2}, \cdots, s_{tN})$.

2. Compute the likelihood ratio $Z_{ti} = p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_0)$ $(1 \leq i \leq N)$ from a set of score vectors $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$ using (3.27).

3. If one or more Z_{ti} exceed a threshold B, accept as the enrollee whose Z_{ti} is the highest. Otherwise, require another biometric input (if the number of inputs has reached an upper limit T, reject).

This scheme can be regarded as the extension of the SPRT-based sequential fusion scheme [107] which was proposed in verification to the identification scenario (both of the schemes compare a likelihood ratio against the hypothesis H_0 to the threshold). Noda and Kawaguchi [78] also proposed a technique which reduces the utterance length in speaker identification by extending SPRT to multi-class classification. Their technique computes, for each enrollee, a likelihood ratio against the hypothesis H_0 using features as information sources, and compares it to the threshold. The difference between the LRSI scheme and their technique is that the former is a score level approach (and so is the SPRT-based sequential fusion scheme [107]), while the latter is a feature level approach.

Note that the LRSI scheme can compare a log-likelihood ratio $\log Z_{ti}$ $(1 \leq i \leq N)$ to a threshold $\log B$, instead of comparing Z_{ti} to B. By taking the logarithm of both sides of

(3.27), we can obtain the following equation:

$$\log Z_{ti} = \begin{cases} \log Z_{(t-1)i} + \log f^{(t)}(s_{ti})/g_i^{(t)}(s_{ti}) & (\text{if } i \neq 0) \\ 0 & (\text{if } i = 0). \end{cases} \tag{3.29}$$

Thus, $\log Z_{ti}$ can be updated just by adding $\log f^{(t)}(s_{ti})/g_i^{(t)}(s_{ti})$ to the previous log-likelihood ratio $\log Z_{(t-1)i}$. In particular, if the logistic regression model is used to estimate $\log f^{(t)}(s_{ti})/g_i^{(t)}(s_{ti})$, the algorithm is further simplified and can be carried out very quickly because $\log f^{(t)}(s_{ti})/g_i^{(t)}(s_{ti})$ can be computed just by one multiplication and one addition (see (3.28)). This is particularly helpful in the case of the biometric authentication system whose CPU and memory resources are limited (e.g. physical access control device).

3.5 Optimality of the Proposed Schemes

Since the PPSI scheme uses MSPRT (Test δ_a) as described in the previous section, it can minimize ANI (the average number of inputs) (we discuss the conditions to achieve the optimality in Section 3.5.3). In this section, we also prove the optimality of the LRSI scheme with regard to ANI by proving the equivalence between this scheme and MSPRT (Test δ_b) under some conditions, and discuss the conditions to achieve the optimality.

3.5.1 Properties of N-dimensional Score Distributions

Let $p_i^{(t)}$ $(0 \leq i \leq N)$ be a probability density distribution from which an N-dimensional score vector $\boldsymbol{s_t} = (s_{t1}, s_{t2}, \cdots, s_{tN})$ is generated in the case when the hypothesis H_i is true. We refer to this distribution as an *N-dimensional score distribution*, which is a key to the proof of the optimality of the LRSI scheme. Before proving the optimality of the LRSI scheme, we show some properties of N-dimensional score distributions.

Assume that the following condition holds:

(i) All scores are independent, and scores between the same individual and scores between the i-th enrollee and another person are generated from $f^{(t)}()$ and $g_i^{(t)}()$, respectively, where t is the input number.

Then, $p_i^{(t)}$ $(0 \leq i \leq N)$ can be written as follows:

$$p_i^{(t)}(\boldsymbol{s_t}) = \begin{cases} f^{(t)}(s_{ti}) \cdot \prod_{n=1, n\neq i}^{N} g_n^{(t)}(s_{tn}) & (\text{if } i \neq 0) \\ \prod_{n=1}^{N} g_n^{(t)}(s_{tn}) & (\text{if } i = 0). \end{cases} \tag{3.30}$$

We begin by proving, under the condition (i), the following lemma which gives an information geometric interpretation to the N-dimensional score distribution $p_i^{(t)}$:

Lemma 3.1. (Pythagorean theorem for N-dimensional score distributions) *If the condition (i) holds, then for any $i, j \in \{0, 1, \cdots, N\}$ and $i \neq j$, we have*

$$D(p_i^{(t)}||p_j^{(t)}) = D(p_i^{(t)}||p_0^{(t)}) + D(p_0^{(t)}||p_j^{(t)}), \tag{3.31}$$

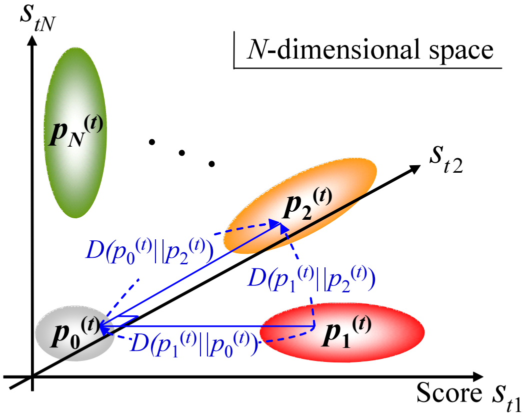


Figure 3.4: Relationship between the N-dimensional score distributions and the KL divergences.

where $D(p_i^{(t)}||p_j^{(t)})$ *is the KL (Kullback-Leibler) divergence [25] from* $p_i^{(t)}$ *to* $p_j^{(t)}$*:*

$$D(p_i^{(t)}||p_j^{(t)}) = \int p_i^{(t)}(\boldsymbol{s_t}) \log \frac{p_i^{(t)}(\boldsymbol{s_t})}{p_j^{(t)}(\boldsymbol{s_t})} d\boldsymbol{s_t}. \tag{3.32}$$

Proof. If $i = 0$ or $j = 0$, it is obvious that (3.31) holds since $D(p_0^{(t)}||p_0^{(t)}) = 0$. Otherwise (i.e. if $i, j \in \{1, 2, \cdots, N\}$ and $i \neq j$), we have

$$\begin{aligned}
D(p_i^{(t)}||p_j^{(t)}) &= \int p_i^{(t)}(\boldsymbol{s_t}) \log \frac{p_i^{(t)}(\boldsymbol{s_t})}{p_j^{(t)}(\boldsymbol{s_t})} d\boldsymbol{s_t} && (3.33)\\
&= \int\int f^{(t)}(s_{ti}) g_j^{(t)}(s_{tj}) \log \frac{f^{(t)}(s_{ti}) g_j^{(t)}(s_{tj})}{g_i^{(t)}(s_{ti}) f^{(t)}(s_{tj})} ds_{ti} ds_{tj} && (3.34)\\
&= \int f^{(t)}(s_{ti}) \log \frac{f^{(t)}(s_{ti})}{g_i^{(t)}(s_{ti})} ds_{ti} + \int g_j^{(t)}(s_{tj}) \log \frac{g_j^{(t)}(s_{tj})}{f^{(t)}(s_{tj})} ds_{tj} && (3.35)\\
&= \int p_i^{(t)}(\boldsymbol{s_t}) \log \frac{p_i^{(t)}(\boldsymbol{s_t})}{p_0^{(t)}(\boldsymbol{s_t})} d\boldsymbol{s_t} + \int p_0^{(t)}(\boldsymbol{s_t}) \log \frac{p_0^{(t)}(\boldsymbol{s_t})}{p_j^{(t)}(\boldsymbol{s_t})} d\boldsymbol{s_t} && (3.36)\\
&= D(p_i^{(t)}||p_0^{(t)}) + D(p_0^{(t)}||p_j^{(t)}). && (3.37)
\end{aligned}$$

From (3.33) to (3.34) and (3.35) to (3.36), we used (3.30) and the fact that $\int g_k^{(t)}(s_{tk}) ds_{tk} = 1$ for any $k \in \{1, 2, \cdots, N\}$ and $k \neq i, j$. □

The equation (3.31) is known as the *Pythagorean theorem for divergences* which holds when the primal geodesic (e-geodesic) connecting $p_i^{(t)}$ and $p_0^{(t)}$ is orthogonal at $p_0^{(t)}$ to the dual geodesic (m-geodesic) connecting $p_0^{(t)}$ and $p_j^{(t)}$ [3]. That is, what we proved here is that this orthogonality holds for the N-dimensional score distributions. Figure 3.4 shows the relationship between the N-dimensional score distributions and the KL divergences.

We further assume that the following condition holds:

(ii) The KL divergence from $f^{(t)}$ to $g_i^{(t)}$ takes the value $D_{f\to g_i}$, and that from $g_i^{(t)}$ to $f^{(t)}$ takes $D_{g_i\to f}$, both of which are independent of the modality:

$$\begin{aligned} D_{f\to g_i} &= D(f^{(1)}||g_i^{(1)}) = \cdots = D(f^{(T)}||g_i^{(T)}) \quad (1 \leq i \leq N) & (3.38)\\ D_{g_i\to f} &= D(g_i^{(1)}||f^{(1)}) = \cdots = D(g_i^{(T)}||f^{(T)}) \quad (1 \leq i \leq N). & (3.39)\end{aligned}$$

Here we explain the meaning of this condition. The KL divergence has a meaning of a distance measure between two probability distributions, and some studies proposed to use the KL divergence between a genuine distribution and an impostor distribution as a metric of identification performance [105, 108]. Thus, the condition (ii) means that the identification accuracy with regard to the i-th enrollee is independent of the modality. Note that this may be too restrictive, especially in the case of multiple biometric traits (e.g. fingerprint, face and iris), as will be discussed in Section 3.5.3. We advance discussions under this ideal condition.

Under the condition (ii), we can prove the following lemma:

Lemma 3.2. *If the condition (i) and (ii) hold, then for any $i, j \in \{0, 1, \cdots, N\}$ and $i \neq j$, the KL divergence from $p_i^{(t)}$ to $p_j^{(t)}$ takes the value $D_{p_i\to p_j}$ independent of the modality:*

$$D_{p_i\to p_j} = D(p_i^{(1)}||p_j^{(1)}) = \cdots = D(p_i^{(T)}||p_j^{(T)}). \qquad (3.40)$$

Proof. If $i, j \in \{1, 2, \cdots, N\}$ and $i \neq j$, (3.35) can be further written as follows:

$$\begin{aligned} D(p_i^{(t)}||p_j^{(t)}) &= \int f^{(t)}(s_{ti}) \log \frac{f^{(t)}(s_{ti})}{g_i^{(t)}(s_{ti})} ds_{ti} + \int g_j^{(t)}(s_{tj}) \log \frac{g_j^{(t)}(s_{tj})}{f^{(t)}(s_{tj})} ds_{tj} & (3.41)\\ &= D(f^{(t)}||g_i^{(t)}) + D(g_j^{(t)}||f^{(t)}) & (3.42)\\ &= D_{f\to g_i} + D_{g_j\to f}. & (3.43)\end{aligned}$$

From (3.42) to (3.43), we used (3.38) and (3.39). Thus, (3.40) holds in this case ($D_{p_i\to p_j} = D_{f\to g_i} + D_{g_j\to f}$). Similarly, if $i = 0$, we have

$$\begin{aligned} D(p_i^{(t)}||p_j^{(t)}) &= \int g_j^{(t)}(s_{tj}) \log \frac{g_j^{(t)}(s_{tj})}{f^{(t)}(s_{tj})} ds_{tj} & (3.44)\\ &= D(g_j^{(t)}||f^{(t)}) & (3.45)\\ &= D_{g_j\to f}, & (3.46)\end{aligned}$$

and (3.40) holds ($D_{p_i\to p_j} = D_{g_j\to f}$). If $j = 0$, we have

$$\begin{aligned} D(p_i^{(t)}||p_j^{(t)}) &= \int f^{(t)}(s_{ti}) \log \frac{f^{(t)}(s_{ti})}{g_i^{(t)}(s_{ti})} ds_{ti} & (3.47)\\ &= D(f^{(t)}||g_i^{(t)}) & (3.48)\\ &= D_{f\to g_i}, & (3.49)\end{aligned}$$

and (3.40) holds ($D_{p_i\to p_j} = D_{f\to g_i}$). □

Finally, we show the relationship between the log-likelihood ratio $\log p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_j)$ and the KL divergence $D(p_i^{(t)}||p_j^{(t)})$, in the case where H_i is a true hypothesis ($0 \leq i, j \leq N$, $i \neq j$). We simply denote the log-likelihood ratio $\log p_i^{(t)}(\boldsymbol{s_t})/p_j^{(t)}(\boldsymbol{s_t})$ in this case by LLR_t and its variance by $\sigma_t^2 = V(LLR_t)$, and assume the following condition about the finiteness of the variances: $\sum_{t=1}^{\infty} \sigma_t^2/t^2 < \infty$. Then, the following lemma holds:

Lemma 3.3. *Assume that H_i is a true hypothesis, where $i \in \{0, 1, \cdots, N\}$. If the condition (i) and (ii) hold, then for any $j \in \{0, 1, \cdots, N\}$ and $i \neq j$, we have*

$$\frac{1}{t} \log p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_j) \xrightarrow{a.s.} D_{p_i \to p_j} \quad (t \to \infty), \tag{3.50}$$

where $X_t \xrightarrow{a.s.} X$ $(t \to \infty)$ represents that a sequence of random variables $\{X_t\}$ converges to X almost surely (i.e. almost sure convergence):

$$Pr(\lim_{t \to \infty} X_t = X) = 1. \tag{3.51}$$

Proof. If the condition (i) holds, the left side of (3.50) can be written as follows:

$$\begin{aligned} \frac{1}{t} \log p(\boldsymbol{S_t}|H_i)/p(\boldsymbol{S_t}|H_j) &= \frac{1}{t} \log \prod_{\tau=1}^{t} \frac{p_i^{(\tau)}(\boldsymbol{s_\tau})}{p_j^{(\tau)}(\boldsymbol{s_\tau})} &(3.52)\\ &= \frac{1}{t} \sum_{\tau=1}^{t} \log \frac{p_i^{(\tau)}(\boldsymbol{s_\tau})}{p_j^{(\tau)}(\boldsymbol{s_\tau})} &(3.53)\\ &= \frac{1}{t} \sum_{\tau=1}^{t} LLR_\tau. &(3.54)\end{aligned}$$

By Lemma 3.2, if (i) and (ii) hold, we have

$$D_{p_i \to p_j} = D(p_i^{(t)}||p_j^{(t)}) = \int p_i^{(t)}(\boldsymbol{s_t}) \log \frac{p_i^{(t)}(\boldsymbol{s_t})}{p_j^{(t)}(\boldsymbol{s_t})} d\boldsymbol{s_t} = E(LLR_t). \tag{3.55}$$

Thus, in this case, $\{LLR_t\}$ is a sequence of independent random variables with mean $D_{p_i \to p_j} (= E(LLR_t))$ and variance $\sigma_t^2 (= V(LLR_t))$ such that $\sum_{t=1}^{\infty} \sigma_t^2/t^2 < \infty$. Then, by Kolmogorov's strong law of large numbers [59], we have

$$\frac{1}{t} \sum_{\tau=1}^{t} LLR_\tau \xrightarrow{a.s.} D_{p_i \to p_j} \quad (t \to \infty). \tag{3.56}$$

By (3.54) and (3.56), (3.50) holds. □

3.5.2 Proof of the Optimality of the LRSI Scheme

We now prove the optimality of the LRSI scheme using the properties of N-dimensional score distributions (Lemma 3.1, 3.2, and 3.3). To prove the optimality, we further assume that the following conditions hold:

(iii) The LRSI scheme can perfectly estimate the likelihood ratio $f^{(t)}/g_i^{(t)}$ (or the log-likelihood ratio $\log f^{(t)}/g_i^{(t)}$) $(1 \leq t \leq T,\ 1 \leq i \leq N)$.

(iv) The prior distribution $P(H_i)$ $(0 \leq i \leq N)$ is uniform.

Under the conditions, we prove that the LRSI scheme is equivalent to MSPRT (Test δ_b) which sets the required identification error probabilities to be $\overline{\alpha}_1 = \cdots = \overline{\alpha}_N$ and $\overline{\alpha}_0 = 0$[2], in the asymptotic case where the number of inputs t goes to ∞ (the optimality of MSPRT's is also proved in this asymptotic case, as described in Section 3.3).

Under the condition (iv), the generalized likelihood ratio L_{ti} $(1 \leq i \leq N)$ used in MSPRT (Test δ_b) (see (3.22)) can be written as follows:

$$L_{ti} = \min_{0 \leq n \leq N, n \neq i} \frac{p(\boldsymbol{S_t}|H_i)}{p(\boldsymbol{S_t}|H_n)}. \tag{3.57}$$

We can also obtain $B_1 = B_2 = \cdots = B_N$ and $B_0 = \infty$ from (3.24), $\overline{\alpha}_1 = \cdots = \overline{\alpha}_N$, $\overline{\alpha}_0 = 0$, and the assumption that $P(H_i)$ $(0 \leq i \leq N)$ is uniform.

On the other hand, under the conditions (i) and (iii), the LRSI scheme perfectly estimates the likelihood ratio Z_{ti} between H_i and H_0 $(1 \leq i \leq N)$ which is given by

$$Z_{ti} = \frac{p(\boldsymbol{S_t}|H_i)}{p(\boldsymbol{S_t}|H_0)}, \tag{3.58}$$

and compares it to the common threshold B.

Let $i^* \in \{1, 2, \cdots, N\}$ be a hypothesis ID whose likelihood is the highest:

$$i^* = \operatorname*{arg\,max}_{1 \leq i \leq N} p(\boldsymbol{S_t}|H_i). \tag{3.59}$$

Then, we can say that the LRSI scheme and MSPRT (Test δ_b) identify the claimant by comparing Z_{ti^*} and L_{ti^*} to the threshold, respectively. Thus, to prove that the LRSI scheme is equivalent to MSPRT (Test δ_b), it suffices to prove that n which minimizes the likelihood ratio $p(\boldsymbol{S_t}|H_{i^*})/p(\boldsymbol{S_t}|H_n)$ $(0 \leq n \leq N,\ n \neq i^*)$ is $n = 0$. We prove this in the asymptotic case where the number of inputs t goes to ∞:

Theorem 3.1. (Equivalence between the LRSI scheme and MSPRT (Test δ_b)) *If the conditions (i), (ii), (iii), and (iv) hold, then*

$$\operatorname*{arg\,min}_{0 \leq n \leq N, n \neq i^*} \frac{p(\boldsymbol{S_t}|H_{i^*})}{p(\boldsymbol{S_t}|H_n)} \xrightarrow{a.s.} 0 \quad (t \to \infty), \tag{3.60}$$

and thus the LRSI scheme is equivalent to MSPRT (Test δ_b) for sufficiently large t.

Proof. Let $k \in \{0, 1, \cdots, N\}$ be a true hypothesis ID. We first prove this theorem in the case where $k \in \{1, 2, \cdots, N\}$. In this case, since the likelihood of the true hypothesis $p(\boldsymbol{S_t}|H_k)$ is the highest for sufficiently large t, $i^* \xrightarrow{a.s.} k$ $(t \to \infty)$. Thus, we have

$$\begin{aligned} \frac{1}{t} \log \frac{p(\boldsymbol{S_t}|H_{i^*})}{p(\boldsymbol{S_t}|H_n)} &\to D_{p_k \to p_n} \quad (t \to \infty) & (3.61) \\ &= D_{p_k \to p_0} + D_{p_0 \to p_n}. & (3.62) \end{aligned}$$

[2] It follows from $\overline{\alpha}_0 = 0$ and (3.24) that $B_0 = \infty$. Thus, $\overline{\alpha}_0 = 0$ means that the system does not *actively* reject a claimant until the number of inputs exceeds an upper limit T. Although it causes an increase of *the number of inputs of non-enrollees* (i.e. attackers), we consider this is not a problem as long as *the number of inputs of enrollees* is small.

In (3.61), we used $i^* \xrightarrow{a.s.} k$ $(t \to \infty)$ and Lemma 3.3. From (3.61) to (3.62), we used Lemma 3.1 and 3.2. It follows from the non-negativity of the KL divergence [25] that the value in (3.62) is minimized in the case where $n = 0$.

We then prove this theorem in the case where $k = 0$. In this case, we have

$$\begin{aligned} \frac{1}{t}\log\frac{p(\boldsymbol{S_t}|H_{i^*})}{p(\boldsymbol{S_t}|H_n)} &= \frac{1}{t}\log\frac{p(\boldsymbol{S_t}|H_{i^*})p(\boldsymbol{S_t}|H_0)}{p(\boldsymbol{S_t}|H_0)p(\boldsymbol{S_t}|H_n)} && (3.63)\\ &= -\frac{1}{t}\log\frac{p(\boldsymbol{S_t}|H_0)}{p(\boldsymbol{S_t}|H_{i^*})} + \frac{1}{t}\log\frac{p(\boldsymbol{S_t}|H_0)}{p(\boldsymbol{S_t}|H_n)} && (3.64)\\ &\to -D_{p_0 \to p_{i^*}} + D_{p_0 \to p_n}. && (3.65) \end{aligned}$$

From (3.64) to (3.65), we used Lemma 3.3. From the non-negativity of the KL divergence, the value in (3.65) is minimized in the case where $n = 0$. □

What we proved here is that the LRSI scheme is equivalent to MSPRT (Test δ_b) in the asymptotic case where $t \to \infty$. Since MSPRT (Test δ_b) is optimal with regard to the trade-off between the average number of observations and the identification error probabilities (i.e. $\alpha_0, \cdots, \alpha_N$), the LRSI scheme is also optimal with regard to the trade-off between ANI and the identification error probabilities (EFRP/EFAP/NFAP).

It should be noted that although Noda and Kawaguchi [78] proposed a feature level approach which computes, for each enrollee, the likelihood ratio against the hypothesis H_0 in the same way as the LRSI scheme to reduce the utterance length in speaker identification, we did not prove that their technique can minimize the average utterance length. This is because we used some properties of *not feature distributions but N-dimensional score distributions* (Lemma 3.1, 3.2, and 3.3) in proving Theorem 3.1.

3.5.3 Conditions to Achieve the Optimality

In Section 3.5.2, we proved the optimality of the LRSI scheme under the following conditions:

(i) All scores are independent, and scores between the same individual and scores between the i-th enrollee and another person are generated from $f^{(t)}()$ and $g_i^{(t)}()$, respectively, where t is the input number.

(ii) The KL divergence from $f^{(t)}$ to $g_i^{(t)}$ takes the value $D_{f \to g_i}$, and that from $g_i^{(t)}$ to $f^{(t)}$ takes $D_{g_i \to f}$, both of which are independent of the modality.

(iii) The LRSI scheme can perfectly estimate the likelihood ratio $f^{(t)}/g_i^{(t)}$ (or the log-likelihood ratio $\log f^{(t)}/g_i^{(t)}$) $(1 \leq t \leq T,\ 1 \leq i \leq N)$.

(iv) The prior distribution $P(H_i)$ $(0 \leq i \leq N)$ is uniform.

(v) The identification error probabilities (EFRP/EFAP/NFAP) are sufficiently small (the number of inputs t is sufficiently large).

The last condition follows from the fact that the asymptotic case where $t \to \infty$ is equivalent to the asymptotic case where the required identification error probabilities go to 0 ($\overline{\alpha}_{max}(= \max_{0 \leq i \leq N} \overline{\alpha}_i) \to 0$). The condition (i), (iii), (iv), and (v) are the same for the optimality of the PPSI scheme described in Section 3.4.1 and 3.4.2 since it also assumes (i) and (iv), estimates the likelihood ratio $f^{(t)}()/g_i^{(t)}()$ in (3.27), and MSPRT (Test δ_a) is also optimal in the asymptotic case where $t \to \infty$.

Unfortunately, the above conditions are not satisfied in reality. We describe, for each condition, this in detail below:

- Although we assume the enrollee-specific impostor distribution model, the condition (i) is still not satisfied because impostor scores also vary depending on the claimant. For example, there exist claimants who have high similarity scores against many enrollees, also known as *wolves*. This indicates that an impostor score is *not independently distributed* from each of the enrollee-specific impostor distributions. Similarly, genuine scores also vary depending on the enrollee (e.g. there exist enrollees who have low similarity scores against themselves, also known as *goats*).

- The condition (ii) is also not satisfied because the KL divergence, which is used as a metric of identification performance [105, 108] (as described in Section 3.5.1), is different from modality to modality. Although it is not much different in the case of multiple instances (e.g. index and middle fingers), it can be significantly different in the case of multiple biometric traits (e.g. fingerprint, face and iris).

- It is obvious that the condition (iii) is not satisfied: it is impossible to *perfectly* estimate $f^{(t)}/g_i^{(t)}$ or $\log f^{(t)}/g_i^{(t)}$ in general. In our experiment in Section 3.6, we used logistic regression and confirmed that it worked very well though not perfect.

- It is also obvious that the condition (iv) is not satisfied: the prior distribution $P(H_i)$ $(0 \leq i \leq N)$ can be non-uniform. As described in Section 3.4.2, since it is generally very difficult to correctly estimate $P(H_i)$, we use a *noninformative prior* (i.e. $P(H_0) = \cdots = P(H_N) = 1/(N+1)$) which have little influence on the posterior distribution.

- Finally, since the number of inputs t is generally very small (e.g. 1, 2, or 3), the condition (v) is not satisfied in such a case.

The optimality of our proposals is not proved without the conditions (i)-(v). Thus, we need to show the effectiveness of our proposals through experimental evaluation, which is described in the next section.

3.6 Experimental Evaluation

3.6.1 Experimental Set-up

We evaluated our proposals using the NIST BSSR1 Set1 dataset [77] which contains scores of faces, left index fingerprints, and right index fingerprints obtained from 517 subjects. The reason we used this dataset is because they are freely available and relatively large-scale (many other datasets contain fewer subjects [8, 66, 72, 86]). We also considered that fingerprints and faces account for the highest and the second highest share in the world market, respectively [69]. Although there were face scores obtained using two algorithms ("C" and "G"), we adopted ones obtained using the algorithm C. Then we excluded one person who has inappropriate scores (the values "-1"), and extracted 3 × 516 × 516 scores.

We randomly selected 200 enrollees ($N = 200$) and 200 non-enrollees from 516 subjects and used the remaining 116 subjects for training the log-likelihood ratio $\log f^{(t)}/g_i^{(t)}$. Here, we assumed the logistic regression model and trained the regression coefficients using the BRLR (Bias-Reduced Logistic Regression) package for R [80]. For training data, we used 116 genuine scores from the training subjects and 116 impostor scores between the training subjects and the i-th enrollee. We tried 10 ways of randomly selecting 200 enrollees and 200 non-enrollees, and carried out the experiment where each of the enrollees and non-enrollees sequentially inputs his/her biometric samples. As for the input order, we tried all of 6 (= 3!) ways. In this experiment, both the number of accesses by enrollees and that of attacks by non-enrollees were 12000 (= 200 × 10 × 6).

For comparison, we evaluated the performance of the following schemes:

1. **OR**: the OR rule [14]. Since the OR rule was originally proposed in the verification scenario, we extended it to the identification scenario as follows: if one or more scores (which are normalized to the FAR values) fall below the threshold, identify as the enrollee whose score is the lowest; otherwise, require another biometric input.

2. **LRSI**: the LRSI scheme described in Section 3.4.3. That is, after the t-th query sample is input, it computes the log-likelihood $\log Z_{ti}$ $(1 \leq i \leq N)$ using a set of score vectors $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$, and compares the maximum of $\log Z_{ti}$ to the threshold.

3. **PPSI**: the PPSI scheme described in Section 3.4.1 and 3.4.2. That is, after the t-th query sample is input, it computes the posterior probability $P(H_i|\boldsymbol{S_t})$ $(0 \leq i \leq N)$ using a set of score vectors $\boldsymbol{S_t} = \{\boldsymbol{s_\tau} | 1 \leq \tau \leq t\}$, and compares the maximum of $P(H_i|\boldsymbol{S_t})$ to the threshold. We assumed that the prior probability $P(H_i)$ $(0 \leq i \leq N)$ is uniform.

Both of the LRSI scheme and the PPSI scheme described above estimated the log-likelihood ratio $\log f^{(t)}/g_i^{(t)}$ using the logistic regression model. To demonstrate the effectiveness of the logistic regression model, we also evaluated the performances of the two schemes in the case where we assumed the Gaussian distribution models as $f^{(t)}$ and $g_i^{(t)}$. In this case, we estimated $f^{(t)}$ using 116 genuine scores from the training subjects, and $g_i^{(t)}$ using 116 impostor scores between the training subjects and the i-th enrollee. We adopted the maximum likelihood estimator as an estimation method.

Similarly, to demonstrate the effectiveness of the enrollee-specific impostor distribution model, we also evaluated the performances of the LRSI scheme and the PPSI scheme in the case where we used the impostor distribution common to all enrollees $g^{(t)}$, and estimated the log-likelihood ratio $\log f^{(t)}/g^{(t)}$ using the logistic regression model. In this case, we used 116 genuine scores and 116 × 115 impostor scores from the training subjects as training data.

3.6.2 Experimental Results

Figure 3.5 shows the relationship between ANI (the average number of inputs) and EFAR/ NFAR. Here we fixed EFRR of all the schemes to be 2[%] by changing the identification threshold at the last input from the one until the second last input[3]. In this figure, "Common", "Specific", "Gaussian", and "Logistic" represent the case where we used an impostor distribution common to all enrollees, enrollee-specific impostor distributions, the Gaussian

[3] A variant of SPRT which forcibly outputs accept/reject at the T-th observation using a threshold different from the ones until the $(T-1)$-th observation is referred to as the *truncated* SPRT [121]. By changing the last threshold, we can control the trade-off between EFRR and EFAR/NFAR without affecting ANI.

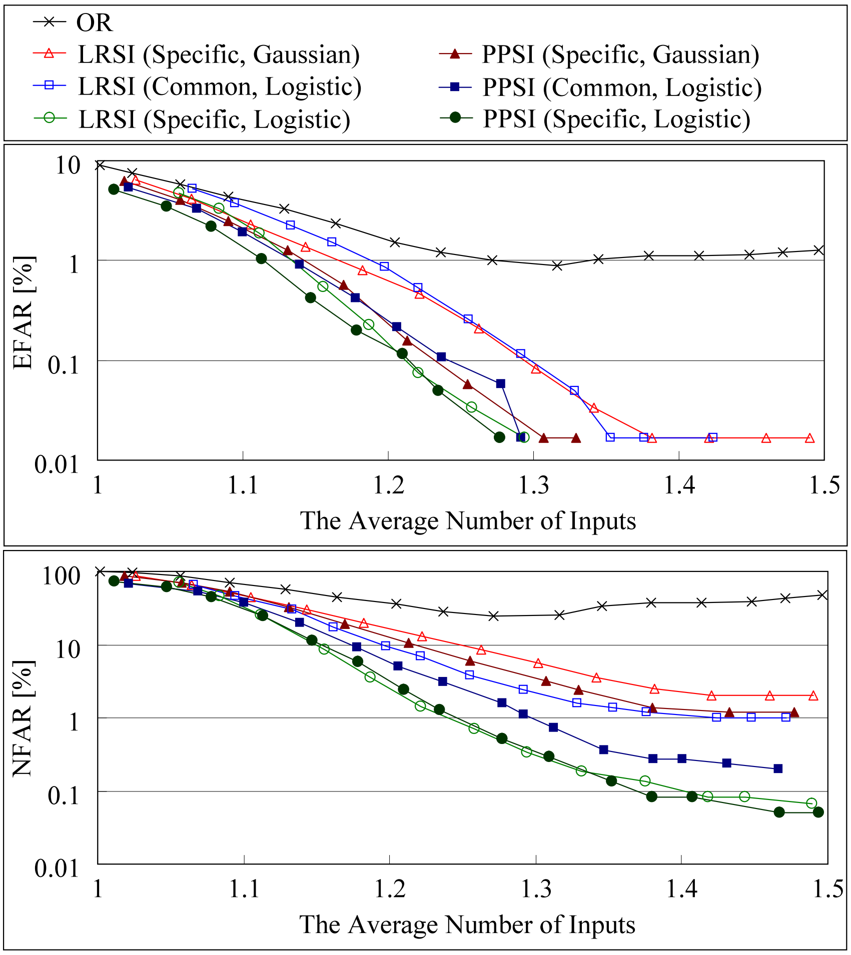
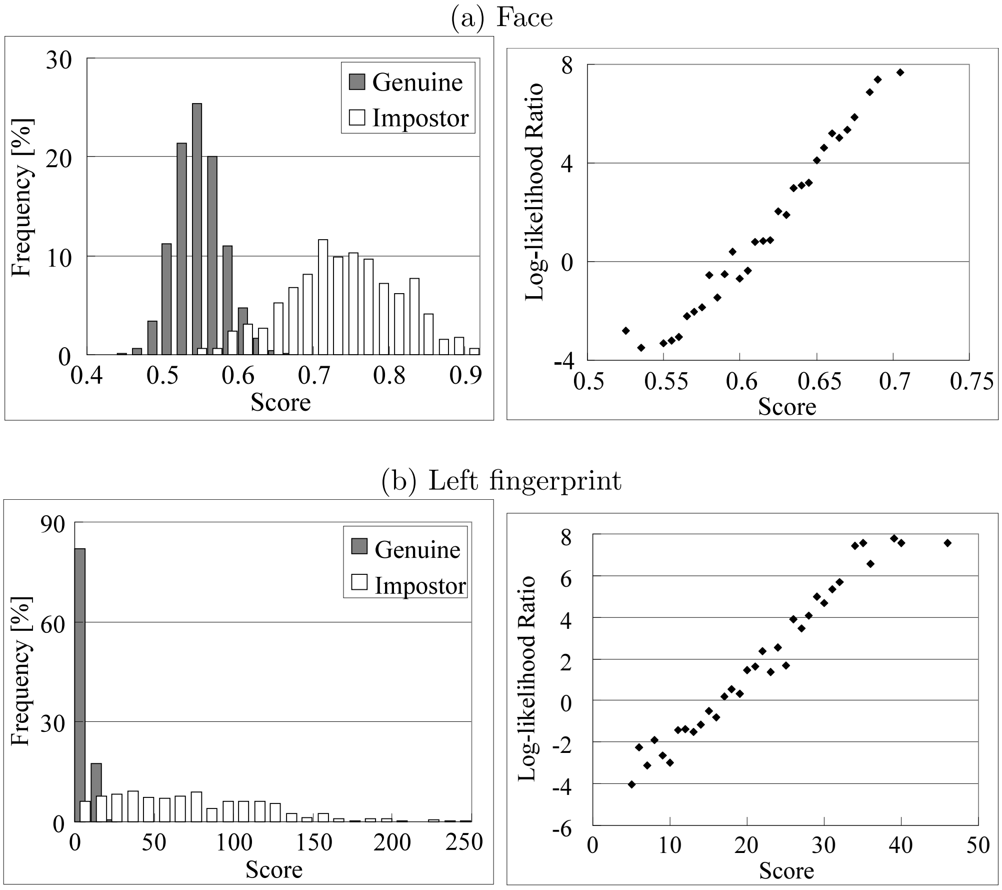


Figure 3.5: Relationship between the average number of inputs and EFAR/NFAR.

distribution model, and the logistic regression model, respectively. In LRSI (Specific, Logistic), PPSI (Specific, Gaussian), PPSI (Common, Logistic), and PPSI (Specific, Logistic), EFAR was 0% when ANI was more than about 1.3.

It was found from Figure 3.5 that our proposals (the LRSI scheme and the PPSI scheme) significantly outperformed the OR rule with regard to both ANI and EFAR/NFAR, demonstrating that our proposals are really effective. It was also found that the enrollee-specific impostor distributions (Specific, Logistic) provided the better performance than the impostor distibution common to all enrollees (Common, Logistic). Similarly, the logistic regression model (Specific, Logistic) outperformed the Gaussian distribution model (Specific, Gaussian).

To show the validity of the logistic regression model, we show in Figure 3.6 a frequency distribution of genuine/impostor scores and the logarithm of their ratio in the faces and left fingerprints (we omitted the result of the right fingerprints because it was similar to that of the left fingerprints). Here we computed the genuine frequency distributions and impostor frequency distributions using 516 genuine scores and 516 × 515 impostor scores, respectively. It can be seen that the frequency distributions of the left fingerprints have complex shapes, while there is a close-to-linear relationship between the logarithm of their ratio and a score. The logistic regression model does not have to estimate the genuine/impostor distribution


(a) Face

(b) Left fingerprint

Figure 3.6: Frequency distribution of genuine/impostor scores and the logarithm of their ratio.

itself, but models the logarithm of their ratio as a linear function of a score. This is the reason that the logistic regression model provided better identification performance.

We also obtained an interesting result from Figure 3.5 that the PPSI scheme outperformed the LRSI scheme in almost all cases. We consider this is because there existed wolves who have high similarity scores against many enrollees in this dataset (i.e. the condition (i) in Section 3.5.3 was not satisfied), and the security against wolves was improved by normalizing likelihood ratios to posterior probabilities. Since wolves have high similarity scores against many enrollees, they also have high likelihood ratios against them. However, they have low posterior probabilities against them because the summation of posterior probabilities is 1. For example, even if high likelihood ratios are equally obtained against M $(2 \leq M \leq N)$ enrollees, posterior probabilities against them are about $1/M$. Thus, if the identification threshold is set to be more than 0.5, the identification error does not happen in this case. In [A5, C5], we also showed that the maximum of the claimant-specific FAR (i.e. FAR caused by the most threatening wolf) was significantly reduced by normalizing likelihood ratios to posterior probabilities through experimental evaluation.

In addition to the performance shown in Figure 3.5, we evaluated the performance of the uni-modal biometric system which identifies claimants using only the first query sample. The result was poor: [EFAR, NFAR]=[7.6%, 89%]. The performance was significantly improved by applying our proposals: when ANI was fixed to be 1.4, EFAR/NFAR in the OR rule,

LRSI (Specific, Logistic), and **PPSI (Specific, Logistic)** were [1.1%, 38%], [0%, 0.10%], and [0%, 0.083%], respectively. We finally evaluated the performance of the Bayes decision rule-based parallel fusion scheme in identification [76] which compares posterior probabilities to a threshold after the claimant inputs all query samples (described in Section 2.1.3). The result was [EFAR, NFAR] = [0%, 0.050%]. The PPSI scheme reduced ANI from 3 to 1.47 without increasing the above error rates. That is, more than half of the query samples were not required in the PPSI scheme.

3.7 Conclusions

In this chapter, we proposed the LRSI scheme and PPSI scheme as sequential fusion schemes in biometric identification. We clarified the optimal property of these schemes and showed their effectiveness through experimental evaluation.

In Section 3.5.3, we discussed the conditions to achieve the optimality of the LRSI scheme and PPSI scheme. The proof of the optimality without the conditions (i), (ii), (iii), (iv), and (v), or developing such sequential fusion schemes is one of our major future work. For example, the optimality of SPRT is proved even if the number of observations t is small [71, 122]. This may be helpful in proving the optimality of our schemes without the condition (v).

It should also be noted that the "optimality" in this chapter means the optimality with regard to the trade-off between ANI (the average number of inputs) and the identification error probabilities (EFRP/EFAP/NFAP). In Chapter 4 and 5, we make an attempt to further optimize the response time. In Chapter 6, we attempt to optimize the trade-off between security against wolves and lambs and convenience in terms of the number of inputs and false rejects.

Chapter 4

Towards Optimal Metric Space Indexing Methods

Contents

4.1 Introduction

Metric space indexing methods (also known as metric access methods, or distance-based indexing methods) have been developed in the area of similarity search [23, 82]. They reduce the number of score computations at query time, using the index which is constructed based on scores. Since score computations are generally expensive, they can find a set of objects in the database whose distance (or similarity) to the query object is less than (or more than) the threshold (i.e. answer to the range query) or the k-nearest neighbor objects (i.e. answer to the k-NN query), faster than the sequential scan which computes scores for all the objects.

Since they can be applied as long as a score measure (i.e. a measure of distances or similarities between objects) is defined, they have a variety of applications which retrieve complex data such as audio, images, videos, documents, and biometrics.

Although a considerable number of metric space indexing methods have been proposed as described in Section 2.2.2, we focus on *pseudo-score based indexing schemes* [4, 6, 22, 32] which fall under the category of approximate indexing schemes. They compute, for each object in the database, a *pseudo-score* which is easily computed and highly relevant to a score, and compute scores in order of the pseudo-score. By stopping searching at some halting point, they reduce the number of score computations. However, since they do not search the remaining objects, they may fail to search an object in the correct answer. In this dissertation, we refer to the errors in which the system fails to search an object in the correct answer as *retrieval errors*[1]. They can control, by adjusting the halting point, the trade-off between the number of score computations required and retrieval errors.

The permutation-based indexing scheme [4, 22], one of the pseudo-score based schemes, is known as one of the most successful metric space indexing schemes. At indexing time, this scheme first selects some objects (referred to as *pivots*) from the database. Then, it computes, for each of the remaining objects (referred to as *non-pivots*), a permutation where the pivot IDs are written in ascending (or descending) order of distances (or similarities) to the object. At query time, it computes the rank correlation between the permutation of the query object and that of the object in the database as a pseudo-score. It was shown that this scheme outperformed other schemes with regard to the trade-off between the number of score computations and retrieval errors, in some cases by a wide margin.

Another example of pseudo-score based schemes is the distance regression-based indexing scheme proposed by Edsberg and Hetland [32] which outperformed the permutation-based scheme in some cases, and did not in other cases in their experiment. At indexing time, this scheme computes the regression coefficients in a linear regression model. At query time, it computes, for each object, using the distances between the query object and all pivots, either of the following as a pseudo-score: the estimate of the distance to the query object or the probability of being in the answer to the range query. In this dissertation, we refer to the former as a *distance-based pseudo-score* and the latter as a *probability-based pseudo-score*. The distance regression-based scheme computes distance-based pseudo-scores by assuming the linear regression model, and computes probability-based pseudo-scores by further assuming that the estimation errors are normally distributed.

Probability-based pseudo-scores make it possible to compare the objects in descending order of the probability of being in the answer to the range query. Thus, they minimize, for any given halting point, the expected number of retrieval errors if the probability is correctly estimated. In other words, they can optimize the trade-off between the number of score computations and retrieval errors. However, they performed worse than the distance-based pseudo-scores in their experiment, which indicates the further assumption about the estimation errors may not be appropriate.

[1] The performance of information retrieval systems is often evaluated using two measures: recall and precision. The recall is the ratio tp / (tp + fn), where tp and fn are the number of true positives and false negatives, respectively. The precision is the ratio tp / (tp + fp), where fp is the number of false positives. Then, the *retrieval error rate* can be expressed as 1− recall. Note that *the precision of metric space indexing schemes is always* 1 *in range queries* which we focus on in this chapter since there is no false positives.

4.1.1 Our Contributions

The motivation of our work is based on the fact that (1) some pseudo-score based schemes provide a very good trade-off between the number of score computations and retrieval errors and (2) probability-based pseudo-scores can provide an optimal trade-off in range queries if the probabilities are correctly estimated. Taking these matters into account, we focus on range queries in similarity search, and attempt to develop metric space indexing schemes which have an optimal property with regard to the above trade-off, and outperform the state-of-the-art metric space indexing schemes such as the permutation-based indexing scheme [4, 22]. The main contributions are as follows:

1. We firstly propose the PPS (Posterior Probability-based Search) scheme which normalizes pseudo-scores to the posterior probabilities of being in the answer to the range query, and uses them as probability-based pseudo-scores to search non-pivots. The difference of our scheme from the approach taken by Edsberg and Hetland [32] is *general applicability:* since our scheme normalizes pseudo-scores to probability-based ones, it can be applied to any pseudo-score based scheme to enhance its performance. We also clarify the optimal property of the PPS scheme with regard to the number of score computations and retrieval errors.

2. We secondly propose an algorithm which computes the probability-based pseudo-scores using the object-specific parameters in logistic regression [12], and learns the parameters using MAP (Maximum a Posteriori) estimation [12]. This is a simple way to correctly estimate the probability-based pseudo-scores.

3. We thirdly propose a technique which speeds up learning the parameters using pseudo-scores. This technique significantly reduces the time to learn the parameters while keeping the trade-off between the number of score computations and retrieval errors.

4. We finally apply our scheme to the two state-of-the-art schemes: the standard pivot-based indexing scheme [22] and the permutation-based indexing scheme [4, 22], and evaluate them using various kinds of datasets from the Metric Space Library [37]. The results show that our scheme outperforms the conventional schemes, with regard to both the number of score computations and the CPU time, in all the datasets.

4.1.2 Organization of This Chapter

This chapter is organized as follows. In Section 4.2, we introduce related work on pseudo-score based schemes. We first provide the framework of pseudo-score based indexing algorithms, and then explain the algorithm of the standard pivot-based scheme [22] and that of the permutation-based scheme [4, 22]. We also review other pseudo-score based schemes. In Section 4.3, we propose the PPS scheme. We first explain the search algorithm of our scheme, clarifying its optimality with regard to the trade-off between the number of score computations and retrieval errors. We then explain the learning algorithm and the technique which speeds up learning the parameters. In Section 4.4, we show the experimental results using various kinds of datasets, and discuss the results. Finally, we conclude this chapter in Section 4.5.

Table 4.1: Basic Notations Used in This Chapter.

Symbol	Description
$\mathbb{X}$	universe of objects
q	query object $q \in \mathbb{X}$
$\mathbb{O}$	set of objects in the database $\mathbb{O} = \{o_i \| 1 \leq i \leq N\} \subset \mathbb{X}$
$\mathbb{O}_{piv}$	set of pivots $\mathbb{O}_{piv} = \{o_i \| 1 \leq i \leq K\}$
$\mathbb{O}_{non}$	set of non-pivots $\mathbb{O}_{non} = \{o_i \| K+1 \leq i \leq N\}$
$\mathbb{S}_{piv}$	set of scores for pivots $\mathbb{S}_{piv} = \{s(q, o_i) \| o_i \in \mathbb{O}_{piv}\}$
$\tilde{\mathbb{S}}_{non}$	set of pseudo-scores for non-pivots $\tilde{\mathbb{S}}_{non} = \{\tilde{s}(q, o_i) \| o_i \in \mathbb{O}_{non}\}$
N	number of objects in the database
K	number of pivots
M	number of score computations at query time ($K \leq M \leq N$)
s	score function $s : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$
$\tilde{s}$	pseudo-score function $\tilde{s} : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$

4.2 Pseudo-score Based Indexing Schemes

4.2.1 Pseudo-score Based Indexing Framework

Pseudo-score based indexing schemes are approximate indexing schemes which reduce the number of score computations using pseudo-scores, which are easily computed and highly relevant to scores, as a clue. Their examples include the standard pivot-based indexing scheme [22], the permutation-based indexing scheme [4, 22], the distance regression-based indexing scheme [32], and BoostMAP [6]. Although they are independently proposed, we provide a generalized framework of pseudo-score based indexing schemes to describe them in a unified point of view.

We now explain the pseudo-score based indexing framework using the notations shown in Table 4.1. Let $\mathbb{X}$ be a universe of objects, $q \in \mathbb{X}$ be a query object, and $\mathbb{O} = \{o_i | 1 \leq i \leq N\} \subset \mathbb{X}$ be a finite set of objects in the database. Let further $s : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ be a score function, and $\tilde{s} : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ be a pseudo-score function. Pseudo-scores are either positively or negatively correlated with scores.

To begin with, they construct an index $\mathbb{I}$ in advance, which is used to compute pseudo-scores at query time, as follows:

[Indexing Algorithm]

1. Select K ($< N$) objects from $\mathbb{O}$ (at random [22], for example). The selected objects are referred to as *pivots*, and the remaining objects are referred to as *non-pivots*. We assume that $\mathbb{O}_{piv} = \{o_i | 1 \leq i \leq K\}$ is a set of pivots and $\mathbb{O}_{non} = \{o_i | K+1 \leq i \leq N\}$ is a set of non-pivots, without loss of generality.
2. Compute a score $s(o_i, o_j)$ for each pivot $o_i \in \mathbb{O}_{piv}$ and each non-pivot $o_j \in \mathbb{O}_{non}$. Let $\mathbb{S}_{pnp} = \{s(o_i, o_j) | o_i \in \mathbb{O}_{piv}, o_j \in \mathbb{O}_{non}\}$ be a set of scores between pivots and non-pivots ($K(N-K)$ scores in total).
3. Construct an index $\mathbb{I}$ using $\mathbb{S}_{pnp}$, and store it in the database (the example of $\mathbb{I}$ is described later).

At query time, they carry out the search as follows:

[Search Algorithm]

1. Compute a score $s(q, o_i)$ for each pivot $o_i \in \mathbb{O}_{piv}$. Let $\mathbb{S}_{piv} = \{s(q, o_i) | o_i \in \mathbb{O}_{piv}\}$ be a set of scores for pivots (K scores in total).
2. Compute a pseudo-score $\tilde{s}(q, o_i)$ for each non-pivot $o_i \in \mathbb{O}_{non}$ using $\mathbb{S}_{piv}$ and $\mathbb{I}$ (the example of $\tilde{s}(q, o_i)$ is described later). Let $\tilde{\mathbb{S}}_{non} = \{\tilde{s}(q, o_i) | o_i \in \mathbb{O}_{non}\}$ be a set of pseudo-scores for non-pivots ($N - K$ pseudo-scores in total).
3. Compute scores for $M - K$ ($K \leq M \leq N$) non-pivots in ascending (or descending) order of pseudo-scores, where M represents the number of score computations at query time.
4. Output the answer of range queries (or k-NN queries).

The crucial point here is that they reduce the number of score computations at query time from N to M ($K \leq M \leq N$) by searching $M - K$ non-pivots in ascending (or descending) order of pseudo-scores, and stopping searching after that. Since pseudo-scores are designed to be highly relevant to scores, they can effectively narrow down the search by sorting non-pivots by pseudo-score. They can control the trade-off between the number of score computations and retrieval errors by changing the value of M.

4.2.2 Standard Pivot-based Indexing Scheme

The standard pivot-based indexing scheme [22] is the most basic type of pseudo-score based scheme. We now explain the algorithm of this scheme by clarifying the index $\mathbb{I}$ and the pseudo-score $\tilde{s}(q, o_i)$ in the above framework.

Let $\Phi : \mathbb{X} \rightarrow \mathbb{R}^K$ be a score vector function which maps the object $x \in \mathbb{X}$ to a score vector $(s(x, o_1), \cdots, s(x, o_K))$ which is composed of scores for pivots. In the step 3 of the indexing algorithm, this scheme simply stores a set of scores $\mathbb{S}_{pnp}$ in the database as an index. The index can then be expressed as a set of score vectors $\mathbb{I} = \{\Phi(o_i) | o_i \in \mathbb{O}_{non}\}$. In the step 2 of the search algorithm, it first generates a score vector $\Phi(q)$ by lining up scores in $\mathbb{S}_{piv}$, and then computes, for each non-pivot $o_i \in \mathbb{O}_{non}$, the L_p norm between $\Phi(q)$ and $\Phi(o_i)$ as a pseudo-score:

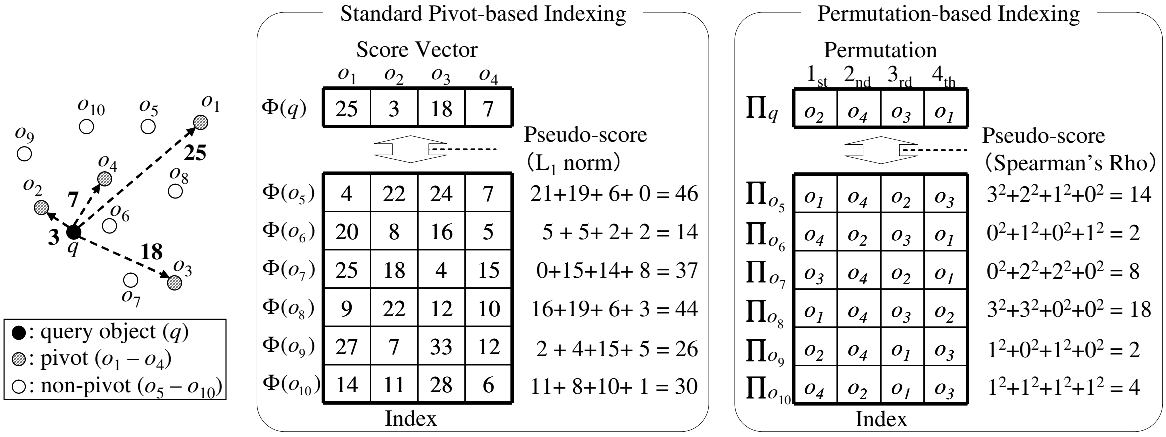
$$\begin{aligned} \tilde{s}(q, o_i) &= L_p(\Phi(q), \Phi(o_i)) && (4.1) \\ &= \left(\sum_{j=1}^{K} (s(q, o_j) - s(o_i, o_j))^p \right)^{1/p}. && (4.2) \end{aligned}$$

This scheme can be regarded as a modification of LAESA (Linear Approximating and Eliminating Search Algorithm) [73] to use the general L_p norm instead of L_∞ in the approximating step and to skip the elimination step using the triangle inequality.

Figure 4.1 shows the example of the standard pivot-based indexing scheme in the case where $K = 4$, $N = 10$, and the L_1 norm is used as a distance measure between two score vectors.

4.2.3 Permutation-based Indexing Scheme

The permutation-based indexing scheme was independently proposed by Chávez *et al.* [22] and Amato and Savino [4]. It computes a permutation, where the pivot IDs are written



o_{10}
o_5
o_1
o_9
o_4
o_8
25
o_2
7
o_6
3
q
18
o_3
o_7
●: query object (q)
◉: pivot ($o_1 - o_4$)
○: non-pivot ($o_5 - o_{10}$)
Standard Pivot-based Indexing
Score Vector
Pseudo-score (L_1 norm)
Index
Permutation-based Indexing
Permutation
Pseudo-score (Spearman's Rho)
Index

	o_1	o_2	o_3	o_4	
$\Phi(q)$	25	3	18	7	
$\Phi(o_5)$	4	22	24	7	21+19+ 6+ 0 = 46
$\Phi(o_6)$	20	8	16	5	5 + 5+ 2+ 2 = 14
$\Phi(o_7)$	25	18	4	15	0+15+14+ 8 = 37
$\Phi(o_8)$	9	22	12	10	16+19+ 6+ 3 = 44
$\Phi(o_9)$	27	7	33	12	2 + 4+15+ 5 = 26
$\Phi(o_{10})$	14	11	28	6	11+ 8+10+ 1 = 30

	1_{st}	2_{nd}	3_{rd}	4_{th}	
Π_q	o_2	o_4	o_3	o_1	
Π_{o_5}	o_1	o_4	o_2	o_3	$3^2+2^2+1^2+0^2 = 14$
Π_{o_6}	o_4	o_2	o_3	o_1	$0^2+1^2+0^2+1^2 = 2$
Π_{o_7}	o_3	o_4	o_2	o_1	$0^2+2^2+2^2+0^2 = 8$
Π_{o_8}	o_1	o_4	o_3	o_2	$3^2+3^2+0^2+0^2 = 18$
Π_{o_9}	o_2	o_4	o_1	o_3	$1^2+0^2+1^2+0^2 = 2$
$\Pi_{o_{10}}$	o_4	o_2	o_1	o_3	$1^2+1^2+1^2+1^2 = 4$

Figure 4.1: Example of the standard pivot-based indexing scheme and the permutation-based indexing scheme ($K = 4, N = 10$). The former scheme computes a score vector for each non-pivot as an index in advance. At query time, it first computes a score for each pivot, and then computes scores for M ($\leq N - K = 6$) non-pivots in ascending order of pseudo-scores (in this case, in an order of o_6, o_9, o_{10}, $\cdots$). The latter scheme uses permutations instead of score vectors.

in ascending (or descending) order of distances (or similarities), and computes a distance between two permutations as a pseudo-score.

That is, in the step 3 of the indexing algorithm, this scheme computes, for each non-pivot $o_i \in \mathbb{O}_{non}$, a permutation Π_{o_i} using $\mathbb{S}_{pnp}$ (i.e. $\mathbb{I} = \{\Pi_{o_i} | o_i \in \mathbb{O}_{non}\}$). In the step 2 of the search algorithm, it first computes a permutation Π_q for the query object $q \in \mathbb{X}$ using $\mathbb{S}_{piv}$, and then computes, for each non-pivot $o_i \in \mathbb{O}_{non}$, a distance between Π_q and Π_{o_i} as a pseudo-score.

As a distance measure between two permutations, the rank correlation such as Spearman's Footrule, Spearman's Rho, and Kendall Tau can be used [22]. For example, if Spearman's Rho is used, a pseudo-score can be expressed as follows [22]:

$$\tilde{s}(q, o_i) = S_\rho(\Pi_q, \Pi_{o_i}) \quad (4.3)$$

$$= \sum_{j=1}^{K} (\Pi_q^{-1}(j) - \Pi_{o_i}^{-1}(j))^2, \quad (4.4)$$

where $\Pi^{-1}(j)$ denotes the position of o_j in Π. For example, if $\Pi_q = o_2, o_4, o_3, o_1$ and $\Pi_{o_i} = o_1, o_4, o_2, o_3$, then $S_\rho(\Pi_q, \Pi_{o_i}) = (4-1)^2 + (1-3)^2 + (3-4)^2 + (2-2)^2 = 14$.

We also show in Figure 4.1 the example of the permutation-based indexing scheme in the case where Spearman's Rho is used as a distance measure between two permutations. In both the standard pivot-based scheme and the permutation-based scheme, the index size can be expressed as $O(N)$ if the number of pivots K is fixed. This is an advantage over the indexing schemes such as AESA (Approximating and Eliminating Search Algorithm) [119], its variant proposed by Maeda *et al.* [65], and iAESA (improved AESA) [35] which require the index size of $O(N^2)$.

It was shown that the permutation-based indexing scheme outperformed other schemes in some cases by a wide margin [22, 4], and since then a number of researches have been made on searching in permutation spaces: Figueroa *et al.* [35] proposed iAESA (improved AESA) which uses Spearman's Footrule instead of L_1 to choose pivots in AESA [119], and reported it outperformed AESA with regard to response time; The extra CPU time in the permutation-based scheme due to computing all pseudo-scores and sorting them was reduced by using another indexing scheme [36]; PP-Index (Permutation Prefix Index) [33] and M-Index (Metric Index) [79] were proposed as permutation-based tree structures; The number of existing permutations out of the $K!$ unrestricted permutations was examined in [102].

4.2.4 Other Pseudo-score Based Indexing Schemes

The distance regression-based indexing scheme, proposed by Edsberg and Hetland [32], is another example of pseudo-score based schemes. This scheme uses distances as scores. At indexing time, it computes, for each non-pivot $o_i \in \mathbb{O}_{non}$, $K+1$ regression coefficients in linear regression ($N(K+1)$ coefficients in total), using $\mathbb{S}_{pnp}$ and other objects used for training. At query time, it first computes $\Phi(q)$, and then computes, for each non-pivot $o_i \in \mathbb{O}_{non}$, either the estimate of $s(q, o_i)$ (i.e. distance-based pseudo-score) or $P(s(q, o_i) \leq r)$ (i.e. probability-based pseudo-score) as a pseudo-score, where r represents a threshold in range queries. $s(q, o_i)$ is estimated from $\Phi(q)$ by assuming the linear regression model, and $P(s(q, o_i) \leq r)$ is estimated by further assuming the estimation errors are normally distributed. This scheme outperformed the permutation-based scheme in some cases, and did not in other cases in their experiment.

BoostMAP [6] is a pseudo-score based scheme using the AdaBoost method. This scheme first constructs a number of one-dimensional (1D) embeddings which are divided into two types: one embed an object as the distance to a single pivot, and the other embeds an object as the projection onto the line between two pivots. Each embedding is treated as a weak classifier which predicts for any three objects $o_1, o_2, o_3 \in \mathbb{X}$ whether o_1 is closer to o_2 or to o_3. AdaBoost is then applied to combine the weak classifiers into a strong classifier. Finally, a pseudo-score is computed as a weighted L_1 distance between two multi-dimensional embeddings, where the weights are determined in the training phase.

4.3 Posterior Probability-based Search Scheme

We propose the PPS (Posterior Probability-based Search) scheme which can be applied to any pseudo-score based scheme reviewed in Section 4.2. We first explain the search algorithm of our scheme, clarifying its optimality in Section 4.3.1. Then, we explain the learning algorithm in Section 4.3.2, and propose a technique which speeds up learning in Section 4.3.3. We note that *we use distances as scores until the end of Section 4.3.* The same argument applies to the case where similarities are used as scores. We also assume that both scores and pseudo-scores are continuous (the discussion below can be easily extended to the discrete case).

4.3.1 Search Algorithm

Pseudo-score based scheme obtains a set of pseudo-scores $\tilde{\mathbb{S}}_{non} = \{\tilde{s}(q, o_i) | o_i \in \mathbb{O}_{non}\}$ after computing scores for pivots. Let us fix objects in a database, pivots, and a pseudo-score based scheme to be used ($\tilde{\mathbb{S}}_{non}$ is determined only depending on $q \in \mathbb{X}$). Then, it would be natural to consider that if it searches non-pivots in descending order of the posterior probability of being in the answer to the range query $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$, it can minimize the expected number of retrieval errors. We formalize this intuition as the following proposition:

Proposition 4.1. *Consider a search scheme which searches non-pivots in descending order of $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$. If the posterior probability $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$ can be perfectly estimated* *for any $\tilde{\mathbb{S}}_{non}$, then the above search scheme minimizes, for any M ($K \leq M \leq N$), the expected number of retrieval errors among all search schemes which search non-pivots using $\tilde{\mathbb{S}}_{non}$ as a clue.*

Proof. Let $n_i \in \{1, 2, \cdots, N\}$ ($1 \leq i \leq N - M$) be an object ID which is not searched after computing scores for K pivots and $M - K$ non-pivots (n_i is determined depending on a search scheme and $\tilde{\mathbb{S}}_{non}$). Then, the expected number of retrieval errors can be written as follows:

$$\int \sum_{i=1}^{N-M} P(s(q, o_{n_i}) \leq r | \tilde{\mathbb{S}}_{non}) p(\tilde{\mathbb{S}}_{non}) d\tilde{\mathbb{S}}_{non}, \tag{4.5}$$

where $P()$ and $p()$ are a probability mass function and probability density function, respectively. This value is minimized, for any M ($K \leq M \leq N$), in the case where we search non-pivots in descending order of $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$, if $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$ can be perfectly estimated for any $\tilde{\mathbb{S}}_{non}$. □

This proposition states that the optimal trade-off between the number of score computations and retrieval errors can be achieved by searching non-pivots in descending order of $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$. We propose the PPS (Posterior Probability-based Search) scheme which is based on this idea. The PPS scheme normalizes pseudo-scores to the posterior probabilities of being in the answer to the range query, and searches non-pivots in descending order of them. However, this scheme does not compute $P(s(q, o_i) \leq r | \tilde{\mathbb{S}}_{non})$ but approximates it by $P(s(q, o_i) \leq r | \tilde{s}(q, o_i))$. There are two reasons for this: (1) the pseudo-score $\tilde{s}(q, o_i)$ is the best clue to the probability $P(s(q, o_i) \leq r)$; (2) $P(s(q, o_i) \leq r | \tilde{s}(q, o_i))$ can be easily computed using logistic regression [12], which is to be hereinafter described.

The posterior probability $P(s(q, o_i) \leq r | \tilde{s}(q, o_i))$ can be written using Bayes' theorem as follows:

$$\begin{aligned} & P(s(q, o_i) \leq r | \tilde{s}(q, o_i)) && (4.6) \\ = \; & \frac{p(\tilde{s}(q, o_i) | s(q, o_i) \leq r) P(s(q, o_i) \leq r)}{p(\tilde{s}(q, o_i))} && (4.7) \\ = \; & \sigma(a_i), && (4.8) \end{aligned}$$

where σ is a logistic sigmoid function which is defined as

$$\sigma(a_i) = \frac{1}{1 + \exp(-a_i)} \tag{4.9}$$

and a_i is given by

$$a_i = \ln \frac{p(\tilde{s}(q, o_i) | s(q, o_i) \leq r) P(s(q, o_i) \leq r)}{p(\tilde{s}(q, o_i) | s(q, o_i) > r) P(s(q, o_i) > r)}. \tag{4.10}$$

Since σ is a monotonically increasing function, we only have to compute a_i and sort the objects in descending order of a_i instead of $P(s(q, o_i) \leq r | \tilde{s}(q, o_i))$. In other words, we can use a_i as a probability-based pseudo-score.

Here we assume the logistic regression model [12] to quickly compute a_i. We use a parameter vector $\mathbf{w_i} = (w_{i1}, w_{i0})^T \in \mathbb{R}^2$ for each non-pivot $o_i \in \mathbb{O}_{non}$ which is determined in

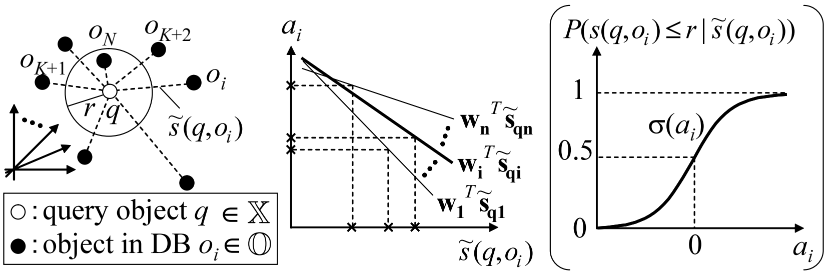

o_N
o_{K+2}
o_{K+1}
o_i
r
q
$\tilde{s}(q,o_i)$
○ : query object $q \in \mathbb{X}$
● : object in DB $o_i \in \mathbb{O}$
a_i
$\mathbf{w_n}^T\tilde{\mathbf{s}}_{\mathbf{qn}}$
$\mathbf{w_i}^T\tilde{\mathbf{s}}_{\mathbf{qi}}$
$\mathbf{w_1}^T\tilde{\mathbf{s}}_{\mathbf{q1}}$
$\tilde{s}(q,o_i)$
$P(s(q,o_i) \leq r \mid \tilde{s}(q,o_i))$
1
0.5
0
$\sigma(a_i)$
0
a_i

Figure 4.2: Overview of the computation of the probability-based pseudo-score in our scheme. After $\tilde{s}(q, o_i)$ is computed using a pseudo-score based scheme, our scheme computes the probability-based pseudo-score a_i as $a_i = \mathbf{w_i}^T\tilde{\mathbf{s}}_{\mathbf{qi}}$, where $\mathbf{w_i} = (w_{i1}, w_{i0})^T$ and $\tilde{\mathbf{s}}_{\mathbf{qi}} = (\tilde{s}(q, o_i), 1)^T$. Since the ordering is not changed by the monotonically increasing function σ, our scheme does not compute $P(s(q, o_i) \leq r|\tilde{s}(q, o_i))$ itself.

advance (see Section 4.3.2 for details). Then, a_i can be computed using logistic regression as follows:

$$a_i = \mathbf{w_i}^T\tilde{\mathbf{s}}_{\mathbf{qi}}, \tag{4.11}$$

where $\tilde{\mathbf{s}}_{\mathbf{qi}} = (\tilde{s}(q, o_i), 1)^T$. If we use the parameter vector $\mathbf{w} = (w_1, w_0)^T$ which is common to all objects instead of $\mathbf{w_i}$, a_i is equivalent to the original pseudo-score $\tilde{s}(q, o_i)$ because the ordering is not changed by a single linear transformation. By using the object-specific parameter vector $\mathbf{w_i}$ which is more strict, our scheme can more correctly estimate a_i and provide a better trade-off between the number of score computations and retrieval errors.

Figure 4.2 shows the overview of the computation of a_i in our scheme. The computation of a_i requires just one multiplication and one addition. Our scheme requires an additional storage of $2(N - K)$ parameters.

Combining the above argument with the search algorithm in Section 4.2.1, the proposed search algorithm can be written as follows:

[Search Algorithm]

1. Compute a set of scores for pivots $\mathbb{S}_{piv} = \{s(q, o_i)|o_i \in \mathbb{O}_{piv}\}$ (K scores in total).
2. Compute a set of pseudo-scores for non-pivots $\tilde{\mathbb{S}}_{non} = \{\tilde{s}(q, o_i)|o_i \in \mathbb{O}_{non}\}$ ($N - K$ pseudo-scores in total).
3. Compute a set of probability-based pseudo-scores $\{a_i = w_{i1}\tilde{s}(q, o_i) + w_{i0}|o_i \in \mathbb{O}_{non}\}$.
4. Compute scores for $M - K$ ($K \leq M \leq N$) non-pivots in descending order of probability-based pseudo-scores a_i.
5. Output the answer of range queries.

The step 1 and 2 can be carried out using any pseudo-score based scheme. The step 3 and 4 are carried out using the PPS scheme.

4.3.2 Learning Algorithm

We now explain the algorithm for determining, for each object $o_i \in \mathbb{O}_{non}$, a parameter vector $\mathbf{w_i}$. We first prepare N' objects $u_1, \cdots, u_{N'} \in \mathbb{X}$ to determine $\mathbf{w_i}$. In this dissertation, we refer to these objects as *training objects*. They can be any objects other than $o_i \in \mathbb{O}_{non}$ and

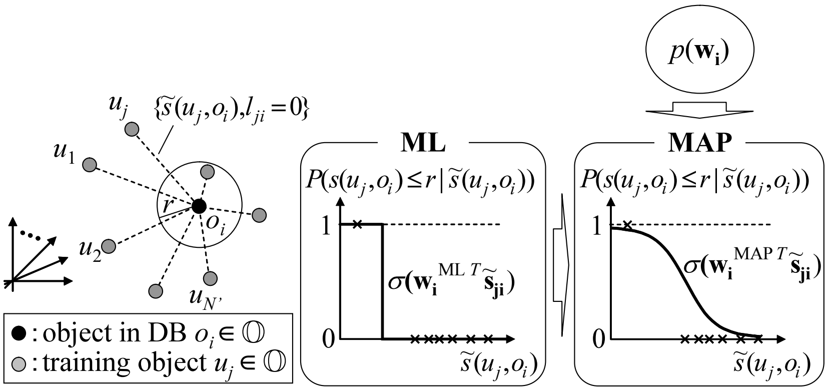

$p(\mathbf{w_i})$
u_j $\{\tilde{s}(u_j,o_i), l_{ji}=0\}$
u_1
r
o_i
u_2
$u_{N'}$
● : object in DB $o_i \in \mathbb{O}$
○ : training object $u_j \in \mathbb{O}$
ML
$P(s(u_j,o_i) \leq r | \tilde{s}(u_j,o_i))$
1
0
$\sigma(\mathbf{w_i}^{\mathrm{ML}\,T}\mathbf{\tilde{s}_{ji}})$
$\tilde{s}(u_j,o_i)$
MAP
$P(s(u_j,o_i) \leq r | \tilde{s}(u_j,o_i))$
1
0
$\sigma(\mathbf{w_i}^{\mathrm{MAP}\,T}\mathbf{\tilde{s}_{ji}})$
$\tilde{s}(u_j,o_i)$

Figure 4.3: Overview of the determination of the object-specific regression parameter $\mathbf{w_i}$ using MAP estimation. $P(s(u_j, o_i) \leq r|\tilde{s}(u_j, o_i))$ can be modeled as $\sigma(\mathbf{w_i}^T\mathbf{\tilde{s}_{ji}})$, and $l_{ji} \in \{1, 0\}$ serves as a correct answer as to whether $s(u_j, o_i) \leq r$ or not. Since r is often set to be very small, it often happens that there are a small number of similar training objects and ML estimation suffers from the over-fitting problem. MAP estimation avoids this problem by introducing the prior distribution $p(\mathbf{w_i})$.

pivots. For example, if we use all non-pivots except $o_i \in \mathbb{O}_{non}$ as training objects, the number of training objects is $N' = N - K - 1$.

Then we compute, for each training object u_j $(1 \leq j \leq N')$, a score $s(u_j, o_i)$ and a pseudo-score $\tilde{s}(u_j, o_i)$. Let $\tilde{\mathbb{S}}_i = \{\tilde{s}(u_j, o_i)|1 \leq j \leq N'\}$ be a set of pseudo-scores and $\mathbb{L}_i = \{l_{ji}|1 \leq j \leq N'\}$ a set of labels, where l_{ji} takes the following values:

$$l_{ji} = \begin{cases} 1 & (\text{if } s(u_j, o_i) \leq r) \\ 0 & (\text{if } s(u_j, o_i) > r). \end{cases} \tag{4.12}$$

We use $\{\tilde{\mathbb{S}}_i, \mathbb{L}_i\}$ as a training dataset and determine $\mathbf{w_i}$ using MAP (Maximum a Posteriori) estimation [12]. This method estimates $\mathbf{w_i^{MAP}}$ that maximizes the posterior probability $p(\mathbf{w_i}|\tilde{\mathbb{S}}_i, \mathbb{L}_i)$, which is written using Bayes' theorem as follows:

$$\mathbf{w_i^{MAP}} = \arg\max_{\mathbf{w_i}} p(\mathbf{w_i}|\tilde{\mathbb{S}}_i, \mathbb{L}_i) \tag{4.13}$$

$$= \arg\max_{\mathbf{w_i}} p(\tilde{\mathbb{S}}_i, \mathbb{L}_i|\mathbf{w_i})p(\mathbf{w_i}). \tag{4.14}$$

Here, we refer to the training object u_j whose distance to o_i is less than or equal to r (i.e. $l_{ji} = 1$) as a *similar training object*. Since the search radius r is often set to be very small so that less than tens or hundreds of objects are found in a correct answer, it often happens that there are a small number of similar training objects, and consequently a training dataset is linearly separable. ML (Maximum Likelihood) estimation, which selects the parameter vector $\mathbf{w_i^{ML}}$ that maximizes $p(\tilde{\mathbb{S}}_i, \mathbb{L}_i|\mathbf{w_i})$, suffers from the over-fitting problem for such a training dataset. MAP estimation avoids such a problem by introducing a prior distribution of the parameter vector $p(\mathbf{w_i})$ in (4.14).

Figure 4.3 shows the overview of the determination of $\mathbf{w_i}$ using MAP estimation. Here, $P(s(u_j, o_i) \leq r|\tilde{s}(u_j, o_i))$ is modeled as $\sigma(\mathbf{w_i}^T\mathbf{\tilde{s}_{ji}})$, where $\mathbf{\tilde{s}_{ji}} = (\tilde{s}(u_j, o_i), 1)^T$ (see (4.8) and (4.11)), and $l_{ji} \in \{1, 0\}$ serves as a correct answer as to whether $s(u_j, o_i) \leq r$ or not.

Since $\tilde{\mathbb{S}}_i$ and $\mathbf{w_i}$ can be regarded as independent unless $\mathbb{L}_i$ is obtained, (4.14) is further written as follows:

$$\begin{aligned}\mathbf{w_i^{MAP}} &= \arg\max_{\mathbf{w_i}} P(\mathbb{L}_i|\tilde{\mathbb{S}}_i, \mathbf{w_i})p(\tilde{\mathbb{S}}_i|\mathbf{w_i})p(\mathbf{w_i}) && (4.15)\\ &= \arg\max_{\mathbf{w_i}} P(\mathbb{L}_i|\tilde{\mathbb{S}}_i, \mathbf{w_i})p(\tilde{\mathbb{S}}_i)p(\mathbf{w_i}) && (4.16)\\ &= \arg\max_{\mathbf{w_i}} P(\mathbb{L}_i|\tilde{\mathbb{S}}_i, \mathbf{w_i})p(\mathbf{w_i}). && (4.17)\end{aligned}$$

In other words, $\mathbf{w_i^{MAP}}$ is a parameter vector which minimizes the following function:

$$E(\mathbf{w_i}) = -\ln P(\mathbb{L}_i|\tilde{\mathbb{S}}_i, \mathbf{w_i})p(\mathbf{w_i}). \quad (4.18)$$

Since $\{\tilde{s}(u_j, o_i), l_{ji}\}$ $(1 \leq j \leq N')$ can be regarded as mutually independent, and $l_{ji} \in \mathbb{L}_i$ takes 1 if $s(u_j, o_i) \leq r$ and 0 otherwise, $P(\mathbb{L}_i|\tilde{\mathbb{S}}_i, \mathbf{w_i})$ is written as follows:

$$\begin{aligned}P(\mathbb{L}_i|\tilde{\mathbb{S}}_i, \mathbf{w_i}) &= \prod_{j=1}^{N'} P(l_{ji}|\tilde{s}(u_j, o_i), \mathbf{w_i}) && (4.19)\\ &= \prod_{j=1}^{N'} y_{ji}^{l_{ji}}(1-y_{ji})^{1-l_{ji}}, && (4.20)\end{aligned}$$

where y_{ji} is given by

$$\begin{aligned}y_{ji} &= P(s(u_j, o_i) \leq r|\tilde{s}(u_j, o_i), \mathbf{w_i}) && (4.21)\\ &= \sigma(\mathbf{w_i}^T\tilde{\mathbf{s}}_{\mathbf{ji}}). && (4.22)\end{aligned}$$

As for $p(\mathbf{w_i})$ in (4.18), we assume a 2D Gaussian distribution model with mean vector $\mathbf{0} = (0,0)^T$ and diagonal covariance matrix $\alpha\mathbf{I}$ $(\alpha > 0)$, for simplicity. That is,

$$p(\mathbf{w_i}) = \mathcal{N}(\mathbf{w_i}|\mathbf{0}, \alpha\mathbf{I}). \quad (4.23)$$

This setting reduces the values of the components of $\mathbf{w_i^{MAP}}$ in (4.17) to avoid the over-fitting problem in the same way as shrinkage methods [12]. The parameter α, which is common to all objects, is called a *hyper-parameter* and can be determined using the empirical Bayes method [12] (for details of determining α using the empirical Bayes method, refer to [A4]).

Since the logistic sigmoid function in (4.22) is nonlinear, there is no closed-form solution to (4.18), (4.20), (4.22) and (4.23). Instead, we use the Newton-Raphson method [12], which iteratively updates the parameter vectors $\mathbf{w_i}$ according to the following formula:

$$\mathbf{w_i}^{(\text{new})} = \mathbf{w_i}^{(\text{old})} - (\nabla\nabla E(\mathbf{w_i}^{(\text{old})}))^{-1}\nabla E(\mathbf{w_i}^{(\text{old})}), \quad (4.24)$$

where

$$\nabla E(\mathbf{w_i}^{(\text{old})}) = (\alpha\mathbf{I})^{-1}\mathbf{w_i}^{(\text{old})} + \sum_{j=1}^{N'}(y_{ji}^{(\text{old})} - l_{ji})\tilde{\mathbf{s}}_{\mathbf{ji}} \quad (4.25)$$

and

$$\nabla\nabla E(\mathbf{w_i}^{(\text{old})}) = (\alpha\mathbf{I})^{-1} + \sum_{j=1}^{N'} y_{ji}^{(\text{old})}(1 - y_{ji}^{(\text{old})})\tilde{\mathbf{s}}_{\mathbf{ji}}\tilde{\mathbf{s}}_{\mathbf{ji}}^T. \quad (4.26)$$

Algorithm 1 LearnParameters($\tilde{\mathbb{S}}_i$, $\mathbb{L}_i$, Λ, τ_{max}, θ_{th})

```
Input:  $\tilde{\mathbb{S}}_i = \{\tilde{s}(u_j, o_i) \in \mathbb{R} | 1 \leq j \leq N'\}$: set of pseudo-scores,
        $\mathbb{L}_i = \{l_{ji} \in \{0, 1\} | 1 \leq j \leq N'\}$: set of labels,
        $\Lambda = \{\lambda_j \in \mathbb{R} | 1 \leq j \leq N'\}$: set of weight parameters,
        $\tau_{max} \in \mathbb{N}$: maximum number of iterations,
        $\theta_{th} \in \mathbb{R}$: convergence threshold
Output: $\mathbf{w_i} = (w_{i1}, w_{i0})^T \in \mathbb{R}^2$: parameter vector

for $j \leftarrow 0$ to $N'$ do
  $\tilde{\mathbf{s}}_{\mathbf{ji}} \leftarrow (\tilde{s}(u_j, o_i), 1)^T$
end for
$\mathbf{w_i} \leftarrow (0, 0)^T$
for $\tau \leftarrow 0$ to $\tau_{max}$ do
  for $j \leftarrow 0$ to $N'$ do
    $y_{ji} \leftarrow \sigma(\mathbf{w_i}^T \tilde{\mathbf{s}}_{\mathbf{ji}})$
  end for
  $\nabla E(\mathbf{w_i}) \leftarrow (\alpha \mathbf{I})^{-1} \mathbf{w_i} + \sum_{j=1}^{N'} \lambda_j (y_{ji} - l_{ji}) \tilde{\mathbf{s}}_{\mathbf{ji}}$
  $\nabla\nabla E(\mathbf{w_i}) \leftarrow (\alpha \mathbf{I})^{-1} + \sum_{j=1}^{N'} \lambda_j y_{ji} (1 - y_{ji}) \tilde{\mathbf{s}}_{\mathbf{ji}} \tilde{\mathbf{s}}_{\mathbf{ji}}^T$
  $(\delta_1, \delta_0)^T \leftarrow (\nabla\nabla E(\mathbf{w_i}))^{-1} \nabla E(\mathbf{w_i})$
  $\mathbf{w_i} \leftarrow \mathbf{w_i} - (\delta_1, \delta_0)^T$
  if ($|\delta_1 / w_{i1}| < \theta_{th}$ and $|\delta_0 / w_{i0}| < \theta_{th}$) then
    break
  end if
end for
Report $\mathbf{w_i}$
```

Algorithm 1 shows the learning algorithm, where $\lambda_j \in \mathbb{R}$ is a weight parameter which represents the number of u_j (i.e. we allow more than one training object which are exactly the same). Although it is natural to set $\lambda_j = 1$ $(1 \leq j \leq N')$, the different values are used in Section 4.3.3. The parameter vector $\mathbf{w_i}$ is updated until the number of iterations reaches the maximum number τ_{max} or $\mathbf{w_i}$ converges (i.e. $|\delta_1 / w_{i1}| < \theta_{th}$ and $|\delta_0 / w_{i0}| < \theta_{th}$, where θ_{th} is a convergence threshold). In our experiment in Section 4.4, we set $\tau_{max} = 100$ and $\theta_{th} = 0.0001$, and confirmed that $\mathbf{w_i}$ converged until τ reached τ_{max} in all cases.

4.3.3 Speeding up Learning Using Pseudo-scores

The drawback of our scheme is a high computational cost in determining the parameter vectors. For each non-pivot $o_i \in \mathbb{O}_{non}$, we have to compute N' pseudo-scores and N' scores to obtain $\tilde{\mathbb{S}}_i$ and $\mathbb{L}_i$, respectively. Furthermore, the iterative update process in Algorithm 1 takes $O(N')$ time for each non-pivot. That is, we have to compute $O(NN')$ pseudo-scores, $O(NN')$ scores, and carry out the iterative update process which takes $O(NN')$ time in total (e.g. if $N' = N - K - 1$, each of them takes $O(N^2)$ time). Especially, $O(NN')$ score computations and $O(NN')$ iterative update process may take too much time if the number of training objects N' is very large. If N' is small, however, shortage of similar training objects becomes a serious problem. The search radius r is often set to be very small so that less than tens or hundreds of objects are captured from N objects. Thus, if N' is much smaller than N, it can happen that there is no similar training objects at all, which makes the parameter vectors $\mathbf{w_i}$ quite difficult to be correctly determined even if we use MAP estimation.

To solve this problem, we propose a technique which reduces the number of training objects while keeping similar training objects as much as possible using pseudo-scores. For each non-pivot $o_i \in \mathbb{O}_{non}$, we select $N''(< N')$ training objects and make a set $\{\tilde{\mathbb{S}}_i, \mathbb{L}_i, \Lambda\}$ used in Algorithm 1 according to the following steps:

1. Compute N' pseudo-scores $\tilde{s}(u_j, o_i)$ $(1 \le j \le N')$.
2. Find the N''_1 smallest (or largest) pseudo-scores and select the corresponding training objects. We assume that they are $u_1, \cdots, u_{N''_1}$ without loss of generality.
3. Select N''_2 training objects at random from the remaining $N' - N''_1$ training objects. We assume that they are $u_{N''_1+1}, \cdots, u_{N''}$ without loss of generality $(N'' = N''_1 + N''_2 < N')$.
4. Compute N'' scores $s(u_j, o_i)$ $(1 \le j \le N'')$ and obtain the corresponding labels l_{ji} $(1 \le j \le N'')$.
5. Make the set $\{\tilde{\mathbb{S}}_i, \mathbb{L}_i, \Lambda\}$ used in Algorithm 1 as follows:

$$\begin{aligned} \tilde{\mathbb{S}}_i &= \{\tilde{s}(u_j, o_i) | 1 \le j \le N''\} && (4.27) \\ \mathbb{L}_i &= \{l_{ji} \in \{0, 1\} | 1 \le j \le N''\} && (4.28) \\ \Lambda &= \{\lambda_j | 1 \le j \le N''\}, && (4.29) \end{aligned}$$

where

$$\lambda_j = \begin{cases} 1 & (\text{if } 1 \le j \le N''_1) \\ (N' - N''_1)/N''_2 & (\text{if } N'_1 + 1 \le j \le N''). \end{cases} \tag{4.30}$$

In step 2, we can select the training objects $u_1, \cdots, u_{N''_1}$ which are highly likely similar to the non-pivot o_i because pseudo-scores are designed to be highly relevant to scores. However, these training objects may make the parameter vectors $\mathbf{w_i}$ incorrectly determined because the probability that a query object q falls close to o_i (i.e. $P(s(q, o_i) \le r)$ in (4.10)) is biased towards much higher than the actual value. To eliminate this bias, we select N''_2 training objects randomly from the remaining $N' - N''_1$ training objects in step 3, and set λ_j as in (4.30) in step 5.

By using this technique, the score computations and the iterative update process are reduced from $O(NN')$ time to $O(NN'')$ time $(N'' < N')$. However, we still have to compute NN' pseudo-scores, which takes $O(NN')$ time. In addition, this technique requires the time to find the N''_1 smallest (or largest) pseudo-scores from N' pseudo-scores in step 2 for every non-pivot, which takes on average $O(NN')$ time in total using the quickselect algorithm [42]. In Section 4.4, we measured the learning time using various kinds of datasets in order to evaluate our speed-up technique.

4.4 Experimental Evaluation

4.4.1 Experimental Set-up

Datasets

We evaluated our scheme using two kinds of synthetic datasets and two kinds of real-life datasets obtained from the Metric Space Library [37]. As synthetic datasets, we used (I) vectors generated from a uniform distribution and (II) vectors generated from a mixture of Gaussian distributions. As real-life datasets, we used (III) documents (short news articles) and (IV) genes (DNA sequences of Listeria monocytogenes). We used them as real-life datasets because the score computations were very expensive and the metric space indexing methods played a big role in reducing the CPU time at query time. For the same reason, we set the dimensionality of the synthetic vectors to be as much as 1024 dimensions. Although this experiment was reported in [A4], we also evaluated our scheme using images and dictionaries in [B3], and showed that our scheme outperformed the standard pivot-based scheme [22] and the permutation-based scheme [4, 22] with regard to the number of score computations in both the datasets.

In the following, we describe the details of each dataset:

Uniform Distribution A set of 10000 vectors uniformly distributed in the unit cube of 1024-dimensional Euclidean space ($N = 10000$). In this dataset, we also carried out the experiment in the case where we increased the number of objects from 10000 to 100000 ($N = 100000$) as a scalability analysis. We used another 1000 vectors as range query objects.

Mixture of Gaussians A set of 10000 vectors generated from a mixture of 1024-dimensional Gaussian distributions which has 32 clusters (components) ($N = 10000$). Objects are uniformly assigned to the clusters, whose centers are uniformly distributed in $[0, 1)$. The variance inside the clusters was 0.01. We used another 1000 vectors as range query objects.

Documents A set of 24276 short news articles extracted from Wall Street Journal from TREC-3 dataset [40] ($N = 24276$). Each document is represented as a vector whose element is proportional to the weight of a vocabulary word in that document. Since the number of vocabulary words is very large (hundreds of thousands), the vectors are in a very high-dimensional space. We used the angle between two vectors as a distance measure (the cosine of this angle is heavily used as a distance measure in the vector space model). We used another 1000 articles as range query objects.

Genes A set of 2000 DNA sequences of Listeria monocytogenes randomly extracted from the Metric Space Library [37] ($N = 2000$). The length of the sequences is as much as 898 characters on average. We used the edit distance as a distance measure between two genes. We used another 100 sequences as range query objects.

Methods

We applied our scheme to the standard pivot-based scheme [22] described in Section 4.2.2 and the permutation-based scheme [4, 22] described in Section 4.2.3. We selected these schemes as pseudo-score based schemes not only because they are state-of-the-art schemes, but because they were easy to implement and did not need much time to build the indexes. We evaluated the performance of the following schemes for comparison:

1. $\mathbf{PI}_{\Phi}$: the standard pivot-based scheme [22] described in Section 4.2.2 (PI denotes a pseudo-score based indexing scheme). We used L_1 as a distance measure between two distance vectors. The memory size required in this scheme was $4K$ [bytes/non-pivot], where K is the number of pivots.

Table 4.2: Parameter settings in our experiment (Uni.: Uniform distribution; Gau.: Mixture of Gaussians; Doc.: Documents; Gen.: Genes). $\overline{x}$ denotes the average number of objects in the correct answer.

Dataset	K	N	N'	N''	$\overline{x}$
Uni.(a)	16	10000	9983	2000	10
Uni.(b)	64	10000	9935	2000	10
Uni.(c)	16	100000	99983	2000	100
Uni.(d)	16	100000	99983	2000	10
Gau.(a)	16	10000	9983	2000	10
Gau.(b)	64	10000	9935	2000	10
Doc.	16	24276	24259	2000	9
Gen.	16	2000	1983	200	2

2. **PI$_\Phi$-PPS**: our scheme applied to the standard pivot-based scheme. We selected N'' ($N''_1 = N''_2 = N''/2$) training objects from all non-pivots except $o_i \in \mathbb{O}_{non}$ ($N' = N - K - 1$) to determine $\mathbf{w_i}$, using our speed-up technique described in Section 4.3.3. We set the number of pivots K to be equal to that in **PI$_\Phi$**. The memory size required was $4K + 8$ [bytes/non-pivot] (additional memory size was 8 [bytes/non-pivot] (= 2 parameters × 4 bytes)).

3. **PI$_\Phi$-PPS***: identical to **PI$_\Phi$-PPS** except that it uses all non-pivots except $o_i \in \mathbb{O}_{non}$ as training objects ($N' = N - K - 1$) to determine $\mathbf{w_i}$.

4. **PI$_\Pi$**: the permutation-based scheme [4, 22] described in Section 4.2.3. We used Spearman's Rho as a distance measure between two permutations. The memory size required was $K\lceil\log_2(K)\rceil/8$ [bytes/non-pivot].

5. **PI$_\Pi$-PPS**: our scheme applied to the permutation-based scheme. We selected N'' ($N''_1 = N''_2 = N''/2$) training objects from all non-pivots except $o_i \in \mathbb{O}_{non}$ ($N' = N - K - 1$) to determine $\mathbf{w_i}$. We set K to be equal to that in **PI$_\Pi$**. The memory size required was $K\lceil\log_2(K)\rceil/8 + 8$ [bytes/non-pivot] (additional memory size was 8 [bytes/non-pivot]).

6. **PI$_\Pi$-PPS***: identical to **PI$_\Pi$-PPS** except that it uses all non-pivots except $o_i \in \mathbb{O}_{non}$ as training objects ($N' = N - K - 1$) to determine $\mathbf{w_i}$.

Table 4.2 shows the parameter settings in our experiment, where $\overline{x}$ denotes the average number of objects in the correct answer over the query objects (we controlled the search radius so that $\overline{x}$ would be the value in Table 4.2). Although Chávez *et al.* [22] set $K = 32$, 64, 128, or 256 and evaluated the performance of the standard pivot-based scheme and the permutation-based scheme, we set K to be very small ($K = 16$) to make the extra CPU time required to compute pseudo-scores very small. In the synthetic datasets, we also evaluated the performance in the case where $K = 64$, for comparison (Uni.(b) and Gau.(b)).

As a way of selecting pivots, we adopted the random selection method which randomly selects pivots from the objects in the database. The reason for this is that Chávez *et al.* [22] reported that the random selection method provided the performance comparable or better than other selection methods in the permutation-based scheme. Note that we used the same pivots for all the 6 schemes described above, in spite of the fact that our scheme required additional memory size of 8 [bytes/non-pivot]. The reason for this is because we would like to investigate how much the number of score computations and the CPU time were reduced at query time *just by applying our scheme*. Additional memory size of 8 [bytes/non-pivot] can also be regarded as very small in most cases in practice.

As shown in Table 4.2, Uni.(c) and Uni.(d) are datasets for scalability analysis, assuming that the number of objects in Uni.(a) was increased from 10000 to 100000. Here we considered two scenarios: (1) Uni.(c) is corresponding to the case where we did not change the search radius from that of Uni.(a) (on average 0.1[%] ($= \overline{x}/N \times 100$[%]) of the objects was included in the correct answer); (2) Uni.(d) is corresponding to the case where we changed the search radius so that the average number of objects in the correct answer $\overline{x}$ would not be changed from that of Uni.(a) ($\overline{x} = 10$).

After building the indexes and learning the parameter vector $\mathbf{w_i}$ for each object $o_i \in \mathbb{O}_{non}$ using MAP estimation, we tested the range query objects.

Performances

We first evaluated the trade-off between the percentage of score computations required and the percentage of the correct answer retrieved (i.e. 100 - retrieval error rate [%]). Here we did not include pivots in both of them because our interest was to compare the performance of the different pseudo-score functions themselves. We then evaluated the total CPU time (including computing scores for pivots, computing pseudo-scores, sorting pseudo-scores, and computing scores for non-pivots) at query time, on an Intel Core2 Duo CPU E7500 (2.93 GHz) with 1.93GB RAM. Here we also evaluated the CPU time required in the sequential scan which merely computes scores for all the objects in the database, for comparison. We further evaluated the time to build the indexes and to learn the parameter vectors.

4.4.2 Experimental Results

Figure 4.4 shows the trade-off between the percentage of score computations and the percentage of the correct answer retrieved. In Table 4.3, we also give the percentage of score computations required to retrieve 90% of the correct answer in each dataset (i.e. retrieval error rate = 10%), where $\overline{y}$ denotes the average number of similar training objects per non-pivot.

Table 4.4 shows the results of measuring the CPU time at query time, where t_{cpu}, t_{ext}, t_{scr} denote the CPU time required to retrieve 90% of the correct answer, the extra CPU time, and the time to compute a score, respectively (all of the three are the average values). Since t_{ext} in the dataset of genes was less than 1 [ms] in every scheme and was much smaller than t_{cpu}, we did not measure the value of t_{ext} itself in the dataset. From this table, we can simply calculate the CPU time to retrieve β% ($0 \leq \beta \leq 100$) of the objects in the database as $t_{ext} + \beta N t_{scr}/100$ [ms].

As for the building time and learning time, Table 4.5 shows the time to build the index and learn the parameter vectors in **PI$_\Pi$-PPS** and **PI$_\Pi$-PPS*** (since **PI$_\Phi$-PPS** and **PI$_\Phi$-PPS*** had similar results, we do not show them here).

Summary of results

Although a number of findings can be obtained from Figure 4.4, Table 4.3, 4.4 and 4.5, we summarize the main findings as follows:

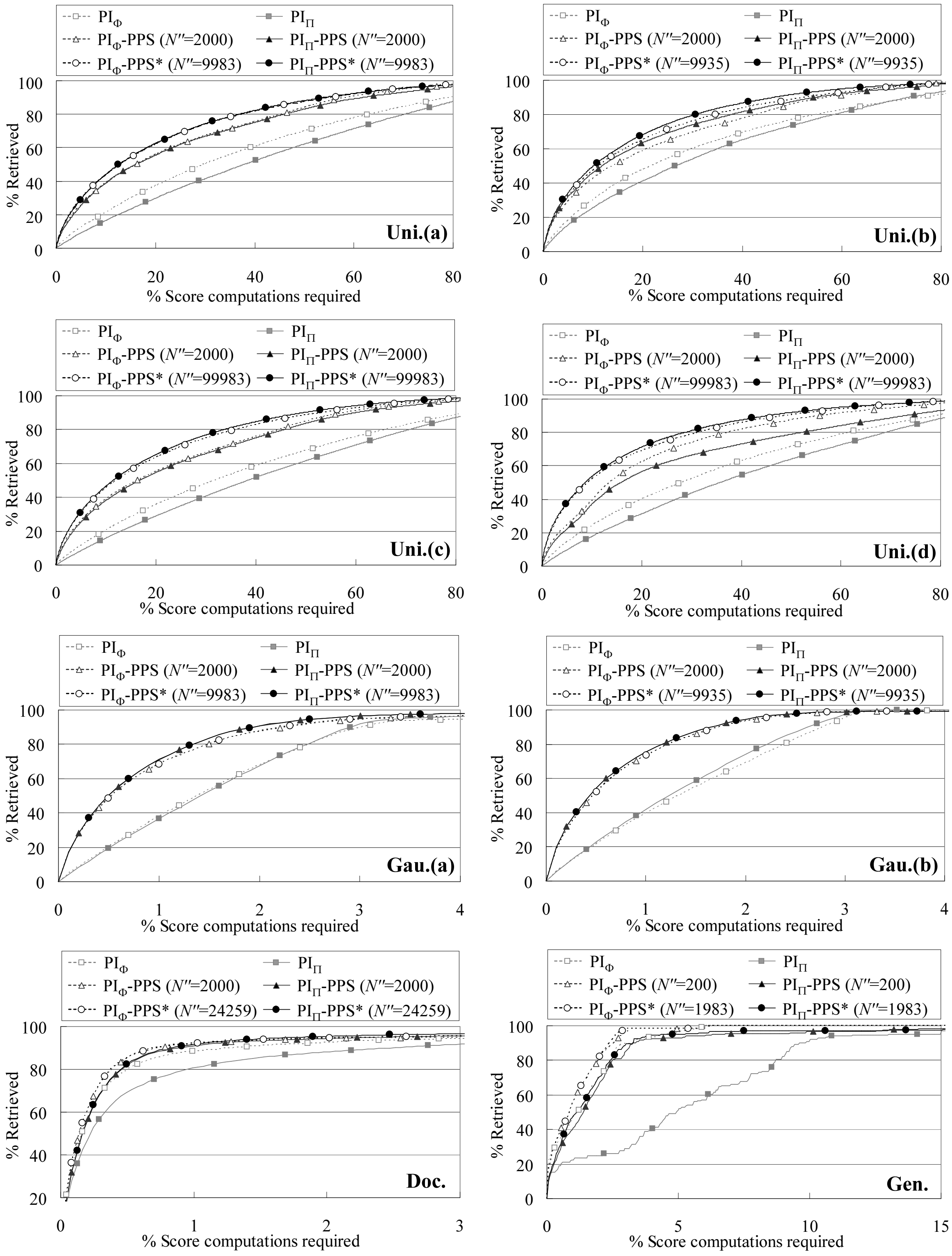


Figure 4.4: Trade-off between the percentage of score computations and the percentage of the correct answer retrieved (Uni.(a): $K = 16$, $N = 10000$, $\overline{x} = 10$; Uni.(b): $K = 64$, $N = 10000$, $\overline{x} = 10$; Uni.(c): $K = 16$, $N = 100000$, $\overline{x} = 100$; Uni.(d): $K = 16$, $N = 100000$, $\overline{x} = 10$; Gau.(a): $K = 16$, $N = 10000$, $\overline{x} = 10$; Gau.(b): $K = 64$, $N = 10000$, $\overline{x} = 10$; Doc.: $K = 16$, $N = 24276$, $\overline{x} = 9$; Gen.: $K = 16$, $N = 2000$, $\overline{x} = 2$).

Table 4.3: Percentage of score computations required to retrieve 90% of the correct answer (i.e. retrieval error rate = 10%). $\overline{y}$ denotes the average number of similar training objects per object in the database.

Dataset	**PI$_\Phi$**	**PI$_\Phi$-PPS**	**PI$_\Phi$-PPS***	**PI$_\Pi$**	**PI$_\Pi$-PPS**	**PI$_\Pi$-PPS***
Uni.(a)	79.6%	58.7% ($\overline{y} = 3.22$)	55.4% ($\overline{y} = 10.6$)	83.2%	61.7% ($\overline{y} = 2.74$)	53.8% ($\overline{y} = 10.6$)
Uni.(b)	75.8%	57.0% ($\overline{y} = 4.04$)	52.6% ($\overline{y} = 10.5$)	73.3%	54.8% ($\overline{y} = 3.66$)	46.3% ($\overline{y} = 10.5$)
Uni.(c)	82.0%	57.7% ($\overline{y} = 4.19$)	53.1% ($\overline{y} = 105$)	83.9%	59.8% ($\overline{y} = 3.18$)	49.3% ($\overline{y} = 105$)
Uni.(d)	79.2%	56.0% ($\overline{y} = 0.54$)	48.8% ($\overline{y} = 11.1$)	82.5%	72.8% ($\overline{y} = 0.37$)	45.1% ($\overline{y} = 11.1$)
Gau.(a)	3.10%	2.30% ($\overline{y} = 10.4$)	2.30% ($\overline{y} = 10.6$)	2.90%	2.00% ($\overline{y} = 10.5$)	2.00% ($\overline{y} = 10.6$)
Gau.(b)	2.82%	1.81% ($\overline{y} = 10.5$)	1.81% ($\overline{y} = 10.5$)	2.62%	1.71% ($\overline{y} = 10.5$)	1.71% ($\overline{y} = 10.5$)
Doc.	1.32%	0.74% ($\overline{y} = 8.79$)	0.74% ($\overline{y} = 9.09$)	2.51%	0.91% ($\overline{y} = 8.70$)	0.87% ($\overline{y} = 9.09$)
Gen.	3.28%	2.57% ($\overline{y} = 1.26$)	2.42% ($\overline{y} = 2.16$)	9.63%	3.28% ($\overline{y} = 2.04$)	3.28% ($\overline{y} = 2.16$)

Table 4.4: CPU time required to retrieve 90% of the correct answer (t_{cpu}), the extra CPU time (t_{ext}), and the time to compute a score (t_{scr}) [ms].

Dataset	**PI$_\Phi$**	**PI$_\Phi$-PPS**	**PI$_\Phi$-PPS***	**PI$_\Pi$**	**PI$_\Pi$-PPS**	**PI$_\Pi$-PPS***	sequential scan
Uni.(a)	$t_{cpu} = 13.7$ $(t_{ext} = 2.99)$	$t_{cpu} = 10.9$ $(t_{ext} = 3.00)$	$t_{cpu} = 10.5$ $(t_{ext} = 3.03)$	$t_{cpu} = 13.3$ $(t_{ext} = 2.14)$	$t_{cpu} = 10.5$ $(t_{ext} = 2.24)$	$t_{cpu} = 9.45$ $(t_{ext} = 2.24)$	$t_{cpu} = 13.4$ $(t_{scr} = 1.34 \times 10^{-3})$
Uni.(b)	$t_{cpu} = 16.0$ $(t_{ext} = 5.89)$	$t_{cpu} = 13.6$ $(t_{ext} = 5.93)$	$t_{cpu} = 13.0$ $(t_{ext} = 5.93)$	$t_{cpu} = 12.6$ $(t_{ext} = 2.81)$	$t_{cpu} = 10.2$ $(t_{ext} = 2.83)$	$t_{cpu} = 9.08$ $(t_{ext} = 2.85)$	$t_{cpu} = 13.4$ $(t_{scr} = 1.34 \times 10^{-3})$
Uni.(c)	$t_{cpu} = 146$ $(t_{ext} = 34.3)$	$t_{cpu} = 113$ $(t_{ext} = 34.9)$	$t_{cpu} = 107$ $(t_{ext} = 35.0)$	$t_{cpu} = 139$ $(t_{ext} = 24.9)$	$t_{cpu} = 109$ $(t_{ext} = 27.5)$	$t_{cpu} = 94.6$ $(t_{ext} = 27.5)$	$t_{cpu} = 136$ $(t_{scr} = 1.36 \times 10^{-3})$
Uni.(d)	$t_{cpu} = 144$ $(t_{ext} = 34.6)$	$t_{cpu} = 112$ $(t_{ext} = 35.0)$	$t_{cpu} = 90.1$ $(t_{ext} = 35.1)$	$t_{cpu} = 140$ $(t_{ext} = 25.6)$	$t_{cpu} = 128$ $(t_{ext} = 27.5)$	$t_{cpu} = 89.8$ $(t_{ext} = 27.5)$	$t_{cpu} = 138$ $(t_{scr} = 1.38 \times 10^{-3})$
Gau.(a)	$t_{cpu} = 3.37$ $(t_{ext} = 2.95)$	$t_{cpu} = 3.31$ $(t_{ext} = 3.00)$	$t_{cpu} = 3.32$ $(t_{ext} = 3.01)$	$t_{cpu} = 2.57$ $(t_{ext} = 2.16)$	$t_{cpu} = 2.53$ $(t_{ext} = 2.27)$	$t_{cpu} = 2.53$ $(t_{ext} = 2.27)$	$t_{cpu} = 13.4$ $(t_{scr} = 1.34 \times 10^{-3})$
Gau.(b)	$t_{cpu} = 6.14$ $(t_{ext} = 5.68)$	$t_{cpu} = 6.01$ $(t_{ext} = 5.68)$	$t_{cpu} = 6.01$ $(t_{ext} = 5.69)$	$t_{cpu} = 3.24$ $(t_{ext} = 2.80)$	$t_{cpu} = 3.17$ $(t_{ext} = 2.86)$	$t_{cpu} = 3.15$ $(t_{ext} = 2.84)$	$t_{cpu} = 13.4$ $(t_{scr} = 1.34 \times 10^{-3})$
Doc.	$t_{cpu} = 18.2$ $(t_{ext} = 7.62)$	$t_{cpu} = 13.8$ $(t_{ext} = 7.63)$	$t_{cpu} = 13.8$ $(t_{ext} = 7.62)$	$t_{cpu} = 25.4$ $(t_{ext} = 5.65)$	$t_{cpu} = 13.4$ $(t_{ext} = 6.01)$	$t_{cpu} = 13.1$ $(t_{ext} = 6.00)$	$t_{cpu} = 765$ $(t_{scr} = 3.15 \times 10^{-2})$
Gen.	$t_{cpu} = 281$ $(t_{ext} < 1.00)$	$t_{cpu} = 231$ $(t_{ext} < 1.00)$	$t_{cpu} = 220$ $(t_{ext} < 1.00)$	$t_{cpu} = 728$ $(t_{ext} < 1.00)$	$t_{cpu} = 280$ $(t_{ext} < 1.00)$	$t_{cpu} = 280$ $(t_{ext} < 1.00)$	$t_{cpu} = 7100$ $(t_{scr} = 3.55)$

Table 4.5: Building and learning time in **PI$_\Pi$-PPS** and **PI$_\Pi$-PPS*** [s]. (Bld.: time to build an index; Lrn.(1): time to determine a hyper-parameter; Lrn.(2): time to compute pseudo-scores; Lrn.(3): time to select training objects; Lrn.(4): time to compute scores; Lrn.(5): time to iteratively update parameter vectors in Algorithm 1; Total: total time to build an index and learn parameters; w/o *, w/ *: performance of **PI$_\Pi$-PPS**, **PI$_\Pi$-PPS***)

Dataset	Bld.	Lrn.(1)	Lrn.(2)	Lrn.(3)	Lrn.(4)	Lrn.(5)	Total
Uni.(a) ($N'' = 2000$)	0.2	2	3	3	27	59	94
Uni.(a)* ($N' = 9983$)	0.2	2	3	-	119	240	364
Uni.(b) ($N'' = 2000$)	0.8	2	11	3	26	47	90
Uni.(b)* ($N' = 9935$)	0.8	2	11	-	118	226	358
Uni.(c) ($N'' = 2000$)	2	23	292	358	289	641	1605
Uni.(c)* ($N' = 99983$)	2	23	264	-	11935	23370	35594
Uni.(d) ($N'' = 2000$)	2	28	296	353	287	977	1943
Uni.(d)* ($N' = 99983$)	2	28	259	-	11923	33244	45456
Gau.(a) ($N'' = 2000$)	0.2	3	3	3	26	64	99
Gau.(a)* ($N' = 9983$)	0.2	3	3	-	119	316	441
Gau.(b) ($N'' = 2000$)	0.8	3	10	3	26	65	108
Gau.(b)* ($N' = 9935$)	0.8	3	11	-	117	320	452
Doc. ($N'' = 2000$)	15	14	18	21	1484	186	1738
Doc.* ($N' = 24276$)	15	15	15	-	18100	2106	20251
Gen. ($N'' = 200$)	89	65	0.1	0.2	1690	2	1846
Gen.* ($N' = 2000$)	89	65	0.1	-	12663	18	12835

1. Our scheme outperformed the standard pivot-based scheme and the permutation-based scheme, with regard to both the number of score computations and the CPU time, in all the datasets (Figure 4.4, Table 4.3, and Table 4.4).

2. Our scheme significantly outperformed the sequential scan with regard to the CPU time, in the case of the real-life datasets (Doc., and Gen. in Table 4.4).

3. The fewer pivots ($K = 16$) provided the smaller CPU time in many cases, compared to the larger pivots ($K = 64$) (Uni.(a), Uni.(b), Gau.(a), and Gau.(b) in Table 4.4).

4. The extra CPU time in our scheme differed only slightly from that in the conventional schemes (Table 4.4).

5. The extra CPU time accounted for much of the total CPU time in some cases (e.g. Gau.(a) and Gau.(b) in Table 4.4).

6. Our speed-up technique significantly reduced the time to compute scores and to iteratively update parameter vectors in Algorithm 1 (Lrn.(4) and Lrn.(5) in Table 4.5).

7. Our speed-up technique required the time to compute pseudo-scores and to select training objects, both of which increased in proportion to NN' (Lrn.(2) and Lrn.(3) in Table 4.5).

8. Our speed-up technique did not significantly affect the performance in Gau.(a), Gau.(b), Doc., and Gen., while it adversely affected the performance in Uni.(d) (Figure 4.4, Table 4.3, and Table 4.4).

In the next subsection, we discuss these findings in detail.

4.4.3 Discussion

Score Computations

Our scheme outperformed the conventional schemes in all the datasets, with regard to the number of score computations (see Figure 4.4 and Table 4.3). Especially, in the case of the documents and genes, our scheme reduced the percentage of score computations required to retrieve 90% of the correct answer to one-third of the percentage in the permutation-based scheme. We consider this is relevant to the validity of the model assumption. That is, from a theoretical perspective, we can say that our scheme works very well iff (a) the probability-based pseudo-score a_i can be correctly modeled as a linear function of the pseudo-score $\tilde{s}(q, o_i)$ and (b) the parameter vector $\mathbf{w_i}$ is different for each object $o_i \in \mathbb{O}_{non}$.

It should also be noted that the permutation-based scheme performed almost the same as or worse than the standard pivot-based scheme with regard to the number of score computations, in the case where $K = 16$ pivots (i.e. Uni.(a), Uni.(c), Uni.(d), Gau.(a), Doc., and Gen.). On the other hand, we confirmed in [B3] that the permutation-based scheme outperformed the standard pivot-based scheme in the case where $K = 128$ pivots or 256 pivots. We consider the reason the permutation-based scheme performed worse in this experiment is that the number of pivots K was very small and the permutations did not have discriminative power compared to the score vectors.

CPU Time

Our scheme also outperformed the conventional schemes in all the datasets with regard to the CPU time (see Table 4.4). In the case of the real-life datasets (i.e. Doc. and Gen.), the CPU time in our scheme was much smaller than that in the sequential scan (less than one-thirtieth). This is because the score computations were very expensive in those datasets, as described in Section 4.4.1, and our scheme significantly reduced the number of score computations required (see Table 4.3). It was also found that the fewer pivots provided the smaller CPU time in many cases (see Uni.(a), Uni.(b), Gau.(a), and Gau.(b) in Table 4.4). This is because the fewer pivots resulted in the decrease of the extra CPU time, while they did not result in a significant increase of score computations (see Uni.(a), Uni.(b), Gau.(a), and Gau.(b) in Table 4.3). Since the number of score computations was very small in the real-life datasets (see Doc. and Gen. in Table 4.3), the significant increase of pivots would also result in the higher CPU time in those datasets.

As for the extra CPU time, our scheme performed almost the same with the conventional schemes. This is because the computation of one probability-based pseudo-score only requires one multiplication and one addition if we use the logistic regression model, as described in Section 4.3.1. However, the extra CPU time required to compute pseudo-scores and to sort them accounted for much of the total CPU time in some cases (e.g. Gau.(a) and Gau.(b) in Table 4.4). To further reduce the total CPU time in such cases, we need to reduce the extra CPU time using, for example, another indexing scheme [36].

The Speed-up Technique and the Learning Time

Our speed-up technique significantly reduced the time to compute scores and to iteratively update parameter vectors in Algorithm 1 (see Lrn.(4) and Lrn.(5) in Table 4.5). This was especially effective in the real-life datasets (i.e. Doc. and Gen.) where the score computations were very expensive, and in the very large datasets (i.e. Uni.(c) and Uni.(d)) where the iterative update process required much time. However, our speed-up technique required the time to compute pseudo-scores and to select training objects, and both of them increased in proportion to NN', as described in Section 4.3.3.

While significantly reducing the learning time, our speed-up technique did not significantly affect the performance in the vectors from a mixture of Gaussians, documents, and genes (see Gau.(a), Gau.(b), Doc., and Gen. in Figure 4.4, Table 4.3, and Table 4.4). The reason for this is that the performance of $\mathbf{PI}_{\Phi}$ and $\mathbf{PI}_{\Pi}$ were very good with regard to score computations (it can be seen from Figure 4.4 that they retrieved most of the correct answer in less than 10% of score computations), and consequently many or most of the similar training objects were selected (see Table 4.3). On the other hand, our speed-up technique adversely affected the performance when our scheme was applied to the permutation-based scheme in Uni.(d). This is because the search radius was set to be very small so that the average number of objects in the correct answer would not be changed as described in Section 4.4.1, and consequently most of the similar training objects were *not* selected (see Uni.(d) in Table 4.3).

As a conclusion, we can say that we should not make the search radius smaller as we add objects into the existing database in our scheme in range queries. There are two reasons for this: (1) if the search radius changes, we have to learn the parameter vector $\mathbf{w_i}$ for each object $o_i \in \mathbb{O}_{non}$ again to adapt to the new search radius; (2) as the search radius is set to be smaller, the number of similar training objects is reduced, and hence the number of score computations and the CPU time at query time is increased. Thus, it would be desirable to fix the search radius, in the same way as Uni.(c). Then, since the number of objects in the correct answer increases as we add objects into the existing database, it may be better to output objects in the correct answer in ascending order of distances to the query object if there are too many objects in the correct answer. That is, we consider one solution would be the combination of the range query and the k-NN query.

4.5 Conclusions

In this chapter, we focused on range queries and proposed the PPS scheme which can be applied to any pseudo-score based indexing scheme. We described the algorithm of the PPS scheme, clarifying its optimal property with regard to the trade-off between the number of score computations and retrieval errors. We applied our scheme to the two state-of-the-art schemes: the standard pivot-based scheme and the permutation-based scheme, and evaluated the performance using vectors from a uniform distribution, vectors from a mixture of Gaussian distributions, documents, and genes. The results showed that our scheme outperformed the conventional schemes with regard to both the number of score computations and the CPU time, in all the datasets. In [B3], we also evaluated our scheme using images and dictionaries, and showed that our scheme outperformed the above schemes with regard to the number of score computations. We consider they are very successful results because our scheme outperformed the conventional schemes including the permutation-based indexing scheme, which is followed by a number of studies [33, 35, 36, 79, 102], in so many datasets.

The drawback of our scheme in range queries is a high computational cost in learning the parameter vectors (our speed-up still requires the time to compute pseudo-scores and to select training objects, both of which is $O(NN'))$. However, we explain that *the problem of the learning time does not occur in biometric identification* in the next chapter.

Chapter 5

Towards Optimal Sequential Indexing and Fusion in Biometric Identification

Contents

5.1 Introduction

In Section 2.2.2, we introduced metric space indexing methods which reduce the number of score computations at query time using the index which is constructed based on scores. We now consider applying them to biometric identification to reduce response time. Although exact indexing schemes guarantee to return the correct answer, they are not useful when features are in a high dimensional space (e.g. Daugman [27] used 2048-bit iriscodes) or score measures do not satisfy the triangle-inequality. Approximate indexing schemes can reduce the number of score computations even in such cases. However, they can fail to compute a genuine score, and consequently they can incorrectly reject an enrollee as a non-enrollee. In this chapter, we refer to the errors in which the system does not compute a genuine score as *retrieval errors* (i.e. a correct answer in this chapter is a set of genuine templates)[1]. There is a trade-off between the number of score computations and retrieval errors, and the increase of retrieval errors causes the increase of false rejects.

Some existing work proposed a metric space indexing method for biometric identification, all of which are approximate indexing schemes [10, 39, 65]. For example, Maeda *et al.* [65] proposed a metric space indexing scheme which uses an $N \times N$ score matrix which is composed of scores between all pairs of N templates as an index. After the claimant inputs a query sample, it computes a score for the first template, and searches remaining templates as follows: (1) Generate a vector composed of scores between the query sample and the searched templates, and compute the correlation between the vector and each row of the matrix; (2) Compute a score for the template whose correlation is the highest and go back to (1). This scheme can be regarded as a modification of AESA (Approximating and Eliminating Search Algorithm) [119] to skip the elimination step based on the triangle inequality.

Glenn and Potts [10] proposed another indexing scheme which uses K $(< N)$ pivots which are randomly selected from N templates in the database. They used a score vector composed of scores for K pivots as a feature vector for clustering templates. Gyaourova and Ross [39] proposed an indexing scheme which is similar to the standard pivot-based scheme [22] described in Section 4.2.2. This scheme uses a distance between two score vectors (composed of scores for pivots) as a pseudo-score, and outputs all non-pivots whose pseudo-score is within a certain threshold. They also proposed a technique which concatenates score vectors from multiple modalities to improve the trade-off between the number of score computations and retrieval errors.

However, no studies combined metric space indexing and sequential fusion to reduce identification errors, the number of inputs, and response time in biometric identification at the same time, to the best of our knowledge. This chapter aims at optimizing all of them.

5.1.1 Our Contributions

In this chapter, we make an attempt to optimize the trade-off between identification errors, the number of inputs, and response time, by combining metric space indexing and sequential fusion. The main contributions are as follows:

1. We firstly propose a sequential indexing and fusion framework in biometric identification which is constructed from the following three schemes:

   (I) a pseudo-score based indexing scheme (described in Section 4.2) [4, 6, 22, 32];

   (II) a sequential search scheme which searches non-pivots using pseudo-scores and scores as a clue each time a query sample is input;

   (III) a sequential fusion scheme in identification which handles missing scores (such as the OR rule [14], the LRSI scheme and PPSI scheme proposed in Chapter 3).

[1] In Chapter 4, we referred to the errors in which the system does not compute a score for an object whose distance to the query object is less than the threshold as retrieval errors, since we focused on range queries.

This framework clarifies the generality of our approach: we can use any existing scheme as (I) or (III). At the same time, it also clarifies the novelty of our work: (II) is a new scheme which is necessary to construct our framework.

2. We secondly propose the PPSS (Posterior Probability-based Sequential Search) scheme as (II). This scheme is a modification of the PPS (Posterior Probability-based Search) scheme proposed in Chapter 4 to use *not only pseudo-scores at the current input but past pseudo-scores and scores* to compote posterior probabilities. We discuss the optimal property of this scheme with regard to the trade-off between the number of score computations and retrieval errors. We also explain that this scheme does not have a problem of the learning time.

3. We thirdly propose a technique which optimizes the number of pivots with regard to retrieval errors.

4. We finally evaluate our proposals using a large-scale multi-modal dataset ($N = 1800$ enrollees; one face and two fingerprints) obtained by combining the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21]. The results show that our proposals reduce the number of score computations to 10[%] while keeping the identification error rates (EFRR/EFAR/NFAR) and ANI (the average number of inputs) of the PPSI scheme proposed in Section 3.4.1.

5.1.2 Organization of This Chapter

This chapter is organized as follows. In Section 5.2, we propose our sequential indexing and fusion framework, and explain the example of (I) a pseudo-score based indexing scheme and (III) a sequential fusion scheme. In Section 5.3, we propose the PPSS (Posterior Probability-based Sequential Search) scheme as (II). We first describe the search algorithm, and explain that this scheme does not have a problem of the learning time. We then discuss the optimality of this scheme with regard to the trade-off between the number of score computations and retrieval errors. In Section 5.4, we propose a technique which optimizes the number of pivots. In Section 5.5, we show the experimental results using a large-scale multi-modal dataset. In Section 5.6, we briefly describe parallel indexing and fusion proposed in [A2, B2], a combination of a metric space indexing scheme and a parallel fusion scheme, and compare it with sequential indexing and fusion. Finally, we conclude this chapter in Section 5.7.

5.2 Sequential Indexing and Fusion Framework

We propose a sequential indexing and fusion framework which can be constructed from (I) a pseudo-score based indexing scheme [4, 6, 22, 32], (II) a sequential search scheme, and (III) a sequential fusion scheme in identification which handles missing scores (e.g. the OR rule [14], the LRSI scheme, and the PPSI scheme). We can use any existing scheme as (I) or (III), while (II) is a new scheme which is necessary to construct our framework.

We now explain our framework using the notations shown in Table 5.1. Suppose that the claimant sequentially inputs a different biometric trait (e.g. fingerprint, face, and iris) or instance (e.g. index and middle fingerprints). Let $\mathbb{X}$ be a universe of biometric samples, $\mathbb{Q} = \{q^t | 1 \leq t \leq T\} \subset \mathbb{X}$ be a set of query samples, where q^t is a query sample at the t-th input, and $\mathbb{O} = \{o_i^t | 1 \leq t \leq T, 1 \leq i \leq N\} \subset \mathbb{X}$ be a finite set of templates, where o_i^t is

Table 5.1: Basic Notations Used in This Chapter.

Symbol	Description
$\mathbb{X}$	universe of biometric samples
$\mathbb{Q}$	set of query samples $\mathbb{Q} = \{q^t \mid 1 \le t \le T\} \subset \mathbb{X}$
$\mathbb{O}$	set of templates $\mathbb{O} = \{o_i^t \mid 1 \le t \le T, 1 \le i \le N\} \subset \mathbb{X}$
$\mathbb{O}_{piv}^t$	set of pivots at the t-th input $\mathbb{O}_{piv}^t \subset \mathbb{O}$
$\mathbb{O}_{non}^t$	set of non-pivots at the t-th input $\mathbb{O}_{non}^t = \mathbb{O} \backslash \mathbb{O}_{piv}^t$
$\mathbb{S}_{piv}^t$	set of scores for K^t pivots at the t-th input $\mathbb{S}_{piv}^t = \{s_i^t = s(q^t, o_i^t) \mid o_i^t \in \mathbb{O}_{piv}^t\}$
$\tilde{\mathbb{S}}_{non}^t$	set of pseudo-scores for $N - K^t$ non-pivots at the t-th input $\tilde{\mathbb{S}}_{non}^t = \{\tilde{s}_i^t = \tilde{s}(q^t, o_i^t) \mid o_i^t \in \mathbb{O}_{non}^t\}$
$\mathbb{S}_{non}^t$	set of scores for $M^t - K^t$ $(\le N - K^t)$ non-pivots at the t-th input
N	number of enrollees
T	number of modalities
K^t	number of pivots at the t-th input
M^t	number of score computations at the t-th input $(K^t \le M^t \le N)$
s	score function $s : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$
$\tilde{s}$	pseudo-score function $\tilde{s} : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$
s_i^t	score between q^t and o_i^t $(s_i^t := s(q^t, o_i^t))$
$\tilde{s}_i^t$	pseudo-score between q^t and o_i^t $(\tilde{s}_i^t := \tilde{s}(q^t, o_i^t))$

a template of the i-th enrollee at the t-th input. Let further $s : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ be a score function, and $\tilde{s} : \mathbb{X} \times \mathbb{X} \rightarrow \mathbb{R}$ be a pseudo-score function, in the same way as Chapter 4. We also assume that scores and pseudo-scores are continuous (the discussion below can be easily extended to the discrete case). We further use the following simple notations: $s_i^t := s(q^t, o_i^t)$, $\tilde{s}_i^t := \tilde{s}(q^t, o_i^t)$.

To begin with, our framework constructs, for each modality, an index $\mathbb{I}^t$ in the same way as Section 4.2.1 as follows:

[Indexing Algorithm]

1. Select K^t $(< N)$ pivots from N templates $o_1^t, \cdots, o_N^t$ (at random [22], for example). Note that the number of pivots K^t can be different from modality to modality. Let $\mathbb{O}_{piv}^t$ be a set of pivots, and $\mathbb{O}_{non}^t$ be a set of non-pivots, respectively.
2. Compute a score $s(o_i^t, o_j^t)$ for each pivot $o_i^t \in \mathbb{O}_{piv}^t$ and each non-pivot $o_j^t \in \mathbb{O}_{non}^t$. Let $\mathbb{S}_{pnp}^t = \{s(o_i^t, o_j^t) \mid o_i^t \in \mathbb{O}_{piv}^t, o_j^t \in \mathbb{O}_{non}^t\}$ be a set of scores between pivots and non-pivots ($K(N - K)$ scores in total).
3. Construct an index $\mathbb{I}^t$ using $\mathbb{S}_{pnp}^t$, and store it in the database (the example of $\mathbb{I}^t$ is described in Section 5.2.1).

Then, after the claimant inputs the t-th query sample, our framework carries out the one-to-many matching process and the identification process as follows $(1 \le t \le T)$:

[Authentication Algorithm (t-th input)]

1. Compute a score $s_i^t = s(q^t, o_i^t)$ for each pivot $o_i^t \in \mathbb{O}_{piv}^t$. Let $\mathbb{S}_{piv}^t = \{s_i^t = s(q^t, o_i^t) \mid o_i^t \in \mathbb{O}_{piv}^t\}$ be a set of scores for pivots (K^t scores in total).
2. Compute a pseudo-score $\tilde{s}_i^t = \tilde{s}(q^t, o_i^t)$ for each non-pivot $o_i^t \in \mathbb{O}_{non}^t$ using $\mathbb{S}_{piv}^t$ and $\mathbb{I}^t$ (the example of $\tilde{s}_i^t$ is described in Section 5.2.1). Let $\tilde{\mathbb{S}}_{non}^t = \{\tilde{s}_i^t = \tilde{s}(q^t, o_i^t) \mid o_i^t \in \mathbb{O}_{non}^t\}$

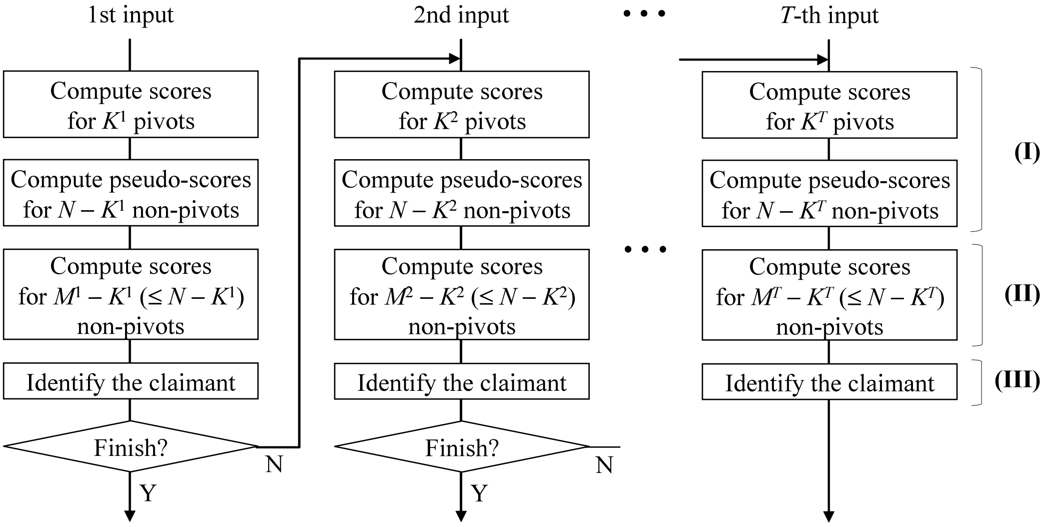

1st input
2nd input
T-th input
Compute scores for K^1 pivots
Compute pseudo-scores for $N - K^1$ non-pivots
Compute scores for $M^1 - K^1$ ($\le N - K^1$) non-pivots
Identify the claimant
Finish?
Y
N
Compute scores for K^2 pivots
Compute pseudo-scores for $N - K^2$ non-pivots
Compute scores for $M^2 - K^2$ ($\le N - K^2$) non-pivots
Identify the claimant
Finish?
Y
N
Compute scores for K^T pivots
Compute pseudo-scores for $N - K^T$ non-pivots
Compute scores for $M^T - K^T$ ($\le N - K^T$) non-pivots
Identify the claimant
(I)
(II)
(III)

Figure 5.1: Flowchart of the authentication process in the sequential indexing and fusion framework ((I): pseudo-score based indexing scheme, (II): sequential search scheme, (III): sequential fusion scheme in identification which handles missing scores).

be a set of pseudo-scores for non-pivots ($N - K^t$ pseudo-scores in total).

3. Narrow down $N - K^t$ non-pivots to $M^t - K^t$ ($K^t \le M^t \le N$) non-pivots using a set of pseudo-scores and scores $\mathbb{W}^t = \{\mathbb{S}^1_{piv}, \cdots, \mathbb{S}^t_{piv}, \tilde{\mathbb{S}}^1_{non}, \cdots, \tilde{\mathbb{S}}^t_{non}, \mathbb{S}^1_{non}, \cdots, \mathbb{S}^{t-1}_{non}\}$, and compute a set of scores for the non-pivots $\mathbb{S}^t_{non}$, where M^t ($= K^t + (M^t - K^t)$) is the number of score computations at the t-th input (the search algorithm is described in Section 5.3).
4. Identify the claimant using a set of scores $\mathbb{S}^t_{tot} = \{\mathbb{S}^1_{piv}, \cdots, \mathbb{S}^t_{piv}, \mathbb{S}^1_{non}, \cdots, \mathbb{S}^t_{non}\}$ ($\sum_{\tau=1}^{t} M^\tau$ scores in total; $\sum_{\tau=1}^{t}(N - M^\tau)$ scores are missing). The identification result is either of the following: $\langle$accept (enrollee ID)$\rangle$, $\langle$reject$\rangle$, or $\langle$require another query sample$\rangle$. If the result is the last one, $t \leftarrow t + 1$ and go back to the step 1 (the example of the decision algorithm is described in Section 5.2.2).

Figure 5.1 shows the flowchart of the authentication process in our framework. The step 1 and 2 are carried out using (I) a pseudo-score based indexing scheme, the step 3 is carried out using (II) a sequential search scheme, and the step 4 is carried out using (III) a sequential fusion scheme. Our framework improves the response time by reducing the number of score computations at the t-th input from N to M^t ($\le N$) using (II), and improves identification accuracy by combining scores from t modalities using (III). M^t is set, for example, to satisfy the requirement for the response time at the t-th input.

We emphasize that the novel component in our framework is (II) a sequential search scheme. As described in Chapter 4, the PPS (Posterior Probability-based Search) scheme searches non-pivots using pseudo-scores. On the other hand, the sequential search scheme searches non-pivots using *not only pseudo-scores at the t-th input* $\tilde{\mathbb{S}}^t_{non}$ *but the past pseudo-scores* $\mathbb{S}^1_{non}, \cdots, \mathbb{S}^{t-1}_{non}$ *and scores* $\mathbb{S}^1_{piv}, \cdots, \mathbb{S}^t_{piv}, \mathbb{S}^1_{non}, \cdots, \mathbb{S}^{t-1}_{non}$, which are available only in multi-biometrics. We demonstrate that these information sources significantly reduces retrieval errors, and hence identification errors, in our experiment in Section 5.5.

5.2.1 Example of a Pseudo-score Based Indexing Scheme

We clarify an index $\mathbb{I}^t$ and a pseudo-score $\tilde{s}_i^t$ computed using (I) a pseudo-score based indexing scheme in our framework. In our experiment in Section 5.5, we used the permutation-based scheme [4, 22] which uses Spearman's Rho as a pseudo-score (described in Section 4.2.3) since it attracted much attention as a very successful scheme [33, 35, 36, 79, 102]. In this case, the index $\mathbb{I}^t$ and the pseudo-score $\tilde{s}_i^t$ are expressed as $\mathbb{I}^t = \{\Pi_{o_i^t} | o_i^t \in \mathbb{O}_{non}^t\}$ and $\tilde{s}_i^t = S_\rho(\Pi_{q^t}, \Pi_{o_i^t})$, respectively, where Π is a permutation.

Instead of the above schemes, we can use any other pseudo-score based indexing scheme such as the standard pivot-based scheme [22], the distance regression-based indexing scheme [32] and BoostMAP [6], all of which are described in Section 4.2.4.

5.2.2 Example of a Sequential Fusion Scheme

We also clarity the decision algorithm using (III) a sequential fusion scheme in identification which handles missing scores. In our experiment in Section 5.5, we used the PPSI (Posterior Probability-based Sequential Identification) scheme described in Section 3.4.1 since it provided the best performance in our experiment in Section 3.6. Let H_i $(0 \leq i \leq N)$ be the following hypothesis as well as Section 3.4.1:

H_i : The claimant is the i-th enrollee. $(1 \leq i \leq N)$

H_0: The claimant is a non-enrollee.

Then, the PPSI scheme computes the posterior probability $P(H_i|\mathbb{S}_{tot}^t)$ $(0 \leq i \leq N)$ using a set of scores $\mathbb{S}_{tot}^t = \{\mathbb{S}_{piv}^1, \cdots, \mathbb{S}_{piv}^t, \mathbb{S}_{non}^1, \cdots, \mathbb{S}_{non}^t\}$, and makes a decision as follows: if the maximum posterior probability exceeds the threshold, accept the corresponding hypothesis H_i and output the corresponding enrollee ID (if $i = 0$, reject); otherwise, require another query sample (if $t = T$, reject).

The posterior probability $P(H_i|\mathbb{S}_{tot}^t)$ $(0 \leq i \leq N)$ can be written, in the same way as Section 3.4.1, as follows:

$$P(H_i|\mathbb{S}_{tot}^t) = \frac{\exp(Z'^t_i)P(H_i)}{\sum_{n=0}^N \exp(Z'^t_n)P(H_n)}, \tag{5.1}$$

where Z'^t_i is the log-likelihood ratio which is given by

$$Z'^t_i = \log \frac{p(\mathbb{S}_{tot}^t|H_i)}{p(\mathbb{S}_{tot}^t|H_0)} \tag{5.2}$$

($p()$ is a probability density function). Here we assume that all scores are independent, and that genuine scores and impostor scores at the t-th input are generated from $f^t()$ and $g^t()$, respectively (although we modeled the enrollee-specific impostor distribution in Section 3.4.1, we assume the impostor distribution common to all enrollees $g^t()$, for simplicity). Then, Z'^t_i can be decomposed as follows:

$$Z'^t_i = \sum_{\tau=1}^t z'^\tau_i, \tag{5.3}$$

where

$$z'^\tau_i = \begin{cases} \log f^\tau(s_i^\tau)/g^\tau(s_i^\tau) & \text{(if } i \neq 0 \text{ and } s_i^\tau \text{ has been computed)} \\ 0 & \text{(otherwise)} \end{cases} \tag{5.4}$$

The important point here is that *missing scores are handled by setting the corresponding log-likelihood ratios to be* 0, which is also described in [76].

In our experiment in Section 5.5, we used the logistic regression model [117] to estimate $\log f^\tau(s_i^\tau)/g^\tau(s_i^\tau)$ as follows:

$$\log f^\tau(s_i^\tau)/g^\tau(s_i^\tau) = w_1^\tau s_i^\tau + w_0^\tau \tag{5.5}$$

(w_1^τ, w_0^τ: regression coefficients). Then, we assumed that the prior probability $P(H_i)$ in (5.1) is uniform (i.e. we used a *noninformative prior* [12]), and computed the posterior probability $P(H_i|\mathbb{S}_{tot}^t)$ using (5.1), (5.3), (5.4), and (5.5).

Instead of the PPSI scheme, we can use other sequential fusion schemes such as the LRSI (Likelihood Ratio-based Sequential Identification) scheme (described in Section 3.4.3) and the OR rule [14]. The LRSI scheme handles missing scores by setting the corresponding log-likelihood ratios to be 0, in the same way as the PPSI scheme. The OR rule handles them by comparing the maximum score among those which have been computed to the threshold. Although we may use other recently proposed parallel fusion schemes in verification which handle missing scores [28, 34, 92], they need to be extended to sequential fusion in identification.

5.3 Posterior Probability-based Sequential Search Scheme

5.3.1 Search Algorithm

A sequential search scheme narrows down $N - K^t$ non-pivots to $M^t - K^t$ ($K^t \leq M^t \leq N$) non-pivots using a set of pseudo-scores and scores $\mathbb{W}^t = \{\mathbb{S}_{piv}^1, \cdots, \mathbb{S}_{piv}^t, \tilde{\mathbb{S}}_{non}^1, \cdots, \tilde{\mathbb{S}}_{non}^t, \mathbb{S}_{non}^1, \cdots, \mathbb{S}_{non}^{t-1}\}$, and computes a set of scores for the non-pivots $\mathbb{S}_{non}^t$. Then, it would be natural to consider, in the same way as Chapter 4, that if the search scheme searches non-pivots in descending order of the posterior probability of being a genuine template, it can minimize retrieval errors. We propose the PPSS (Posterior Probability-based Sequential Search) scheme based on this idea (we discuss its optimality in Section 5.3.3).

In the following, we explain the algorithm of the PPSS scheme. The posterior probability that H_i ($0 \leq i \leq N$) is true after computing $\mathbb{W}^t$ can be written, using Bayes' theorem, as follows:

$$P(H_i|\mathbb{W}^t) = \frac{\exp(Y_i^t)P(H_i)}{\sum_{n=0}^N \exp(Y_n^t)P(H_n)}, \tag{5.6}$$

where Y_i^t is the log-likelihood ratio which is given by

$$Y_i^t = \log \frac{p(\mathbb{W}^t|H_i)}{p(\mathbb{W}^t|H_0)}. \tag{5.7}$$

It follows from (5.6) that if the prior probability $P(H_i)$ ($1 \leq i \leq N$) is uniform, non-pivots sorted by posterior probability $P(H_i|\mathbb{W}^t)$ are the same with those sorted by log-likelihood ratio Y_i^t. Thus, we only have to compute Y_i^t instead of $P(H_i|\mathbb{W}^t)$ in such a case.

Here we assume that all pseudo-scores are independent, and that genuine pseudo-scores and impostor pseudo-scores at the t-th input are generated from $\tilde{f}^t()$, and $\tilde{g}^t()$, respectively, in the same way as Section 3.4.1. However, it should be noted that *pseudo-scores are not* *independent of scores* since they are designed to be highly relevant to scores. Thus, in the case where both a score and a pseudo-score have been computed for the same template, we assume that the two variables are generated from $\hat{f}^t()$ in the case of the genuine template, and from $\hat{g}^t()$ in the case of the impostor template. That is, we assume the following:

$$p(\tilde{s}_i^t|H_j) = \begin{cases} \tilde{f}^t(\tilde{s}_i^t) & (\text{if } i = j) \\ \tilde{g}^t(\tilde{s}_i^t) & (\text{if } i \neq j) \end{cases} \tag{5.8}$$

$$p(s_i^t, \tilde{s}_i^t|H_j) = \begin{cases} \hat{f}^t(s_i^t, \tilde{s}_i^t) & (\text{if } i = j) \\ \hat{g}^t(s_i^t, \tilde{s}_i^t) & (\text{if } i \neq j). \end{cases} \tag{5.9}$$

Then, the log-likelihood ratio Y_i^t in (5.7) can be decomposed as follows:

$$Y_i^t = \sum_{\tau=1}^{t} y_i^\tau, \tag{5.10}$$

where

$$y_i^\tau = \begin{cases} \log f^\tau(s_i^\tau)/g^\tau(s_i^\tau) & (\text{if } o_i^\tau \in \mathbb{O}_{piv}^\tau) \\ \log \tilde{f}^\tau(\tilde{s}_i^\tau)/\tilde{g}^\tau(\tilde{s}_i^\tau) & (\text{if } o_i^\tau \in \mathbb{O}_{non}^\tau \text{ and } s_i^\tau \text{ is missing}) \\ \log \hat{f}^\tau(s_i^\tau, \tilde{s}_i^\tau)/\hat{g}^\tau(s_i^\tau, \tilde{s}_i^\tau) & (\text{if } o_i^\tau \in \mathbb{O}_{non}^\tau \text{ and } s_i^\tau \text{ has been computed}) \end{cases} \tag{5.11}$$

(recall that $s_i^\tau = s(q^\tau, o_i^\tau)$ and $\tilde{s}_i^\tau = \tilde{s}(q^\tau, o_i^\tau)$).

In our experiment in Section 5.5, we estimated $\log \tilde{f}^\tau(\tilde{s}_i^\tau)/\tilde{g}^\tau(\tilde{s}_i^\tau)$ and $\log \hat{f}^\tau(s_i^\tau, \tilde{s}_i^\tau)/\ \hat{g}^\tau(s_i^\tau, \tilde{s}_i^\tau)$ using the logistic regression models in the same way as (5.5), as follows:

$$\log \tilde{f}^\tau(\tilde{s}_i^\tau)/\tilde{g}^\tau(\tilde{s}_i^\tau) = \tilde{w}_1^\tau \tilde{s}_i^\tau + \tilde{w}_0^\tau \tag{5.12}$$

$$\log \hat{f}^\tau(s_i^\tau, \tilde{s}_i^\tau)/\hat{g}^\tau(s_i^\tau, \tilde{s}_i^\tau) = \hat{w}_1^\tau s_i^\tau + \hat{w}_2^\tau \tilde{s}_i^\tau + \hat{w}_0^\tau \tag{5.13}$$

($\tilde{w}_1^\tau$, $\tilde{w}_0^\tau$, $\hat{w}_1^\tau$, $\hat{w}_2^\tau$, $\hat{w}_0^\tau$: regression parameters), and verified the validity of these models.

To sum up, if we assume that the prior probability $P(H_i)$ $(1 \leq i \leq N)$ is uniform and use logistic regression, the algorithm of the PPSS scheme can be written as follows:

1. Compute, for each $o_i^t \in \mathbb{O}_{non}^t$, the log-likelihood ratio Y_i^t using (5.5), (5.10), (5.11), (5.12), and (5.13);

2. Compute scores for $M^t - K^t$ $(\leq N - K^t)$ non-pivots in descending order of Y_i^t.

It follows from (5.5), (5.10), (5.11), (5.12), and (5.13) that each time a score or a pseudo-score is computed, we can update the corresponding log-likelihood ratio Y_i^t by only a few additions and multiplications. Since the computation of a score or a pseudo-score generally requires much more operations (e.g. it follows from (4.4) that the computation of $\tilde{s}_i^\tau = S_\rho(\Pi_{q^\tau}, \Pi_{o_i^\tau})$ requires K^τ additions and K^τ multiplications), the time to update Y_i^t is very small, compared to the computation of a score or a pseudo-score. In [A2], we also confirmed this in the experiment (refer to [A2] for details).

5.3.2 Learning Time

In Chapter 4, we proposed the PPS scheme for range queries, and described that it requires a high computational cost in learning the parameters in logistic regression. The cause of this problem is that *there are a very small number of similar training objects in training objects*, as described in Section 4.3.3. The PPS scheme requires a large number of training objects to avoid shortage of similar training objects.

On the other hand, the PPSS scheme does not have a problem of the learning time. The reason for this is that *the correct answer in biometric identification is different from that in range queries:* the former is a set of genuine templates, while the latter is a set of objects whose distance is less than the threshold. Indeed, the regression coefficients (w_1^τ, w_0^τ) in (5.5) are trained using genuine scores and impostor scores prepared in advance. Similarly, the coefficients $(\tilde{w}_1^\tau, \tilde{w}_0^\tau)$ in (5.12) are trained using genuine pseudo-scores and impostor pseudo-scores. The coefficients $(\hat{w}_1^\tau, \hat{w}_2^\tau, \hat{w}_0^\tau)$ in (5.13) are trained using genuine training samples and impostor training samples, where the training sample is a set of the score and pseudo-score obtained from the same user pair. Thus, all of them can be trained using biometric samples from the same individual and those from the different individuals. It does not need to prepare a large number of biometric samples to obtain genuine scores or pseudo-scores.

In our experiment in Section 5.5, we also confirmed that the time to learn the regression coefficients $(w_1^\tau, w_0^\tau, \tilde{w}_1^\tau, \tilde{w}_0^\tau, \hat{w}_1^\tau, \hat{w}_2^\tau, \hat{w}_0^\tau)$ was very small.

5.3.3 Optimality of the PPSS Scheme

We now discuss the optimality of the PPSS scheme with regard to the trade-off between the number of score computations and retrieval errors. Let us fix templates, pivots, a pseudo-score based indexing scheme and a sequential fusion scheme to be used, and assume that *it is allowed to change a sequential search scheme at each biometric input* (i.e. we can use a certain sequential search scheme at the first input, and then use a different scheme at the second input, and so on). We also refer to the probability that a retrieval error occurs at the t-th input as the *retrieval error probability at the t-th input.* Then, we can prove the following proposition:

Proposition 5.1. *Fix a sequential search scheme at each input from the first to the $(t-1)$-th input. Then, if the posterior probability $P(H_i|\mathbb{W}^t)$ can be perfectly estimated for any $\mathbb{W}^t$, the PPSS scheme minimizes, for any M^t $(K^t \leq M^t \leq N)$, the retrieval error probability at the t-th input among all sequential search schemes used at the t-th input.*

Proof. Let $n_i^t \in \{1, 2, \cdots, N\}$ $(1 \leq i \leq N - M^t)$ be an enrollee ID whose corresponding score at the t-th input is missing after computing scores for K^t pivots and $M^t - K^t$ non-pivots (n_i^t is determined depending on a sequential search scheme at the t-th input and $\mathbb{W}^t$; $\mathbb{W}^t$ is determined depending on $q^1, \cdots, q^t \in \mathbb{X}$ since we fix templates, pivots, an indexing scheme, a fusion scheme, and a search scheme at each input from the first to the $(t-1)$-th input). Then, the retrieval error probability at the t-th input can be written as follows:

$$\int \sum_{i=1}^{N-M^t} P(H_{n_i^t}|\mathbb{W}^t) p(\mathbb{W}^t) d\mathbb{W}^t. \tag{5.14}$$

This value is minimized, for any M^t $(K^t \leq M^t \leq N)$, in the case where we search non-pivots in descending order of $P(H_i|\mathbb{W}^t)$, if $P(H_i|\mathbb{W}^t)$ can be perfectly estimated for any $\mathbb{W}^t$. □

This proposition states that if we fix a sequential search scheme at each input from the first to the $(t-1)$-th input, then the PPSS scheme can minimize, for any number of score computations, the retrieval error probability at the t-th input (i.e. optimizes the trade-off between the number of score computations and retrieval errors at the t-th input). However, it should be noted that we just proved the optimality of the PPSS scheme in a limited sense. For example, we can say that the PPSS scheme minimizes the retrieval error probability at the first input. We can also say that this scheme minimizes the retrieval error probability at the t-th input, under the condition that we use a certain, fixed search scheme (or schemes) until the $(t-1)$-th input. However, we cannot say that continuing to use the PPSS scheme from the first to the t-th input is equivalent to or better than continuing to use any other sequential search schemes from the first to the t-th input, because the schemes until the $(t-1)$-th input are different (and hence the prior distributions of $\mathbb{W}^t$ are different) in this case.

5.4 Optimization of the Number of Pivots

In Section 5.3, we proposed the PPSS scheme to minimize, for any number of score computations, the retrieval error probability under the condition that the number of pivots K^t $(0 \leq t \leq T)$ is fixed. However, the number of pivots K^t can also significantly affect the retrieval error probability. For example, since K^t in the permutation-based scheme represents the length of permutations, it seems that the correlation between pseudo-scores and scores increases with an increase of K^t. However, since the number of score computations for non-pivots $M^t - K^t$ decreases with an increase of K^t, we can expect that as K^t increases, the retrieval error probability decreases until some point and then increases after that. In this section, we propose a technique which optimizes the number of pivots K^t with regard to the retrieval error probability.

The proposed technique estimates, for each modality, the retrieval error probability R^t_{err} in the PPSS scheme in the case where K^t is fixed $(0 < K^t \leq M^t)$, and selects K^t whose R^t_{err} is minimized. That is, it computes K^{t*} which can be expressed as follows:

$$K^{t*} = \underset{0<K^t\leq M^t}{\arg\min}\, R^t_{err}. \tag{5.15}$$

We now explain how to estimate R^t_{err}. Here we assume that pseudo-scores are *positively* correlated with distances and the prior probability $P(H_i)$ $(1 \leq i \leq N)$ is uniform. We first explain the case of the first input $(t = 1)$. After computing a set of pseudo-scores for non-pivots $\tilde{\mathbb{S}}^1_{non} = \{\tilde{s}^1_i = \tilde{s}(q^1, o^1_i)|o^1_i \in \mathbb{O}^1_{non}\}$, the log-likelihood ratio Y^1_i of the non-pivot $o^1_i \in \mathbb{O}^1_{non}$ can be written, using (5.10) and (5.11), as follows:

$$Y^1_i = \log \tilde{f}^\tau(\tilde{s}^1_i)/\tilde{g}^\tau(\tilde{s}^1_i). \tag{5.16}$$

If the log-likelihood ratio function $\log \tilde{f}^\tau()/\tilde{g}^\tau()$ is monotonically decreasing, non-pivots sorted in descending order of Y^1_i are the same with those sorted in ascending order of pseudo-scores. Thus, the PPSS scheme fails to compute a genuine score in the case where a genuine template is included in the $N - K^1$ non-pivots, and more than or equal to $M^1 - K^1$ out of $N - K^1 - 1$ impostor pseudo-scores are smaller than the genuine pseudo-score.

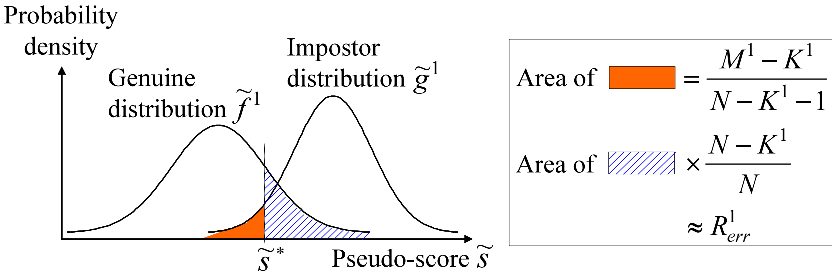

Probability density
Genuine distribution $\tilde{f}^1$
Impostor distribution $\tilde{g}^1$
$\tilde{s}^*$
Pseudo-score $\tilde{s}$
Area of 🟧 $= \frac{M^1 - K^1}{N - K^1 - 1}$
Area of ▨ $\times \frac{N - K^1}{N}$
$\approx R^1_{err}$

Figure 5.2: Relationship between the pseudo-score distributions and the retrieval error probability at the first input.

Let $\tilde{G}^1()$ be a cumulative distribution function of an impostor pseudo-score (i.e. $\tilde{G}^1(\tilde{s}) = \int_{-\infty}^{\tilde{s}} \tilde{g}^1(\tilde{s}_0)d\tilde{s}_0$), and $\tilde{s}^*$ be a pseudo-score which satisfies the following equation:

$$\tilde{G}^1(\tilde{s}^*) = \frac{M^1 - K^1}{N - K^1 - 1}. \tag{5.17}$$

Then, it follows from the asymptotic equipartition property (AEP) [25] that the proportion of the number of impostor pseudo-scores which are smaller than $\tilde{s}^*$ converges to $\tilde{G}^1(\tilde{s}^*)(= (M^1 - K^1)/(N - K^1 - 1))$ in probability as $N \to \infty$. Thus, for N sufficiently large, if the genuine template is included in the $N - K^1$ non-pivots and the genuine pseudo-score is more than $\tilde{s}^*$, the search fails with probability near 1. Otherwise, the search succeeds with probability near 1. Thus, the retrieval error probability R^t_{err} can be approximated as follows:

$$\begin{aligned} R^1_{err} &\approx \frac{N - K^1}{N}(1 - \tilde{F}^1(\tilde{s}^*)) && (5.18) \\ &= \frac{N - K^1}{N}\left(1 - \tilde{F}^1\left(\tilde{G}^{1\ -1}\left(\frac{M^1 - K^1}{N - K^1 - 1}\right)\right)\right), && (5.19) \end{aligned}$$

where $\tilde{F}^1()$ is a cumulative distribution function of a genuine pseudo-score (i.e. $\tilde{F}^1(\tilde{s}) = \int_{-\infty}^{\tilde{s}} \tilde{f}^1(\tilde{s}_0)d\tilde{s}_0$). From (5.19), we can simply estimate R^1_{err} using the genuine pseudo-score distribution $\tilde{f}^1$ and the impostor pseudo-score distribution $\tilde{g}^1$ which are trained in advance. Figure 5.2 shows the relationship between the pseudo-score distributions and the retrieval error probability at the first input.

As for the second input or later ($t \geq 2$), since the log-likelihood ratio Y^t_i of the non-pivot $o^t_i \in \mathbb{O}^t_{non}$ is more complicated than (5.16), it is more difficult to estimate the retrieval error probability R^t_{err}. Thus, for simplicity, we estimate R^t_{err} at the second input or later using (5.19) as well. We demonstrate its effectiveness through experimental evaluation in Section 5.5.

5.5 Experimental Evaluation

We carried out an experiment to evaluate the performance of our proposals. Unlike the metric space indexing datasets such as the Metric Space Library [37], there are very few publicly available biometric datasets which contains thousands or tens of thousands subjects. Gyaourova and Ross [39] created a chimeric multi-modal dataset containing 870 subjects by combining the FERET face database [84] and the WVU fingerprint database [26], and treating all 4 different fingers from one subject as ones from different subjects. In this chapter, we combined the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21] and created a larger chimeric multi-modal dataset (2000 subjects; one face and two fingerprints). This is the largest level of multi-modal datasets which contain fingerprints and faces (which account for the highest and the second highest share in the world market [69], respectively), and can be obtained for free, to the best of our knowledge.

5.5.1 Experimental Set-up

Datasets

The NIST BSSR1 Set3 [77] contains face scores from 3000 subjects (each subject contributed one template and two query samples) using two algorithms ("C" and "G"). We adopted the algorithm "C" and excluded 8 subjects who have inappropriate scores (the values "-1"). Then, we extracted a set of $2992 \times 2992 \times 2$ scores (there were no scores between templates). Since there were no scores between templates which were necessary to construct the index, we used scores between the 1st query samples (i.e. query samples at the first input) and templates to construct the index instead (i.e. we assumed that the 1st query samples were obtained before authentication, as well as templates).

The CASIA-FingerprintV5 [21] contains 20000 fingerprint images (left and right thumb/index/middle/ring finger) from 500 subjects (each subject contributed 5 samples per finger). We made 2000 *chimeric subjects* by grouping the same type of finger of both hands (two fingers) together, and assuming that each set of two fingers was obtained from a different subject. Then, we assumed that the 1st samples (i.e. samples at the first input) as templates and the remaining samples as query samples, and computed $2000 \times 2000 \times 2 \times 5$ scores (including scores between templates) using SourceAFIS Version 1.4 [104].

We combined the above two score datasets, and used 1800 subjects for enrollees ($N = 1800$), 100 subjects for non-enrollees, and the remaining subjects for training the regression coefficients (we describe the training method later in detail). Here, we randomly chose 10 ways of combining the above two datasets and dividing subjects into enrollees, non-enrollees, and those for training. In each case, we carried out the experiment where each of the enrollees and non-enrollees sequentially inputs his/her query sample ($T = 3$). As for the input order, considering that faces can be obtained from a distance (i.e. remote biometrics [113]), we tried the following two ways: "face → left fingerprint → right fingerprint" and "face → right fingerprint → left fingerprint." Here we used the 2nd query samples for faces and each of the 4 query samples for fingerprints. The number of genuine attempts was $144000 (= 1800 \times 10 \times 2 \times 4)$, and that of impostor attempts was $8000 (= 100 \times 10 \times 2 \times 4)$.

Authentication Methods

In our experiment, we evaluated the performance of the following schemes:

1. **PPSI**: the PPSI (Posterior Probability-based Sequential Identification) scheme (described in Section 5.2.2). Each time a query sample is input, it first computes scores for all templates ($N = 1800$ scores in total). Then it computes, for each hypothesis H_i, the posterior probability that H_i is true using all tN scores (t: the number of inputs), and compares it to the threshold ($0 \leq i \leq N$, $1 \leq t \leq T$).

2. **PI$_\Pi$-PPSI**: a simple combination of the permutation-based scheme PI$_\Pi$ (described in Section 5.2.1) and the PPSI scheme (described in Section 5.2.2). That is, each time a query sample is input, it first computes scores for K^t pivots, and scores for $M^t - K^t$ non-pivots in ascending order of pseudo-scores defined by Spearman's Rho (M^t scores in total). Then, it compares the posterior probability $P(H_i|\mathbb{S}^t_{tot})$ to the threshold ($0 \leq i \leq N$, $1 \leq t \leq T$).

3. **PI$_\Pi$-PPSS-PPSI**: our proposal in this chapter which uses the permutation-based scheme PI$_\Pi$, the PPSS (Posterior Probability-based Sequential Search) scheme (described in Section 5.3), and the PPSI scheme. That is, each time a query sample is input, it first computes scores for K^t pivots, and scores for $M^t - K^t$ non-pivots in descending order of the log-likelihood ratio Y^t_i, assuming that $P(H_i)$ is uniform (M^t scores in total). Then, it compares $P(H_i|\mathbb{S}^t_{tot})$ to the threshold ($0 \leq i \leq N$, $1 \leq t \leq T$).

We determined, for each modality, the number of pivots K^t using the optimization technique described in Section 5.4. Here we used multiples of 20 as a candidate for K^t. Then we randomly selected pivots in the same way as [22]. However, since the way of randomly selecting pivots can affect the performance, we tried 10 ways of randomly selecting pivots and evaluated the average and standard deviation of the retrieval error rate (i.e. the error rate that the system does not compute a genuine score) and identification error rates (EFRR/EFAR/NFAR). As for the number of score computations M^t, since the matching speed of the algorithm "C" in the NIST BSSR1 Set3 [77] is unknown, we set M^t for all modalities to the same value M ($= M^1 = M^2 = M^3$), and changed it to various values.

Training Methods

We estimated the posterior probability $P(H_i|\mathbb{S}^t_{tot})$ and the log-likelihood ratio Y^t_i using the logistic regression models, as described in Section 5.2.2 and 5.3.1, respectively. We trained the regression coefficients using the BRLR (Bias-Reduced Logistic Regression) package for R [80]. For training data in the NIST BSSR1 Set3, we used the 1002×1002 score matrix and 1002×1002 pseudo-score matrix obtained from the 2nd query samples of 1002 ($= 2992 - 1800 - 100$) subjects for training. As for the CASIA-FingerprintV5, we used the $100 \times 100 \times 4$ score matrices and $100 \times 100 \times 4$ pseudo-score matrices obtained from the 4 query samples of 100 ($= 2000 - 1800 - 100$) subjects for training.

We also need to estimate pseudo-score distributions to automatically determine the number of pivots K^t, as described in Section 5.4. We used the histogram of genuine pseudo-scores and that of impostor pseudo-scores obtained from the above pseudo-score matrices as a genuine pseudo-score distribution and impostor pseudo-score distribution, respectively.

5.5.2 Experimental Results

Results of the Comparison Experiments

Figure 5.3 shows the relationship between the number of score computations M and the retrieval error rate in the case where the enrollees input all the 3 query samples. Here, the dots and error bars represent the average and $\pm 2\cdot$ standard deviation, respectively, in the case where we tried 10 ways of randomly selecting pivots (we omitted the error bars of **PI$_\Pi$-PPSI** because the standard deviations of **PI$_\Pi$-PPSI** were very small compared to the averages).

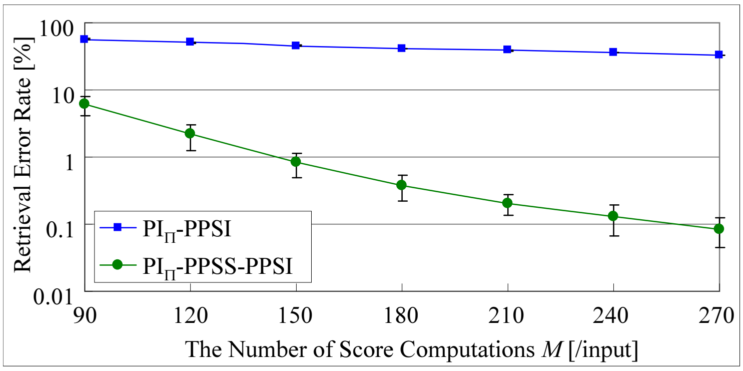


Figure 5.3: Relationship between the number of score computations and the retrieval error rate (dot: average, error bar: ±2· standard deviation).

It was found from Figure 5.3 that **PI$_\Pi$-PPSS-PPSI** significantly outperformed **PI$_\Pi$-PPSI**. For example, the retrieval error rates of **PI$_\Pi$-PPSI** and **PI$_\Pi$-PPSS-PPSI** in the case where $M = 180$ (one tenth of $N = 1800$) were 41.9[%] and 0.37[%], respectively. We consider this was because the PPSS scheme searched non-pivots using not only the current pseudo-scores but the past pseudo-scores and scores, as described in Section 5.2. We also show that the log-likelihood ratio Y_i^t was correctly estimated using the logistic regression models later.

Figure 5.4 shows the relationship between ANI (the average number of inputs) and EFAR/NFAR in the case where the identification threshold at the 3rd input was changed to fix EFRR to be 5[%] (in **PI$_\Pi$-PPSI** and **PI$_\Pi$-PPSS-PPSI**, we tried two cases: $M = 90$ (one twentieth of $N = 1800$) and $M = 180$ (one tenth of $N = 1800$)). Here, in the same way as Figure 5.3, the dots and error bars represent the average and ±2· standard deviation, respectively (we omitted the error bars in the case where EFAR and NFAR were more than 0.1[%] and 1[%], respectively, because the standard deviations were very small; there were also no error bars in **PPSI** because it did not select pivots but computed scores for all templates). The standard deviations were large in the area where the identification error rates were very small (in the area where the error bars reached the lower end, twice the standard deviations were larger than the averages), because the number of errors was very small.

It was found from Figure 5.4 that the identification error rates of **PI$_\Pi$-PPSI** were very high. This was because the retrieval error rate was very high as shown in Figure 5.3. On the other hand, the identification error rates of **PI$_\Pi$-PPSS-PPSI** were much smaller, and were close to those of **PPSI** in the case where $M = 180$. That is, our proposal in this chapter successfully reduced the number of score computations in the PPSI scheme to 10[%] without significantly affecting the identification performance. Interestingly, it was also found that NFAR of **PI$_\Pi$-PPSS-PPSI** was smaller than that of **PPSI** when the average number of inputs was less than 1.8. We consider this was because the PPSS scheme computed almost all genuine scores (retrieval error rate = 0.37[%]) while reducing scores for impostor templates which can cause false accepts.

Although we also evaluated the identification performance of the unimodal biometric system which computes scores for all templates ($N = 1800$ scores in total) and compares the maximum score to the threshold, the result was poor: [EFAR, NFAR] of the faces, left fingerprints, and right fingerprints was [14[%], 77[%]], [11[%], 83[%]], and [11[%], 80[%]],

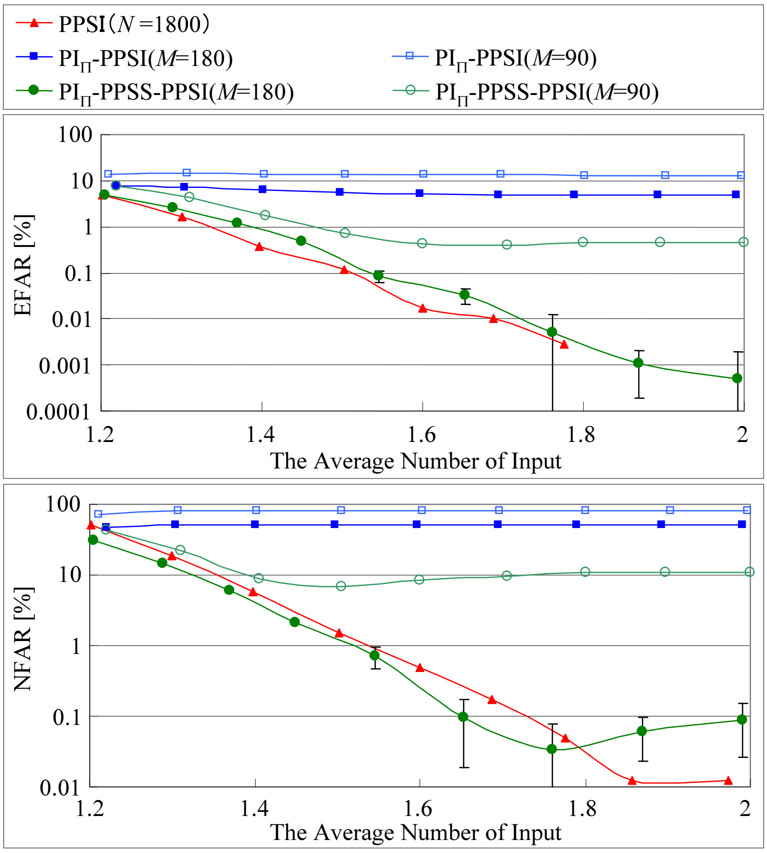


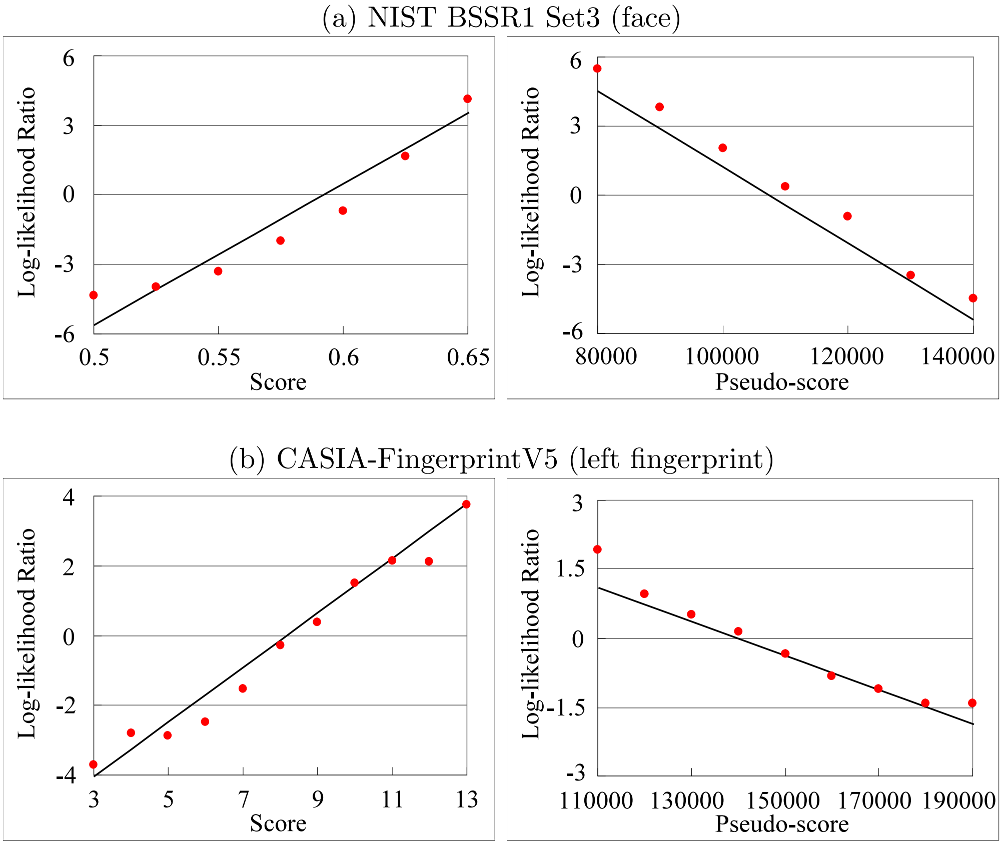
Figure 5.4: Relationship between the average number of inputs and EFAR/NFAR (number in parentheses: number of score computations, dot: average, error bar: ±2· standard deviation).

respectively. Thus, multi-modal fusion was necessary to improve the accuracy. When the average number of inputs was fixed to 1.8, for example, our proposal successfully reduced [EFAR, NFAR] to $[0.0020[\%], 0.034[\%]]$ while reducing the number of score computations to $10[\%]$.

Validation of the Regression Models

To explain more clearly the reason that **PI$_\Pi$-PPSS-PPSI** significantly outperformed **PI$_\Pi$-PPSI**, we investigated the validity of the logistic regression models used in the PPSS scheme.

Figure 5.5 shows the relationship between scores (or pseudo-scores) and log-likelihood ratios in the faces and the left fingerprints (we omitted the result of the right fingerprints which was similar to that of the left fingerprints). Here we chose the first way of randomly selecting pivots (out of 10 ways), and set the number of pivots to be 100. A dot represents the logarithm of the ratio between the genuine histogram and impostor histogram. Here, the genuine histogram was obtained from the last query sample of each enrollee and non-enrollee and the corresponding template (1900 genuine scores or pseudo-scores) and impostor histogram was obtained from the above query sample and the randomly selected impostor template (1900 impostor scores or pseudo-scores). A straight line represents the regression line using the estimated regression coefficients (w_1^t, w_0^t, $\tilde{w}_1^t$, $\tilde{w}_0^t$). From Figure 5.5, it can


(a) NIST BSSR1 Set3 (face)

(b) CASIA-FingerprintV5 (left fingerprint)

Figure 5.5: Relationship between scores (or pseudo-scores) and log-likelihood ratios (left: score, right: pseudo-score, dot: logarithm of the ratio between the genuine histogram and impostor histogram, straight line: estimated regression line).

be seen that the log-likelihood ratios ($\log f^t()/g^t()$ and $\log \tilde{f}^t()/\tilde{g}^t()$) are correctly estimated using the logistic regression models.

Figure 5.6 shows the two dimensional genuine/impostor distributions obtained using the above scores and pseudo-scores, where a straight line represents the contour line of a log-likelihood ratio obtained using the estimated regression coefficients ($\hat{w}_1^t$, $\hat{w}_2^t$, $\hat{w}_0^t$) and "L" is the estimated value of a log-likelihood ratio. It can be seen that there is a correlation between scores and pseudo-scores, and the log-likelihood ratio $\log \hat{f}^t()/\hat{g}^t()$ is correctly estimated using the logistic regression model. The correlation coefficients of the genuine distribution and the impostor distribution were -0.15 and -0.46 in the faces, respectively, and -0.53 and -0.29 in the left fingerprints, respectively.

To sum up, we consider the reason that **PI$_\Pi$-PPSS-PPSI** significantly outperformed **PI$_\Pi$-PPSI** was because the PPSS scheme (1) used not only the current pseudo-scores but the past pseudo-scores and scores, and (2) correctly estimated the log-likelihood ratios using the logistic regression models.

Effectiveness of the Pivot Number Optimization Technique

In Section 5.5.2, we determined, for each modality, the number of pivots K^t using the optimization technique described in Section 5.4. To investigate its effectiveness, we finally

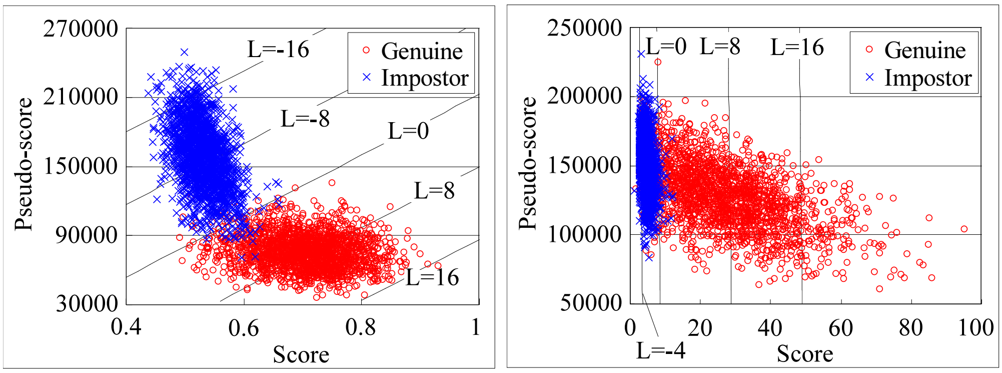


Figure 5.6: Two dimensional genuine/impostor distributions obtained using scores and pseudo-scores (left: NIST BSSR1 Set3 (face), right: CASIA-FingerprintV5 (left fingerprint), straight line: contour line of a log-likelihood ratio, L: estimated value of a log-likelihood ratio).

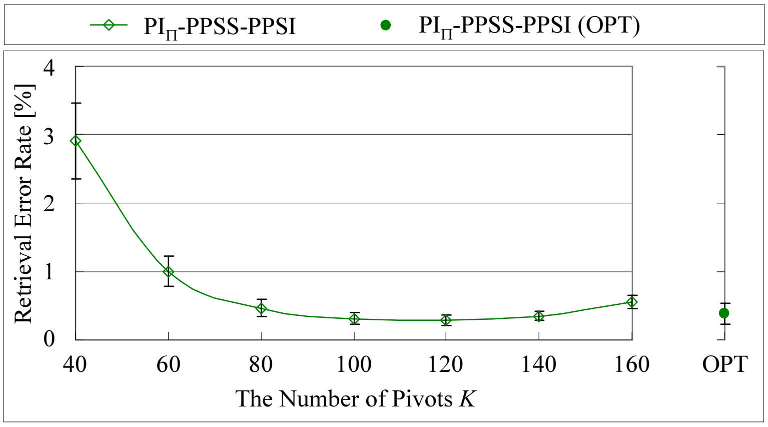


Figure 5.7: Relationship between the number of pivots and the retrieval error rate ($M = 180$, dot: average, error bar: ±2· standard deviation, OPT: the case where we used the optimization technique).

evaluated the retrieval error rate of **PI$_\Pi$-PPSS-PPSI** in the case where we set K^t for all modalities to the same value K ($= K^1 = K^2 = K^3$), and changed it to various values.

Figure 5.7 shows the relationship between the number of pivots K and the retrieval error rate, where the number of score computations was $M = 180$ (the dots and error bars represent the average and ±2· standard deviation, respectively, in the case where we tried 10 ways of randomly selecting pivots). The right edge represents the retrieval error rate in the case where we used the optimization technique (retrieval error rate = 0.37[%]). It was found that as K increased, the retrieval error rate decreased until some point ($K = 120$) and then increased after that (as described in Section 5.4). Interestingly, the optimization technique achieved almost the same performance with the case where the number of pivots was fixed to $K = 120$. Although the number of pivots which was automatically determined was changed depending on the way of dividing subjects and the way of selecting pivots, the determined number was 80, 100, or 120 in most cases, in any of faces, left fingerprints, and right fingerprints. Although the proposed optimization technique roughly estimates the retrieval error probability R^t_{err} as described in Section 5.4, these results demonstrate that it can nonetheless determine the almost optimal number of pivots with regard to retrieval errors.

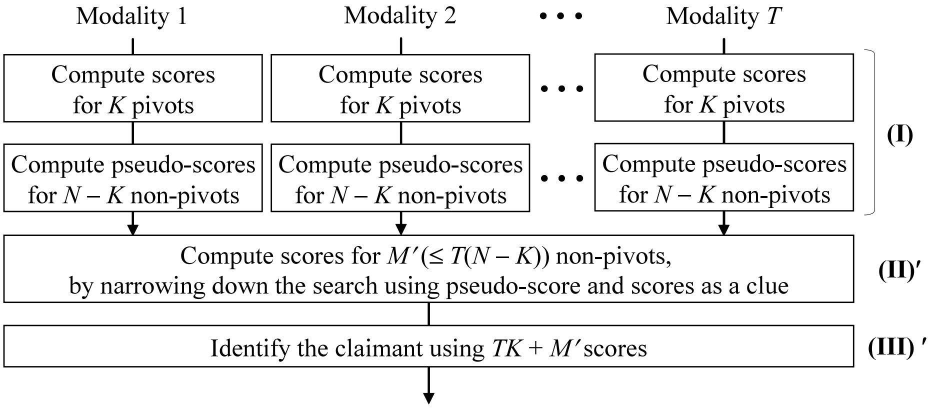

Modality 1
Modality 2
Modality T
Compute scores for K pivots
Compute pseudo-scores for N – K non-pivots
(I)
Compute scores for M′(≤ T(N – K)) non-pivots, by narrowing down the search using pseudo-score and scores as a clue
(II)′
Identify the claimant using TK + M′ scores
(III) ′

Figure 5.8: Flowchart of the authentication process in the parallel indexing and fusion framework ((I): pseudo-score based indexing scheme, (II)′: parallel search scheme, (III)′: parallel fusion scheme in identification which handles missing scores).

5.6 Parallel Indexing and Fusion

We have so far proposed sequential indexing and fusion for biometric identification, and showed its effectiveness through experimental evaluation. In [A2, B2], we also proposed parallel indexing and fusion which identifies a claimant after he/she inputs all query samples. In this section, we briefly describe this.

Here we proposed a parallel indexing and fusion framework which is constructed from (I) a pseudo-score based indexing scheme [4, 6, 22, 32], (II)′ a parallel search scheme which searches non-pivots using pseudo-scores and scores as a clue *after all query samples are input*, and (III)′ a parallel fusion scheme in identification which handles missing scores [76]. In this framework, (I) computes, for each modality, scores for K ($< N$) pivots and pseudo-scores for $N - K$ non-pivots. Then, (II)′ computes scores for M' ($\leq T(N-K)$) non-pivots. Finally, (III)′ identifies a claimant using $TK + M'$ scores (i.e. TK scores for pivots and M' scores for non-pivots; $TN - (TK + M')$ scores are missing). Figure 5.8 shows the flowchart of the authentication process.

As (II)′ a parallel search scheme, we proposed a scheme which selects non-pivots as follows: (1) Compute, for each enrollee, the posterior probability of being identical to the claimant using pseudo-scores and scores; (2) Compute a score for a non-pivot of the enrollee whose posterior probability is the highest and go back to (1). In this dissertation, we refer to this scheme as the PPPS (Posterior Probability-based Parallel Search) scheme to distinguish it from the PPSS scheme.

Since parallel indexing and fusion always requires the claimant to input all query samples, it can cause inconvenience (and this is the reason we have so far focused on sequential indexing and fusion). However, we consider the PPPS scheme has an advantage over the PPSS scheme with regard to retrieval errors. The PPSS scheme searches non-pivots at the t-th input using pseudo-scores and scores at each input from the 1st to the t-th input. In other words, it cannot use pseudo-scores after the $(t+1)$-th input. On the other hand, the PPPS scheme uses all pseudo-scores since it searches non-pivots after the claimant inputs all query samples. Thus, we consider that the PPPS scheme can achieve the lower retrieval error rate than the PPSS scheme.

In [A2], we also proposed a technique which identifies the claimant using (III)$'$ a parallel fusion scheme on the way of searching non-pivots using (II)$'$ a parallel search scheme. Instead of always computing scores for predetermined number of non-pivots, this technique stops searching if it finds out who the claimant is with enough confidence. By doing so, we can reduce the average number of score computations without significantly affecting identification accuracy.

In [A2], we evaluated the performance of our proposals (the PPPS scheme and the above technique) using the NIST BSSR1 Set3 [77], assuming the multiple matchers scenario (i.e. fusion of the algorithm “C” and “G”). The results showed that they significantly improved identification accuracy and reduced the average number of score computations to less than twentieth, compared to the unimodal biometrics. Since the claimant only has to input one query sample, we consider that parallel indexing and fusion is effective especially in the multiple matchers scenario.

5.7 Conclusions

In this chapter, we first proposed a sequential indexing and fusion framework which can be constructed from (I) a pseudo-score based indexing scheme, (II) a sequential search scheme, and (III) a sequential fusion scheme in identification which handles missing scores. Then we proposed the PPSS scheme as (II), and clarified its optimal property with regard to the trade-off between the number of score computations and retrieval errors. We further proposed a technique which optimizes the number of pivots with regard to retrieval errors. The experimental results using a large-scale chimeric multi-modal dataset ($N = 1800$; one face and two fingerprints) showed that our proposals successfully reduced the number of score computations required in the PPSI scheme (described in Section 3.4.1) to 10[%] while keeping the identification error rates (EFRR/EFAR/NFAR) and ANI of the PPSI scheme. Here we did not measure the response time since the NIST BSSR1 Set3 dataset [77] used in the experiment was a score dataset. In [A2, B2], we also measured the response time and showed the effectiveness of our proposals.

Recall that the PPSI scheme proposed in Section 3.4.1 has an optimal property with regard to the trade-off between identification errors and the number of inputs. We only proved the optimality of the PPSS scheme with regard to the trade-off between the number of score computations and retrieval errors in a limited sense (see Section 5.3.3), and the conditions to achieve the optimality of the PPSI scheme are not satisfied in reality (see Section 3.5.3). Nevertheless, we believe that these schemes have made a significant progress towards the optimization of the trade-off between identification errors, the number of inputs, and response time. The optimization of the number of pivots also contributes to this progress.

Chapter 6

Towards Optimal Countermeasures against Wolves and Lambs in Biometrics

Contents

6.1 Introduction

As described in Section 2.3, the accuracy in biometric authentication is different from user to user, and Doddington *et al.* [29] classified users in speaker recognition as follows:

- Sheep: those who are easily recognized (default users);
- Goats: those who are particularly difficult to recognize;
- Lambs: those who are particularly easy to imitate;
- Wolves: those who are particularly successful at imitating others.

Among the above animals, wolves and lambs are particularly problematic because they can cause false accepts against many others and compromise the security of the system. As it is often said that the overall security of the system is determined by the *weakest link in the chain* [5], the overall security of the biometric system can also be determined by these animals.

To have security against wolves and lambs, we have to reduce false accepts caused by them. To reduce the false accepts, we can raise a threshold for a similarity (or lower a threshold for a distance), or use multi-modal biometric fusion schemes which combine multiple sources of information (e.g. fingerprint, face and voice; index and middle fingers). However, such solutions can increase false rejects or the number of inputs, and consequently make the system inconvenient. That is, there is a trade-off between security against wolves and lambs and convenience in biometrics. Although we introduced the previous countermeasures against wolves and lambs in Section 2.3.1, they do not intend to optimize this trade-off.

6.1.1 Our Contributions

In this chapter, we aim at optimizing the trade-off between security against wolves and lambs and convenience in biometrics. Here we consider the verification mode to simplify the problem. We first introduce a taxonomy of wolves and lambs to clarify our target, and define security measures for the animals to enable security evaluation. We then propose a sequential fusion scheme which intends to optimize the above trade-off. Finally, we propose an input order decision scheme to further reduce the number of biometric inputs. More specifically, the contributions of this chapter are as follows:

- We firstly introduce a taxonomy which classifies wolves into three categories (zero-effort wolves, non-adaptive spoofing wolves, and adaptive spoofing wolves) and lambs into two categories (zero-effort lambs and spoofing lambs). This taxonomy clarifies the definition of wolves and lambs, and our target as well. We also define LAP (Lamb Accept Probability), the maximum of the enrollee-specific FAP (False Accept Probability), as a security measure for lambs in the same way as WAP (Wolf Attack Probability) [116], the maximum of the claimant-specific FAP, to enable security evaluation.
- We secondly propose the MLRSV (Minimum Likelihood Ratio-based Sequential Verification) scheme as a sequential fusion scheme which has an optimal property with regard to security against wolves and lambs and convenience. We prove that this scheme keeps WAP and LAP less than a desired value except in the case of adaptive spoofing wolves, if log-likelihood ratios are perfectly estimated. We also prove that this scheme can minimize ANI (the average number of inputs) and FRP (False Reject Probability) under some conditions.

- We thirdly propose an input order decision scheme based on the KL (Kullback-Leibler) divergence [25]. This scheme further reduces ANI of the MLRSV scheme in the case where the KL divergence differs from one modality to another by maximizing the expectation of a genuine log-likelihood ratio at any number of biometric inputs.

- We finally evaluate our schemes using a virtual multi-modal (one face and eight fingerprints) dataset obtained by combining the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21]. The results demonstrate the effectiveness of our proposals.

6.1.2 Organization of This Chapter

This chapter is organized as follows. In Section 6.2, we introduce a taxonomy of wolves and lambs, and security measures for the animals. In Section 6.3, we propose the MLRSV scheme, and proves its optimal property with regard to security against wolves and lambs, false rejects, and the number of inputs. In Section 6.5, we propose the input order decision scheme using the KL divergence. In Section 6.6, we show the experimental results. Finally, we conclude this chapter in Section 6.7.

6.2 Wolves and Lambs

6.2.1 Taxonomy of Wolves and Lambs

Recall that there are two different ways of biometric authentication: verification and identification. In verification, a claimant claims an identity along with a query sample, and the system computes a score between the query sample and a template corresponding to the claimed identity, making a decision whether the claimant is genuine or not. In identification, a claimant only inputs a query sample, and the system computes scores between the query sample and all templates in the database (i.e. one-to-many matching), making a decision who the claimant is. In both cases, since wolves and lambs have high similarity scores against many others, they cause a number of false accepts in the most basic type of biometric system which makes a decision by comparing a score (or scores) to the threshold.

As described in Section 1.1, wolves and lambs are more problematic in identification. Wolves can cause many false accepts even if they do not intend to impersonate others. Then, since they know that they have such threatening biometrics, they may even be encouraged to actively attack many databases to impersonate others. Lambs can also be a threat in identification because they can make the system identify many claimants as them, and lose the availability of the system.

To clarify our target, we introduce a taxonomy which classifies wolves as follows:

- **Zero-effort wolves:** These wolves are those who (happen to) have their own biometrics similar to many others, and impersonate many enrollees by attempting a zero-effort attack [53]. That is, they directly input their own query sample as if they were attempting successful authentication against themselves. Although Doddington *et al.* [29] statistically tested the existence of claimants whose voice has high similarity scores against many enrollees, they fall into this category. They are very powerful in that they cannot be blocked using anti-spoofing measures (e.g. liveness detection; supervising the authentication process) [100] because *they do not spoof the system*. They can be a threat even in the modalities which are very difficult to spoof (e.g. iris, retina [58]).

- **Spoofing wolves:** These wolves are those who make particular effort (e.g. change their biometrics; input an artifact [70]) to impersonate many enrollees, and are further divided into the following sub-categories:
  - **Non-adaptive spoofing wolves:** These wolves are *non-adaptive* in the sense that they input the same query sample against all enrollees. For example, they input an artifact, called a *universal wolf* sample [116], which has extremely high similarity scores against all templates.
  - **Adaptive spoofing wolves:** These wolves are *adaptive* in the sense that they change the query sample depending on the enrollee. For example, they change their voice depending on the enrollee to imitate him/her. We note that this kind of attack is limited to the modality such as voice where the attackers can easily know the query sample of each enrollee, and change their biometrics (or create an artifact) to imitate it.

Similarly, our taxonomy classifies lambs as follows:

- **Zero-effort lambs:** These lambs are those who (happen to) have their own biometrics similar to many others, and enroll their own template without particular effort. They cannot be blocked using anti-spoofing measures, and can be a vulnerability (or threat in identification) in the modalities which are very difficult to spoof.

- **Spoofing lambs:** These lambs are those who make particular effort to enroll the template similar to many others. However, they are *non-adaptive* in the sense that they cannot change the template after the enrollment (as described in Section 2.3, we assume a general biometric authentication system without template update mechanism [90]).

Among them, adaptive spoofing wolves cannot be blocked using our proposals, as described in detail in Section 6.4.5, and are outside the scope of this dissertation.

6.2.2 Security Measures for Wolves and Lambs

We also define security measures for wolves and lambs. Some statistical tests on scores were used to demonstrate the existence of the animals defined by themselves [29, 124], and some score-based measures to quantify recognizability of a user or the extent of the biometric zoo were studied in [88, 89]. However, these tests and measures are not designed to directly evaluate security against wolves and lambs. To evaluate security against wolves and lambs, we should take into account false accepts caused by them, instead of the statistics on scores.

Thus we start with FRR and FAR, the most commonly used error rates in verification, and FRP (False Reject Probability) and FAP (False Accept Probability), the error probabilities corresponding to FRR and FAR. Let $d \in \{0, 1\}$ be a variable which takes 1 or 0 if the final decision result is *accept* or *reject*, respectively. Let further W_1 be the event that a genuine user attempts verification against him/herself, and W_0 be the event that an impostor attempts verification against someone else. Then, as described in Section 3.2.1, FRP and FAP can be written as follows:

$$\begin{aligned} FRP &= P(d=0|W_1) && (6.1) \\ FAP &= P(d=1|W_0), && (6.2) \end{aligned}$$

where $P()$ is a probability mass function, and FRR and FAR can be written as follows:

$$\begin{aligned} FRR &= \frac{\text{The number of false rejects}}{\text{The total number of genuine attempts}} && (6.3) \\ FAR &= \frac{\text{The number of false accepts}}{\text{The total number of impostor attempts}}. && (6.4) \end{aligned}$$

Since FRR and FAR are the average error rates taken over all biometric samples, they do not measure the performance for a particular user.

Taking this into account, we define performance measures for a particular user. Let V be a finite set of claimants, E be a finite set of enrollees. Let further $W_{e,e}$ be the event that $e \in E$ attempts verification against him/herself, $W_{v,*}$ be the event that $v \in V$ attempts an impostor attack against someone else, $W_{*,e}$ be the event that someone else attempts an impostor attack against $e \in E$. Then, we can define the following three error probabilities:

$$\begin{aligned} FRP_e &= P(d=0|W_{e,e}) && (6.5) \\ FAP_{v,*} &= P(d=1|W_{v,*}) && (6.6) \\ FAP_{*,e} &= P(d=1|W_{*,e}). && (6.7) \end{aligned}$$

They are the enrollee-specific FRP, the claimant-specific FAP, and the enrollee-specific FAP, respectively. Goats cause high FRP_e, wolves cause high $FAP_{v,*}$, and lambs cause high $FAP_{*,e}$. We also define the corresponding three error rates:

$$\begin{aligned} FRR_e &= \frac{\text{The number of false rejects caused by } e}{\text{The total number of genuine attempts of } e} && (6.8) \\ FAR_{v,*} &= \frac{\text{The number of false accepts caused by } v}{\text{The total number of impostor attempts of } v} && (6.9) \\ FAR_{*,e} &= \frac{\text{The number of false accepts caused by } e}{\text{The total number of impostor attempts of } e}. && (6.10) \end{aligned}$$

Une *et al.* [116] defined WAP (Wolf Attack Probability) as a security measure for wolves, which can be expressed as follows:

$$WAP = \underset{v \in V}{max} FAP_{v,*}. \quad (6.11)$$

That is, WAP is the false accept probability caused by the most threatening wolf. Similarly, we define LAP (Lamb Accept Probability), a security measure for lambs, as follows:

$$LAP = \underset{e \in E}{max} FAP_{*,e}. \quad (6.12)$$

LAP is the false accept probability caused by the most vulnerable lamb. We further define WAR (Wolf Attack Rate) and LAR (Lamb Accept Rate) as the error rate corresponding to WAP and LAP, respectively:

$$\begin{aligned} WAR &= \underset{v \in V}{max} FAR_{v,*} && (6.13) \\ LAR &= \underset{e \in E}{max} FAR_{*,e}. && (6.14) \end{aligned}$$

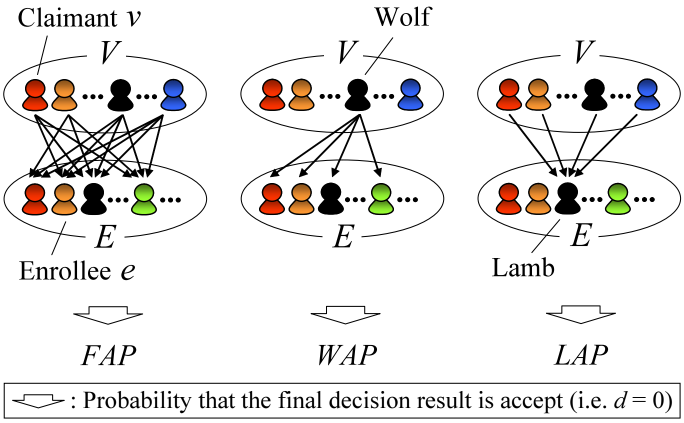

Claimant v
Wolf
V
V
V
E
E
E
Enrollee e
Lamb
FAP
WAP
LAP
: Probability that the final decision result is accept (i.e. $d = 0$)

Figure 6.1: Three kinds of false accept probabilities in verification (FAP/WAP/ LAP). WAP and LAP are the maximum of the claimant-specific FAP and the enrollee-specific FAP, respectively.

We use error probabilities such as FRP, FAP, WAP and LAP in a theoretical analysis, and error rates such as FRR, FAR, WAR, LAR in an experimental evaluation. Figure 6.1 shows the three kinds of false accept probabilities in verification (FAP/WAP/LAP).

As described in Section 3.2, in identification FAR can be divided into EFAR (Enrollee FAR) and NFAR (Non-enrollee FAR), the false accept rate caused by enrollees and non-enrollees, respectively. Thus, security measures for wolves and lambs in identification can be more complicated, and are not defined in this dissertation.

6.3 Minimum Likelihood Ratio-based Sequential Verification Scheme

Our first proposal is the MLRSV (Minimum Likelihood Ratio-based Sequential Verification) scheme, a sequential fusion scheme in verification. This scheme is a modification of the LRSV (Likelihood Ratio-based Sequential Verification) scheme [107] to have security against wolves and lambs by computing the minimum of two log-likelihood ratios obtained using two kinds of user-specific impostor distributions. After describing its algorithm, we clarify its security and optimality from a theoretical point of view.

6.3.1 Assumptions about Scores

Before describing the algorithm of the MLRSV scheme, we make some assumptions about scores. First of all, we focus on sequential fusion of multiple biometric traits (e.g. fingerprint, face and iris) or multiple instances (e.g. index and middle fingers) as described in Section 2.1.1, and assume that all scores are independent.

Let $s^{(t)} \in \mathbb{R}$ be a score between the t-th query sample and the corresponding template (in this paper, we assume that $s^{(t)}$ is continuous; the discussion below can be easily extended to the discrete case). In addition to the independence of scores, we assume, in the same way as Section 3.4.2, that genuine scores are generated from a genuine distribution $f^{(t)}$ which is common to all enrollees:

$$f^{(t)}(s^{(t)}) = p(s^{(t)}|W_{e,e}), \tag{6.15}$$

where $p()$ is a probability density function. The reason we assume a genuine distribution common to all enrollees is that (1) generally very few genuine scores per enrollee (e.g. 2 or 3 scores) are available as training samples, and (2) we cannot train the enrollee-specific genuine distributions if each enrollee presents only one biometric sample during enrollment, as described in Section 3.4.2. $f^{(t)}$ is trained using genuine scores from templates enrolled in the database or any other biometric samples which are collected in advance.

We further define the following two impostor distributions:

$$\begin{aligned} g_v^{(t)}(s^{(t)}) &= p(s^{(t)}|W_{v,*}) && (6.16) \\ g_e^{(t)}(s^{(t)}) &= p(s^{(t)}|W_{*,e}). && (6.17) \end{aligned}$$

That is, $g_v^{(t)}$ is a claimant-specific impostor distribution, and $g_e^{(t)}$ is an enrollee-specific impostor distribution. Suppose $v \in V$ attempts verification against $e \in E$. Then, $g_v^{(t)}$ is trained using scores between a query sample of $v \in V$ and biometric samples other than the template of $e \in E$. We refer to the biometric samples used for estimating $g_v^{(t)}$ as *dummy-templates*. For example, we can use all or part of templates of other enrollees in the database as dummy-templates. We can also use any other biometric samples which are collected separately from the templates (e.g. biometric samples collected for performance evaluation). Similarly, $g_e^{(t)}$ is trained using scores between a template of $e \in E$ and dummy-templates. Note that $f^{(t)}$ and $g_e^{(t)}$ are trained before authentication (e.g. right after enrollment), while $g_v^{(t)}$ is trained after $v \in V$ inputs the t-th query sample.

As will be described in detail in Section 6.3.2, $f^{(t)}$, $g_v^{(t)}$ and $g_e^{(t)}$ are used for estimating the following two kinds of log-likelihood ratios:

$$\begin{aligned} Z_v^{(t)} &= \log \frac{f^{(t)}(s^{(t)})}{g_v^{(t)}(s^{(t)})} && (6.18) \\ Z_e^{(t)} &= \log \frac{f^{(t)}(s^{(t)})}{g_e^{(t)}(s^{(t)})}. && (6.19) \end{aligned}$$

Thus, it is also possible to directly train $Z_v^{(t)}$ and $Z_e^{(t)}$, instead of training $f^{(t)}$, $g_v^{(t)}$, and $g_e^{(t)}$ (in this case, $Z_e^{(t)}$ is trained before authentication, and $Z_v^{(t)}$ is trained after $v \in V$ inputs the t-th query sample). For example, we can use logistic regression [12] which models $Z_v^{(t)}$ and $Z_e^{(t)}$ as follows:

$$\begin{aligned} Z_v^{(t)} &= w{1}_v^{(t)} s^{(t)} + w{0}_v^{(t)} && (6.20) \\ Z_e^{(t)} &= w{1}_e^{(t)} s^{(t)} + w{0}_e^{(t)}, && (6.21) \end{aligned}$$

where $w{1}_v^{(t)}$, $w{0}_v^{(t)}$, $w{1}_e^{(t)}$, and $w{0}_e^{(t)}$ are regression coefficients. We also show the validity of this model in our experiments in Section 6.6.

6.3.2 Algorithm

We now describe the algorithm of the MLRSV scheme. Suppose the claimant $v \in V$ attempts verification against the enrollee $e \in E$. Let $\boldsymbol{r^{(t)}} = (r_1^{(t)}, \cdots, r_N^{(t)})$ be a sequence of scores between the t-th query sample of $v \in V$ and N dummy-templates. $\boldsymbol{r^{(t)}}$ is just used to train $g_v^{(t)}$ (or $Z_v^{(t)}$). Let further $\boldsymbol{s^t} = (s^{(1)}, \cdots, s^{(t)})$ be a sequence of scores between the query

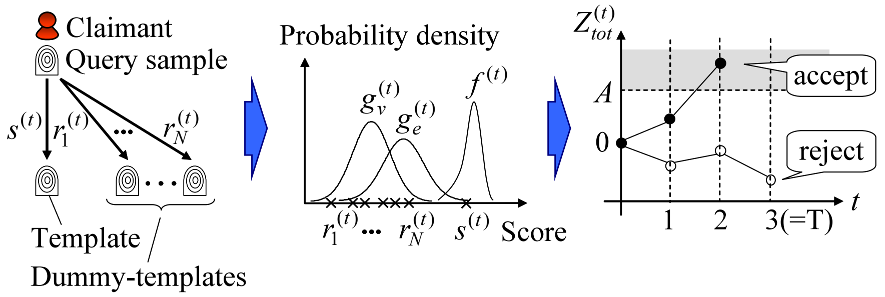


Figure 6.2: Overview of the MLRSV scheme in the case where the score distributions $f^{(t)}$, $g_e^{(t)}$, and $g_v^{(t)}$ are trained. $f^{(t)}$ and $g_e^{(t)}$ are trained in advance, while $g_v^{(t)}$ is trained using $\boldsymbol{r^{(t)}} = (r_1^{(t)}, \cdots, r_N^{(t)})$. After computing $s^{(t)}$, $Z_{tot}^{(t)}$ is updated and compared to A. Here, two examples are given: one results in acceptance at the second input; the other results in rejection ($T = 3$).

samples of $v \in V$ and the templates of $e \in E$, and H_i ($i = 0, 1$) be the following hypotheses:

H_1: The claimant is a genuine user.

H_0: The claimant is an impostor.

Then, since we assume that all scores are independent, the log-likelihood ratio after obtaining $\boldsymbol{s^t}$ can be written as follows:

$$\log \frac{p(\boldsymbol{s^t}|H_1)}{p(\boldsymbol{s^t}|H_0)} = \sum_{\tau=1}^{t} \log \frac{p(s^{(\tau)}|H_1)}{p(s^{(\tau)}|H_0)}. \tag{6.22}$$

The MLRSV scheme estimates the two kinds of log-likelihood ratios $Z_v^{(\tau)}$ and $Z_e^{(\tau)}$ using the two kinds of impostor distributions $g_v^{(\tau)}$ and $g_e^{(\tau)}$ (see (6.18) and (6.19)), and adopts the minimum value of them (i.e. the one which causes less false accepts) as $\log p(s^{(\tau)}|H_1)/p(s^{(\tau)}|H_0)$. That is, the MLRSV scheme computes $Z_{tot}^{(t)}$ after obtaining $\boldsymbol{s^t}$ as follows:

$$Z_{tot}^{(t)} = \sum_{\tau=1}^{t} Z_{min}^{(\tau)}, \tag{6.23}$$

where

$$Z_{min}^{(\tau)} = \min \left\{ Z_v^{(\tau)}, Z_e^{(\tau)} \right\}. \tag{6.24}$$

Then, it compares $Z_{tot}^{(t)}$ to a verification threshold A, and makes the following decision: If $Z_{tot}^{(t)}$ is greater than or equal to A, accept the hypothesis H_1 (i.e. accept the claimant); Otherwise if the number of inputs t reaches the maximum value T (i.e. the number of modalities), accept the hypothesis H_0 (i.e. reject the claimant); Otherwise, require another biometric input.

To sum up, the algorithm of the MLRSV scheme is as follows:

[The MLRSV Algorithm]

```
1. $t \leftarrow 1$, $Z_{tot}^{(0)} \leftarrow 0$;
2. Compute $\boldsymbol{r^{(t)}} = (r_1^{(t)}, \cdots, r_N^{(t)})$, and train $g_v^{(t)}$ (or $Z_v^{(t)}$);
3. Compute $s^{(t)}$;
```

```
4. $Z_v^{(t)} \leftarrow \log \frac{f^{(t)}(s^{(t)})}{g_v^{(t)}(s^{(t)})}$, $Z_e^{(t)} \leftarrow \log \frac{f^{(t)}(s^{(t)})}{g_e^{(t)}(s^{(t)})}$;
5. $Z_{min}^{(t)} \leftarrow \min \left\{ Z_v^{(t)}, Z_e^{(t)} \right\}$, $Z_{tot}^{(t)} \leftarrow Z_{tot}^{(t-1)} + Z_{min}^{(t)}$;
6. If $Z_{tot}^{(t)} \geq A$, accept the claimant; Otherwise if $t = T$, reject the claimant; Otherwise,
   $t \leftarrow t + 1$ and go to 2).
```

Figure 6.2 shows the overview of the MLRSV scheme.

6.4 Theoretical Properties of the MLRSV scheme

We now show some theoretical properties of the MLRSV scheme. We first briefly explain the outline of them. Let δ be a sequential fusion algorithm, and $WAP(\delta)$ be WAP of δ. We apply the same rule to other performance measures such as LAP, FRP, and FAP. Let further δ_0 be the MLRSV algorithm. In this paper, we prove the following three properties of the MLRSV scheme:

- **Security against wolves and lambs:** The MLRSV scheme can keep the false accept probability caused by any claimant and any enrollee, except for adaptive spoofing wolves, less than a desired value: $WAP(\delta_0) \leq \alpha$ and $LAP(\delta_0) \leq \alpha$, where α is a required WAP and LAP (Theorem 6.1).

- **Optimality with regard to FRP:** The MLRSV scheme can minimize, for any enrollee $e \in E$, the false reject probability among all sequential fusion schemes with the same false accept probability: $FRP_e(\delta_0) \leq FRP_e(\delta)$ for any δ such that $FAP_{*,e}(\delta) = FAP_e(\delta_0)$ (Theorem 6.2).

- **Asymptotic optimality with regard to ANI:** The MLRSV scheme can minimize, for any enrollee $e \in E$, ANI among all sequential fusion schemes in the asymptotic setting where both the false accept probability and the false reject probability ($FAP_{*,e}$ and FRP_e) are sufficiently small (Theorem 6.3).

The first property guarantees the security of the MLRSV scheme in terms of false accepts, while the second one guarantees the optimality with regard to false rejects. We can significantly reduce both false accepts and false rejects by setting the maximum number of inputs (i.e. the number of modalities) T very large. Then, the third property guarantees that ANI of this scheme can achieve almost the minimum value.

In the rest of this section, we formally describe these theoretical properties. Table 6.1 shows the notations used there.

6.4.1 Proof of the Security against Wolves and Lambs

The MLRSV scheme models both the claimant-specific impostor distribution $g_v^{(t)}$ and the enrollee-specific impostor distribution $g_e^{(t)}$, and adopts the one which causes less false accepts (i.e. the minimum value of $Z_v^{(t)}$ and $Z_e^{(t)}$). By this means, it achieves security against any claimant and any enrollee except for adaptive spoofing wolves, if $Z_v^{(t)}$ and $Z_e^{(t)}$ are perfectly estimated:

Table 6.1: Notations used in describing theoretical properties of the MLRSV scheme.

Symbol	Description
δ	a sequential fusion algorithm
δ_0	the MLRSV algorithm
V	a set of claimants
E	a set of enrollees
FRP_e	FRP (False Reject Probability) of $e \in E$ (see (6.5))
$FAP_{v,*}$	FAP (False Accept Probability) of $v \in V$ (see (6.6))
$FAP_{*,e}$	FAP (False Accept Probability) of $e \in E$ (see (6.7))
WAP	WAP (Wolf Attack Probability) (see (6.11))
LAP	LAP (Lamb Accept Probability) (see (6.12))
ANI_e	ANI (the average number of inputs) of $e \in E$
$f^{(t)}$	a genuine distribution common to all enrollees (see (6.15))
$g_v^{(t)}$	an impostor distribution of $v \in V$ (see (6.16))
$g_e^{(t)}$	an impostor distribution of $e \in E$ (see (6.17))
$Z_v^{(t)}/Z_e^{(t)}$	two kinds of log-likelihood ratios (see (6.18) and (6.19))
A	a verification threshold of the MLRSV scheme
T	the maximum number of inputs (the number of modalities)
α	a required WAP and LAP
β	a required FRP_e
$\Delta(\alpha, \beta)$	a set of sequential fusion algorithms whose $FAP_{*,e}$ and FRP_e do not exceed α and β, respectively (see (6.48))

Theorem 6.1. (Security of the MLRSV scheme against wolves and lambs) *If (i) the log-likelihood ratios $Z_v^{(t)}$ and $Z_e^{(t)}$ $(1 \leq t \leq T)$ are perfectly estimated in the MLRSV algorithm δ_0, then we have $FAP_{v,*}(\delta_0) \leq \alpha$ and $FAP_{*,e}(\delta_0) \leq \alpha$ for any $v \in V$ and $e \in E$, and hence we have $WAP(\delta_0) \leq \alpha$ and $LAP(\delta_0) \leq \alpha$ (except in the case of adaptive spoofing wolves), where $\alpha = e^{-A}$ and A is a verification threshold.*

Proof. We prove that if the condition (i) holds, we have $FAP_{v,*}(\delta_0) \leq e^{-A}$ for any $v \in V$. We can prove that we have $FAP_{*,e}(\delta_0) \leq e^{-A}$ for any $e \in E$ in a similar way by interchanging $v \in V$ and $e \in E$.

Suppose the claimant $v \in V$ attempts an impostor attack against each enrollee $e(\neq v) \in E$ in the set E. Let $Z_{vtot}^{(t)} = \sum_{\tau=1}^{t} Z_v^{(\tau)}$, and $\boldsymbol{S_v^t}$ be a set of score sequences $\boldsymbol{s^t} = (s^{(1)}, \cdots, s^{(t)})$ such that $Z_{vtot}^{(1)}, \cdots, Z_{vtot}^{(t-1)} < A$ and $Z_{vtot}^{(t)} \geq A$. That is, $\boldsymbol{S_v^t}$ is a set of $\boldsymbol{s^t}$ such that the attack of $v \in V$ results in success at the t-th input, in the case where *only $g_v^{(t)}$ is used as an impostor distribution model*. Note that $\boldsymbol{S_v^t}$ is independent of $e \in E$ because we assume that $v \in V$ is *non-adaptive* and $g_v^{(t)}$ (and hence $Z_{vtot}^{(t)}$) is independent of $e \in E$.

Since the MLRSV algorithm δ_0 models both $g_v^{(t)}$ and $g_e^{(t)}$, and adopts the one which causes less false accepts (i.e. the minimum value of $Z_v^{(t)}$ and $Z_e^{(t)}$), it is not always true that the attack results in success (i.e. $Z_{tot}^{(t)} \geq A$) in such a score sequence. Thus, $FAP_{v,*}$ in (6.6) can be bounded as follows:

$$FAP_{v,*}(\delta_0) \leq \sum_{t=1}^{T} \int_{\boldsymbol{S_v^t}} p(\boldsymbol{s^t}|W_{v,*})d\boldsymbol{s^t} \tag{6.25}$$

If (i) holds, $Z_v^{(t)}$ can be written, using (6.15), (6.16) and (6.18), as

$$Z_v^{(t)} = \log \frac{p(s^{(t)}|W_{e,e})}{p(s^{(t)}|W_{v,*})}. \tag{6.26}$$

Thus, for any $\boldsymbol{s^t} \in \boldsymbol{S_v^t}$, we can derive the following inequality:

$$\begin{aligned} \frac{p(\boldsymbol{s^t}|W_{e,e})}{p(\boldsymbol{s^t}|W_{v,*})} &= \frac{\prod_{\tau=1}^{t} p(s^{(\tau)}|W_{e,e})}{\prod_{\tau=1}^{t} p(s^{(\tau)}|W_{v,*})} && (6.27) \\ &= \exp\left[\sum_{\tau=1}^{t} Z_v^{(\tau)}\right] && (6.28) \\ &= \exp\left[Z_{vtot}^{(t)}\right] && (6.29) \\ &\geq e^A. && (6.30) \end{aligned}$$

Using (6.25) and (6.30), we have

$$\begin{aligned} FAP_{v,*}(\delta_0) &\leq e^{-A} \cdot \left[\sum_{t=1}^{T} \int_{\boldsymbol{S_v^t}} p(\boldsymbol{s^t}|W_{e,e})d\boldsymbol{s^t}\right] && (6.31) \\ &\leq e^{-A}. && (6.32) \end{aligned}$$

The last inequality follows from the fact that the probability that either of $Z_{vtot}^{(1)}, \cdots, Z_{vtot}^{(T-1)}$ or $Z_{vtot}^{(T)}$ exceeds (or reaches) A is less than or equal to 1. □

It should be noted that this theorem holds irrespective of the input order (i.e. irrespective of which modality to start with). The optimality of the MLRSV scheme (Lemma 6.1 and Theorem 6.3) also holds irrespective of the input order. However, we assume that the input order is fixed in proving the optimality of the MLRSV scheme with regard to FRP (Theorem 6.2).

6.4.2 Proof of the Optimality with Regard to FRP

We then prove the optimality of the MLRSV scheme with regard to FRP. Here we assume that the input order is fixed as mentioned above. Then, the following theorem holds:

Theorem 6.2. (Optimality of the MLRSV scheme with regard to FRP) *If (i) the log-likelihood ratios $Z_v^{(t)}$ and $Z_e^{(t)}$ $(1 \leq t \leq T)$ are perfectly estimated in the MLRSV algorithm δ_0, then we have $FRP_e(\delta_0) \leq FRP_e(\delta)$ for any sequential fusion algorithm δ such that $FAP_{*,e}(\delta) = FAP_{*,e}(\delta_0)$.*

Proof. Suppose $e \in E$ attempts verification against him/herself. Let $\boldsymbol{C_0^t}$ and $\boldsymbol{C^t}$ be a set of score sequences $\boldsymbol{s^t}$ such that the verification attempt results in *reject* (i.e. $d = 0$) at the t-th input in δ_0 and δ, respectively. Then, we have

$$
\begin{aligned}
& FRP_e(\delta) - FRP_e(\delta_0) && (6.33)\\
&= \sum_{t=1}^{T} \int_{\boldsymbol{C^t}} p(\boldsymbol{s^t}|W_{e,e}) d\boldsymbol{s^t} - \sum_{t=1}^{T} \int_{\boldsymbol{C_0^t}} p(\boldsymbol{s^t}|W_{e,e}) d\boldsymbol{s^t}. && (6.34)\\
&= \sum_{t=1}^{T} \int_{\boldsymbol{C^t} \cap \bar{\boldsymbol{C}}_{\boldsymbol{0}}^{\boldsymbol{t}}} p(\boldsymbol{s^t}|W_{e,e}) d\boldsymbol{s^t} - \sum_{t=1}^{T} \int_{\boldsymbol{C_0^t} \cap \bar{\boldsymbol{C}}^{\boldsymbol{t}}} p(\boldsymbol{s^t}|W_{e,e}) d\boldsymbol{s^t}. && (6.35)
\end{aligned}
$$

In (6.35), we excluded the common set $\boldsymbol{C^t} \cap \boldsymbol{C_0^t}$ from both terms.

If (i) holds, $Z_e^{(t)}$ can be written, using (6.15), (6.17) and (6.19), as follows:

$$
Z_e^{(t)} = \log \frac{p(s^{(t)}|W_{e,e})}{p(s^{(t)}|W_{*,e})}. \tag{6.36}
$$

Furthermore, since $e \in E$ attempts verification against him/herself, $g_v^{(t)}$ is equal to $g_e^{(t)}$, and hence $Z_{min}^{(t)} = Z_v^{(t)} = Z_e^{(t)}$. Thus, for any $\boldsymbol{s^t} \in \boldsymbol{C_0^t}$, we can derive the following inequality:

$$
\begin{aligned}
\frac{p(\boldsymbol{s^t}|W_{e,e})}{p(\boldsymbol{s^t}|W_{*,e})} &= \frac{\prod_{\tau=1}^{t} p(s^{(\tau)}|W_{e,e})}{\prod_{\tau=1}^{t} p(s^{(\tau)}|W_{*,e})} && (6.37)\\
&= \exp\left[\sum_{\tau=1}^{t} Z_e^{(\tau)}\right] && (6.38)\\
&= \exp\left[Z_{tot}^{(t)}\right] && (6.39)\\
&< e^A. && (6.40)
\end{aligned}
$$

Note that this inequality holds for any sequential fusion algorithm δ, as long as we fix the input order. Conversely, for any $\boldsymbol{s^t} \in \bar{\boldsymbol{C}}_{\boldsymbol{0}}^{\boldsymbol{t}}$ (i.e. any score sequence which results in acceptance at the t-th input in the MLRSV algorithm δ_0), we have

$$
\frac{p(\boldsymbol{s^t}|W_{e,e})}{p(\boldsymbol{s^t}|W_{*,e})} = \exp\left[Z_{tot}^{(t)}\right] \geq e^A. \tag{6.41}
$$

By (6.35), (6.40), and (6.41), if $FAP_{*,e}(\delta) = FAP_{*,e}(\delta_0)$, then

$$
\begin{aligned}
& FRP_e(\delta) - FRP_e(\delta_0) && (6.42)\\
\geq\ & e^A \cdot \left[\sum_{t=1}^{T} \int_{\boldsymbol{C^t} \cap \bar{\boldsymbol{C}}_{\boldsymbol{0}}^{\boldsymbol{t}}} p(\boldsymbol{s^t}|W_{*,e}) d\boldsymbol{s^t} - \sum_{t=1}^{T} \int_{\boldsymbol{C_0^t} \cap \bar{\boldsymbol{C}}^{\boldsymbol{t}}} p(\boldsymbol{s^t}|W_{*,e}) d\boldsymbol{s^t}\right] && (6.43)\\
=\ & e^A \cdot \left[\sum_{t=1}^{T} \int_{\boldsymbol{C^t}} p(\boldsymbol{s^t}|W_{*,e}) d\boldsymbol{s^t} - \sum_{t=1}^{T} \int_{\boldsymbol{C_0^t}} p(\boldsymbol{s^t}|W_{*,e}) d\boldsymbol{s^t}\right] && (6.44)\\
=\ & e^A \cdot [(1 - FAP_{*,e}(\delta)) - (1 - FAP_{*,e}(\delta_0))] && (6.45)\\
=\ & e^A \cdot [FAP_{*,e}(\delta_0) - FAP_{*,e}(\delta)] && (6.46)\\
=\ & 0. && (6.47)
\end{aligned}
$$

In (6.44), we added the common set $\boldsymbol{C^t} \cap \boldsymbol{C^t_0}$ to both terms. □

This theorem means that the MLRSV scheme can achieve, for any $e \in E$, the minimum FRP_e among all sequential fusion schemes with the same $FAP_{*,e}$. In other words, the MLRSV scheme can provide an optimal trade-off between $FRP_{*,e}$ and FAP_e.

6.4.3 Lower bound for ANI

We finally prove the optimality of the MLRSV scheme with regard to ANI in the asymptotic setting where the identification error probabilities (both the false accept probability and the false reject probability) are sufficiently small. To prove this optimality, we first show the lower bound for ANI of a sequential fusion algorithm δ.

Let $\Delta(\alpha, \beta)$ the following set of sequential fusion algorithms:

$$\Delta(\alpha, \beta) = \{\delta : FAP_{*,e}(\delta) \leq \alpha, FRP_e(\delta) \leq \beta\}, \tag{6.48}$$

where β is a required FRP_e. The MLRSV algorithm δ_0 satisfies this requirement by setting the threshold $A = \log \alpha^{-1}$ (so that $FAP_{*,e} \leq \alpha$; see Theorem 1), and the number of modalities T sufficiently large (so that $FRP_e(\delta_0) \leq \beta$).

Let ANI_e be ANI of $e \in E$. Then, the following lemma gives the lower bound for ANI_e of any $\delta \in \Delta(\alpha, \beta)$:

Lemma 6.1. (Lower bound for ANI) *If (ii) the KL (Kullback-Leibler) divergence between $f^{(t)}$ and $g_e^{(t)}$ takes a value D_e independently of the modality, then*

$$\inf_{\delta \in \Delta(\alpha,\beta)} ANI_e(\delta) \geq \frac{\log \alpha^{-1}}{D_e} \quad \textit{as} \quad \max(\alpha, \beta) \to 0, \tag{6.49}$$

where $\inf_{\delta \in \Delta(\alpha,\beta)} X$ *is the infimum of X over $\delta \in \Delta(\alpha, \beta)$.*

Proof. Suppose $e \in E$ attempts verification against him/herself in any sequential fusion algorithm $\delta \in \Delta(\alpha, \beta)$. The KL divergence $D(f^{(t)}||g_e^{(t)})$ between $f^{(t)}$ and $g_e^{(t)}$ is written as follows [25]:

$$D(f^{(t)}||g_e^{(t)}) = \int f^{(t)}(s^{(t)}) \log \frac{f^{(t)}(s^{(t)})}{g_e^{(t)}(s^{(t)})} ds^{(t)}. \tag{6.50}$$

If the condition (ii) holds, (6.50) can be also written as follows:

$$\begin{aligned} D_e &= D(f^{(t)}||g_e^{(t)}) && (6.51)\\ &= \int f^{(t)}(s^{(t)}) \log \frac{f^{(t)}(s^{(t)})}{g_e^{(t)}(s^{(t)})} ds^{(t)} && (6.52)\\ &= E\left[Z_e^{(t)}\right]. && (6.53) \end{aligned}$$

Here we introduce a random variable t^* which represents the number of biometric inputs required to terminate the verification process. Then, $ANI_e(\delta)$ can be expressed as

$$ANI_e(\delta) = E[t^*]. \tag{6.54}$$

Since $Z_e^{(t)}$ $(1 \leq t \leq t^*)$ is independently distributed with mean D_e (see (6.53)), $E[Z_e^{(1)} + \cdots + Z_e^{(t_e^*)}]$ is decomposed as follows,

$$E\left[Z_e^{(1)} + \cdots + Z_e^{(t^*)}\right] = D_e \cdot E\left[t^*\right]. \tag{6.55}$$

This equation is known as Wald's identity (or Wald's equation) [13, 120]. Using (6.54) and (6.55), we have

$$\begin{aligned}
ANI_e(\delta) &= \frac{E\left[Z_e^{(1)} + \cdots Z_e^{(t^*)}\right]}{D_e} && (6.56)\\
&= \frac{E\left[\sum_{\tau=1}^{t^*} \log f^{(\tau)}(s^{(\tau)})/g_e^{(\tau)}(s^{(\tau)})\right]}{D_e} && (6.57)\\
&= \frac{E\left[-\log \prod_{\tau=1}^{t^*} g_e^{(\tau)}(s^{(\tau)})/f^{(\tau)}(s^{(\tau)})\right]}{D_e} && (6.58)\\
&\geq \frac{-\log E\left[\prod_{\tau=1}^{t^*} g_e^{(\tau)}(s^{(\tau)})/f^{(\tau)}(s^{(\tau)})\right]}{D_e}. && (6.59)
\end{aligned}$$

The last inequality follows from Jensen's inequality [25].

Let $\boldsymbol{C^t}$ be a set of score sequences $\boldsymbol{s^t}$ such that the verification attempt results in *reject* (i.e. $d = 0$) at the t-th input in δ (in the same way as the proof of Theorem 6.2). As $\max(\alpha, \beta) \to 0$, the probability that the genuine attempts of $e \in E$ result in acceptance goes to 1. Then, the expectation in (6.59) is taken over the set $\bar{\boldsymbol{C^1}} \cup \cdots \cup \bar{\boldsymbol{C^T}}$ and can be written, using (6.7) and (6.17), as follows:

$$\begin{aligned}
& E\left[\prod_{\tau=1}^{t^*} g_e^{(\tau)}(s^{(\tau)})/f^{(\tau)}(s^{(\tau)})\right] && (6.60)\\
\to\ & \sum_{t=1}^{T} \int_{\bar{\boldsymbol{C}}^{\boldsymbol{t}}} \prod_{\tau=1}^{t} f^{(\tau)}(s^{(\tau)}) \cdot g_e^{(\tau)}(s^{(\tau)})/f^{(\tau)}(s^{(\tau)}) d\boldsymbol{s^t} && (6.61)\\
=\ & \sum_{t=1}^{T} \int_{\bar{\boldsymbol{C}}^{\boldsymbol{t}}} p(\boldsymbol{s^t}|W_{*,e}) d\boldsymbol{s^t} && (6.62)\\
=\ & FAP_{*,e}(\delta) && (6.63)\\
\leq\ & \alpha. && (6.64)
\end{aligned}$$

It follows from (6.59) and (6.64) that

$$\inf_{\delta \in \Delta(\alpha,\beta)} ANI_e(\delta) \geq \frac{\log \alpha^{-1}}{D_e} \quad as \quad \max(\alpha, \beta) \to 0. \tag{6.65}$$

□

Here we explain the meaning of this lemma. The KL divergence $D(f^{(t)}||g_e^{(t)})$ between $f^{(t)}$ and $g_e^{(t)}$ has a meaning of a distance measure between $f^{(t)}$ and $g_e^{(t)}$. If this takes the value D_e independently of the modality (i.e. if the condition (ii) holds), then ANI_e of any sequential fusion scheme is lower bounded by the right side of (6.49) in the asymptotic case where the identification error probabilities are sufficiently small (i.e. $\max(\alpha, \beta) \to 0$).

6.4.4 Proof of the Optimality with Regard to ANI

From Lemma 6.1, to prove that the MLRSV scheme is optimal with regard to ANI, it suffices to show that ANI_e of the MLRSV algorithm δ_0 can achieve the right side of (6.49). We prove that this is indeed the case:

Theorem 6.3. (Asymptotic optimality of the MLRSV scheme with regard to ANI) *If (i) the log-likelihood ratios $Z_v^{(t)}$ and $Z_e^{(t)}$ $(1 \leq t \leq T)$ are perfectly estimated in the MLRSV algorithm δ_0, and (ii) the KL divergence between $f^{(t)}$ and $g_e^{(t)}$ takes a value D_e independently of the modality, then*

$$ANI_e(\delta_0) \sim \frac{\log \alpha^{-1}}{D_e} \quad as \quad \max(\alpha, \beta) \to 0, \tag{6.66}$$

where $X \sim Y$ as $r \to 0$ means that $\lim_{r \to 0}(X/Y) = 1$ [30].

Proof. Suppose $e \in E$ attempts verification against him/herself in the MLRSV algorithm δ_0. Let t^* be a random variable which represents the number of biometric inputs required to terminate the verification process (in the same way as the proof of Lemma 6.1). Then, if (ii) holds, $ANI_e(\delta_0)$ can be expressed, in the same way as (6.56), as

$$ANI_e(\delta_0) = \frac{E\left[Z_e^{(1)} + \cdots Z_e^{(t^*)}\right]}{D_e}. \tag{6.67}$$

Since $e \in E$ attempts verification against him/herself, if (i) holds, then $g_v^{(t)}$ is equal to $g_e^{(t)}$, and hence $Z_{min}^{(t)} = Z_v^{(t)} = Z_e^{(t)}$. Thus, $ANI_e(\delta_0)$ in (6.67) can be further written as follows:

$$ANI_e(\delta_0) = \frac{E\left[Z_{min}^{(1)} + \cdots Z_{min}^{(t^*)}\right]}{D_e} \tag{6.68}$$

$$= \frac{E\left[Z_{tot}^{(t^*)}\right]}{D_e}. \tag{6.69}$$

As $\max(\alpha, \beta) \to 0$, the probability that the genuine attempts of $e \in E$ result in acceptance goes to 1. At the same time, since A $(= \log \alpha^{-1})$ goes to infinity while the expected gain of the minimum log-likelihood ratio $E[Z_{min}^{(t)}]$ $(= E[Z_e^{(t)}])$ at each input is fixed to D_e (see (6.53)), the excess of $Z_{tot}^{(t^*)}$ over the threshold A becomes negligible compared to A. Thus, we have

$$E\left[Z_{tot}^{(t^*)}\right] \sim A \tag{6.70}$$

$$= \log \alpha^{-1} \tag{6.71}$$

as $\max(\alpha, \beta) \to 0$. It follows from (6.69) and (6.71) that

$$ANI_e(\delta_0) \sim \frac{\log \alpha^{-1}}{D_e} \quad as \quad \max(\alpha, \beta) \to 0. \tag{6.72}$$

□

This theorem means that the MLRSV scheme achieves, for any $e \in E$, the minimum ANI_e (i.e. the right side of (6.49)) among all sequential fusion schemes in the asymptotic case where the identification error probabilities are sufficiently small (i.e. $\max(\alpha, \beta) \to 0$).

6.4.5 Limitations

We have so far clarified the theoretical properties of security and optimality of the MLRSV scheme. However, this scheme has some limitations. Firstly, we excluded adaptive spoofing wolves in Theorem 6.1. Suppose there is an adaptive spoofing wolf who can perfectly imitate the voice of others. If this wolf attempts an impostor attack against each enrollee $e \in E$ by presenting the voice which has high similarity score *only* against $e \in E$, there is no way to block such an attack and consequently WAP reaches 100[%]. We need to adopt a modality which is very difficult to spoof (e.g. iris and retina), to prevent such an attack.

Secondly, the condition (i), which is common to Theorem 6.1, 6.2, and 6.3, does not hold in general. That is, it is generally impossible to *perfectly* estimate the log-likelihood ratios $Z_v^{(t)} (= \log f^{(t)}/g_v^{(t)})$ and $Z_e^{(t)} (= \log f^{(t)}/g_e^{(t)})$. In our experiments in Section 6.6, we use the logistic regression model which directly estimates $\log f^{(t)}/g_v^{(t)}$ and $\log f^{(t)}/g_e^{(t)}$ as a linear function of a score (see (6.20) and (6.21)), and show that it works very well though not perfect.

Thirdly, Lemma 6.1 and Theorem 6.3 only guarantee the optimality of the MLRSV scheme with regard to ANI of $e \in E$ in the asymptotic setting where $FAP_{*,e}$ and FRP_e are sufficiently small (i.e. A and T are sufficiently large). It remains unsettled whether it is optimal in the case where $FAP_{*,e}$ and FRP_e are not small.

Last but not least, the condition (ii) in Lemma 6.1 and Theorem 6.3 also does not hold in general. As mentioned above, the KL divergence is a distance measure between two probability distributions, and some studies proposed to use the KL divergence between the genuine distribution and the impostor distribution as a metric of identification performance [105, 108]. Thus, it is natural to consider that the KL divergence $D(f^{(t)}||g_e^{(t)})$ differs from one modality to another, especially in the case of multiple biometric traits (e.g. fingerprint, face and iris) [96]. In Section 6.5, we propose an input order decision scheme using the KL divergence as an improvement of the MLRSV scheme in the case where (ii) does not hold.

6.5 Input Order Decision Scheme Using the Kullback-Leibler Divergence

6.5.1 Algorithm

Our second proposal is an input order decision scheme based on the KL divergence, which further reduces ANI of the MLRSV scheme in the case where the KL divergence differs from one modality to another (i.e. when the condition (ii) in Lemma 6.1 and Theorem 6.3 does not hold).

This scheme decides, for each enrollee $e \in E$, *the input order* using the KL divergence. Let $D_e^{(t)}$ the KL divergence between $f^{(t)}$ and $g_e^{(t)}$. After $e \in E$ enrolls his/her templates, this scheme decides the input order (the order of modalities) as follows:

1. Compute the KL divergence $D_e^{(t)}$ for each modality ($1 \leq t \leq T$);

2. Sort the modalities in descending order of $D_e^{(t)}$.

Then, when the claimant claims his/her identity as $e \in E$, this scheme requires or recommends him/her to sequentially input a query sample according to the above order. Alternatively, it can present the above order to $e \in E$ right after $e \in E$ enrolls his/her templates, because $f^{(t)}$ and $g_e^{(t)}$ can be estimated right after the enrollment (as described in Section 6.3.1), and so can the KL divergence $D_e^{(t)}$.

Suppose $e \in E$ attempts verification against him/herself by sequentially inputting his/her query sample according to the above order. The KL divergence $D_e^{(t)}$ can be written, using (6.19) and (6.50), as follows:

$$\begin{aligned} D_e^{(t)} &= D(f^{(t)}||g_e^{(t)}) && (6.73) \\ &= \int f^{(t)}(s^{(t)}) \log \frac{f^{(t)}(s^{(t)})}{g_e^{(t)}(s^{(t)})} ds^{(t)} && (6.74) \\ &= E\left[Z_e^{(t)}\right]. && (6.75) \end{aligned}$$

That is, $D_e^{(t)}$ can be regarded as the expectation of $Z_e^{(t)}$. Since $e \in E$ attempts verification against him/herself, $g_v^{(t)}$ is equal to $g_e^{(t)}$, and hence $D_e^{(t)}$ can be regarded as the expectation of $Z_{min}^{(t)}$ $(= Z_v^{(t)} = Z_e^{(t)})$. Thus, the expectation of $Z_{tot}^{(t)}$ is maximized at any number of biometric inputs t, by sorting the input order in descending order of $D_e^{(t)}$. As a result, it can be expected that $Z_{tot}^{(t)}$ exceeds A with the smaller number of biometric inputs, and ANI is further reduced.

Note that the above input order can be either a requirement or a recommendation to further reduce ANI. In the latter case, the system allows other input orders, which can be particularly helpful in the case where $e \in E$ cannot use some modalities at the verification time (due to injury, for example). Theorem 6.1 in Section 6.4.1 also guarantees the security against wolves and lambs, irrespective of the input order.

6.5.2 Estimation of the KL divergence

We also explain how to estimate the KL divergence $D_e^{(t)}$ in our input order decision scheme. Here it is important to note that we only have to compute the *order relation between the KL divergences*. In Section 6.3.1, we described that it is difficult to correctly estimate an enrollee-specific genuine distribution from a small number of training samples. In the same way, it is difficult to correctly estimate the KL divergence $D_e^{(t)}$ using a small number of genuine scores from $e \in E$.

However, since we use *not the KL divergences themselves but the order relation between them* to decide the input order, a little estimation error of the KL divergences does not matter. Furthermore, Poh *et al.* [91] showed that the average of the enrollee-specific genuine distribution can be more reliably estimated than the standard deviation. Taking these matters into account, we propose to estimate $D_e^{(t)}$ by taking the average of $Z_e^{(t)} = \log f^{(t)}(s^{(t)})/g_e^{(t)}(s^{(t)})$ over genuine scores from $e \in E$ (recall that $D_e^{(t)}$ is the expectation of $Z_e^{(t)}$ as shown in (6.75)), in the case where more than one biometric sample can be obtained from $e \in E$. By doing so, we can expect that the order relation between the KL divergences of $e \in E$ can be reliably estimated, while considering the recognizability of $e \in E$. In our experiments in Section 6.6, we demonstrate the effectiveness of this estimation method.

6.6 Experimental Evaluation

6.6.1 Experimental Set-up

As described in Section 6.2.1, zero-effort wolves and lambs cannot be blocked using anti-spoofing measures such as liveness detection. Thus, it is especially important to demonstrate the security of our schemes against these animals, using datasets whose subjects have no intent to spoof the system. To this end, we evaluated our two schemes using the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21]. We assumed that face and fingerprint biometrics are mutually independent, and created 500 *virtual* subjects who have one face and 8 fingerprints ($T = 9$), in a similar way to [39, 103]. We describe this in detail below.

Datasets

The NIST BSSR1 Set3 dataset contains face scores from 3000 subjects, each of whom contributed two query samples and one template ($3000 \times 3000 \times 2$ scores in total; there are no scores between templates). Although there are scores from two algorithms ("C" and "G") in this dataset, we used those from the algorithm "G".

The CASIA-FingerprintV5 contains 20000 fingerprint images (left and right thumb / index / middle / ring finger) from 500 subjects, each of whom contributed 5 samples per finger. We assumed, for each finger, the first sample as a template, the remaining four samples as query samples, and computed scores using SourceAFIS Version 1.4 [104], a freely available fingerprint matcher ($8 \times 500 \times 500 \times 5$ scores in total, including scores between templates). We then randomly selected 500 subjects from 3000 subjects in the face dataset, and created 500 virtual subjects who have one face and 8 fingerprints ($T = 9$). Here, we tried 100 ways to randomly select 500 subjects from 3000 subjects, and carried out, for each set of virtual subjects, the following experiment.

Experiment

From 500 virtual subjects, we randomly selected 300 subjects for evaluation (i.e. claimants or enrollees), and used the templates of the remaining 200 subjects as dummy-templates. We then carried out an experiment where each of 300 claimants attempts verification against each of 300 enrollees by sequentially inputting the last query sample of each modality (we used the remaining one face query sample and three fingerprint query samples to estimate the KL divergence, which is described later in details). The number of genuine attempts was 300, while the number of impostor attempts was 89700 ($= 300 \times 299$).

We evaluated FAR, WAR, LAR, FRR, and ANI in the above experiment, and averaged them over the 100 sets of virtual subjects to obtain stable performance.

Evaluated schemes

For comparison, we evaluated the following sequential fusion schemes:

- **LRSV:** the LRSV scheme [107]. This scheme computes a log-likelihood ratio $\log f^{(t)}/g^{(t)}$, where $g^{(t)}$ is an impostor distribution common to all users. We used the logistic regression model which estimates $\log f^{(t)}/g^{(t)} = {w_1}^{(t)} s^{(t)} + {w_0}^{(t)}$, where ${w_1}^{(t)}$ and ${w_0}^{(t)}$ are regression coefficients (we describe the training method later in detail). We randomly decided the input order for each $e \in E$.

- **MLRSV:** the MLRSV scheme. As described in Section 6.3.2, it computes the minimum of $\log f^{(t)}/g_e^{(t)}$ and $\log f^{(t)}/g_v^{(t)}$. We used the logistic regression model which estimates $\log f^{(t)}/g_e^{(t)} = w_{1e}^{(t)} s^{(t)} + w_{0e}^{(t)}$ and $\log f^{(t)}/g_v^{(t)} = w_{1v}^{(t)} s^{(t)} + w_{0v}^{(t)}$ (see (6.20) and (6.21)). The input order is the same as that of the LRSV scheme.
- **MLRSV-KL:** the MLRSV scheme using the input order decision scheme based on the KL divergence.

Training of the regression coefficients and the KL divergences

We trained each of the regression coefficients (${w_1}^{(t)}$, ${w_0}^{(t)}$, $w_{1e}^{(t)}$, $w_{0e}^{(t)}$, $w_{1v}^{(t)}$, and $w_{0v}^{(t)}$) using 200 genuine scores and 200 impostor scores. First, we trained ${w_1}^{(t)}$ and ${w_0}^{(t)}$ in the LRSV scheme using 200 genuine scores and 200 impostor scores between 200 dummy-templates and 200 corresponding first query samples (we randomly selected one impostor query sample for each dummy-template). Then, we trained $w_{1e}^{(t)}$ and $w_{0e}^{(t)}$ in the MLRSV scheme using the above 200 genuine scores and 200 impostor scores between dummy-templates and the template of $e \in E$ (here we substituted the first query samples for dummy-templates in the NIST BSSR1 Set3 because there were no scores between templates). Similarly, we trained $w_{1v}^{(t)}$ and $w_{0v}^{(t)}$ using the above 200 genuine scores and 200 impostor scores between the query sample of $v \in V$ and dummy-templates. As a training method, we adopted the Newton-Raphson method [12] which approximates the maximum likelihood estimates since there were a number of (hundreds of) training samples.

As for the KL divergence $D_e^{(t)}$, we used a small number of genuine scores from $e \in E$ as described in Section 6.5.2: three scores for a fingerprint and one score for a face. First, we computed, for each fingerprint, three genuine scores between the remaining three query samples (other than the last one query sample which was used for evaluation) and the template. Then, we estimated the KL divergence $D_e^{(t)}$ by taking the average of the estimated log-likelihood ratios $\log f^{(t)}/g_e^{(t)}$ ($= w_{1e}^{(t)} s^{(t)} + w_{0e}^{(t)}$) over the three scores. In the same way, we computed, for each face, one genuine score between the remaining one query sample and the template, and estimated the KL divergence by computing the corresponding log-likelihood ratio $\log f^{(t)}/g_e^{(t)}$.

6.6.2 Experimental Results

Figure 6.3 shows the relationship between WAR/LAR and a required WAR/LAR (i.e. $\alpha = e^{-A}$) in our proposals. It was found that the MLRSV scheme kept WAR and LAR less than the required value, irrespective of the input order, as described in Theorem 6.1 in Section 6.4.1. We consider this is because the logistic regression parameters were correctly estimated. We also show in Figure 6.4 the logarithm of the ratio between the frequency distribution of genuine scores and that of impostor scores (left: face, right: left thumb), obtained using all the scores between query samples and templates (left: $3000 \times 3000 \times 2$ scores, right: $500 \times 500 \times 4$ scores). There is a close-to-linear relationship between the log-likelihood ratio and the score, especially in the case of left thumbs. We also confirmed that a similar tendency was obtained for the other types of fingers.

Figure 6.5 shows the trade-off between FRR and FAR in the three schemes. It was found that the MLRSV scheme outperformed the LRSV scheme. We consider this is because the MLRSV scheme can provide an optimal trade-off between FRP_e and FAP_e for any $e \in E$

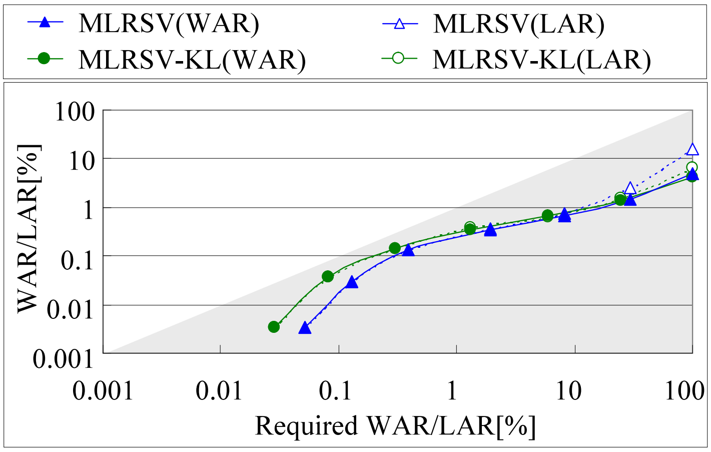


Figure 6.3: Relationship between WAR/LAR and a required WAR/LAR (i.e. $\alpha = e^{-A}$) in our proposals. WAR/LAR is less than α in the gray area.

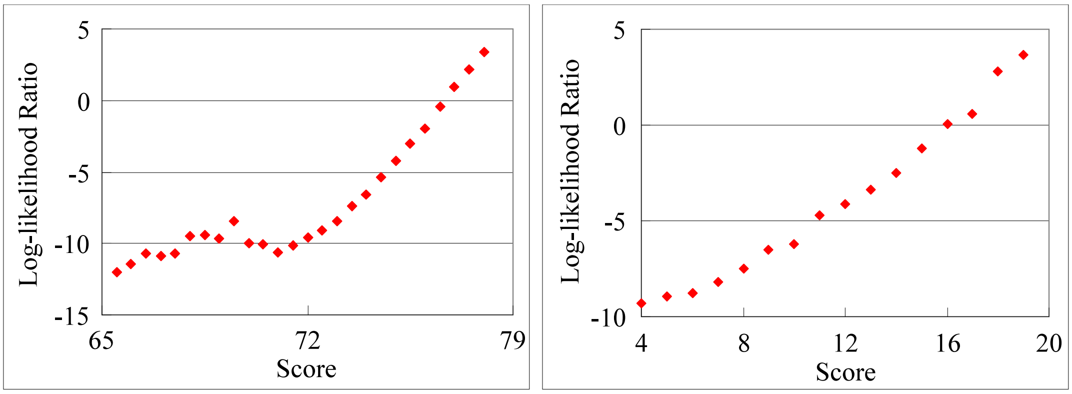


Figure 6.4: Logarithm of the ratio between the frequency distribution of genuine scores and that of impostor scores (left: face, right: left thumb). The width of the interval is 0.5 and 1.0, respectively. We do not show both side edges where the number of genuine scores or impostor scores is less than 10.

if the input order is the same (see Theorem 6.2 in Section 6.4.2). It was also found from Figure 6.5 that the trade-off between FRR and FAR was further improved by changing the input order using our input order decision scheme. We consider this is because ANI was significantly improved by using this scheme, as shown in the following.

Figure 6.6 shows the trade-off between FAR/WAR/LAR and ANI in the three schemes. It was found that the trade-off of the MLRSV scheme was better than that of the LRSV scheme and was significantly improved by using the input order decision scheme. We examined, for each enrollee and each modality, the estimated value of the KL divergence, and confirmed that the value differed from one modality to another (i.e. the condition (ii) did not hold). Thus, we consider the reason the input order decision scheme worked very well is that the order relation of the KL divergences of $e \in E$ was reliably estimated using genuine scores from $e \in E$, and the expectation of the likelihood ratio $Z_{tot}^{(t)}$ was maximized using the order relation, as described in Section 6.5. For example, when the maximum of WAR and LAR was fixed to 0.01%, ANI of the LRSV scheme, the MLRSV scheme, and the MLRSV scheme

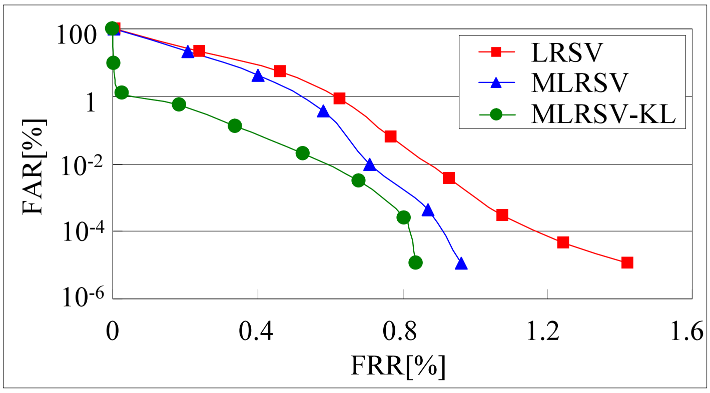


Figure 6.5: Trade-off between FRR and FAR in the three schemes.

Table 6.2: Proportion of each modality for each rank decided by the input order decision scheme [%].

	1st	2nd	3rd	4th	5th	6th	7th	8th	9th
Face	0.25	0.40	1.5	2.5	4.3	6.1	11.3	22.5	51.3
Left thumb	14.4	13.0	12.9	10.9	12.1	10.2	10.5	9.1	7.0
Left index	13.2	11.9	12.4	11.3	12.3	11.4	12.8	9.9	4.8
Left middle	18.6	15.4	13.9	12.9	10.3	9.9	10.1	6.2	2.8
Left ring	11.6	10.6	10.7	11.4	11.6	13.0	9.1	11.2	10.7
Right thumb	11.5	11.9	10.9	12.1	11.5	11.6	11.6	11.8	7.1
Right index	10.5	12.6	13.3	14.4	12.1	11.8	10.7	9.1	5.5
Right middle	14.8	14.5	13.7	12.4	12.6	11.6	9.8	7.1	3.5
Right ring	5.1	9.7	10.9	12.1	13.2	14.4	14.2	12.9	7.4

using the input order decision scheme were 1.94, 1.72, and 1.19, respectively.

To more thoroughly investigate the effectiveness of the input order decision scheme, we finally examined, for each rank (1st, 2nd, $\cdots$, 9th) decided by the input order decision scheme, the proportion of each modality. Table 6.2 shows the results. It can be seen that a face was ranked very low in most cases, while the 1st modality was a fingerprint in most cases. This indicates that the KL divergence significantly differs from one biometric trait (e.g. face, fingerprint, iris) to another. Thus, in the case of multiple biometric traits, the optimal input order can be common to most enrollees, and finding the order can be a trivial problem (i.e. we only have to start from the accurate biometric trait). On the other hand, it can be seen from Table 6.2 that the type of the most discriminative finger differed from one enrollee to another, which indicates finding the optimal input order in the case of multiple instances (e.g. multiple fingerprints, multiple finger-veins) is *not* trivial. Even in such a case, the input order decision scheme can provide an optimal input order for each enrollee, and significantly reduce ANI, as shown in Figure 6.6. Thus, we can conclude that this scheme is highly effective not for multiple biometric traits but for multiple instances.

6.7 Conclusions

In this chapter, we first introduced a taxonomy of wolves and lambs, defined security measures for the animals, and proposed the MLRSV scheme as a countermeasure against the animals.

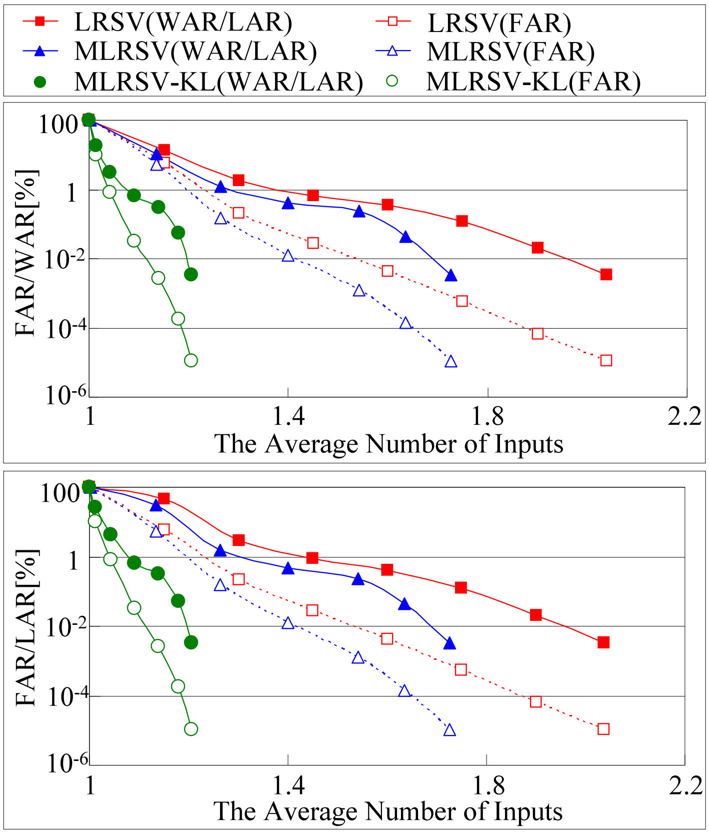


Figure 6.6: Trade-off between FAR/WAR/LAR and ANI in the three schemes.

We proved its security against the animals except for adaptive spoofing wolves and optimality with regard to FRP under the condition (i) (Theorem 1 and 2). We also proved its asymptotic optimality with regard to ANI under the condition (i) and (ii) (Lemma 1 and Theorem 3). We finally proposed the input order decision scheme based on the KL divergence to further reduce ANI in the case where the condition (ii) does not hold.

The experimental results demonstrated the security of our schemes against zero-effort wolves and lambs who cannot be blocked using anti-spoofing measures such as liveness detection. On the other hand, our schemes are not secure against adaptive spoofing wolves as discussed in Section 6.4.5, while anti-spoofing measures can block spoofing wolves and lambs to some extent. Thus, it is desirable to use our schemes in conjunction with anti-spoofing measures to have total security against wolves and lambs.

It should also be noted that although we focused on verification in this chapter, wolves and lambs cause more serious security problems in identification. The MLRSV scheme is a modification of the LRSV sequential fusion scheme to have security against wolves and lambs by computing the minimum of two log-likelihood ratios obtained using two kinds of user-specific impostor distributions. Since the LRSI scheme proposed in Section 3.4.3 is the extension of the LRSV sequential fusion scheme to the identification mode, we consider we can construct a sequential fusion scheme in identification which has security against wolves and lambs by modifying the LRSI scheme to compute the minimum of two log-likelihood ratios using two kinds of user-specific impostor distributions. We shall refer to such a scheme as the *MLRSI (Minimum Likelihood Ratio-based Sequential Identification) scheme.* As future work, we plan to define security measures for wolves and lambs in identification, and prove the security and optimality of the MLRSI scheme.

Chapter 7

Conclusion

Contents

In this dissertation, we studied on optimization of security and convenience in biometric identification. We included false accepts, wolves, and lambs as factors which affect security, and false rejects, the number of inputs, and response time as factors which affect convenience. We mainly addressed the following three challenges: (1) optimization of identification errors (false accepts and false rejects) and the number of inputs; (2) optimization of response time along with the above factors; (3) optimization of security against wolves and lambs and convenience in terms of the number of inputs and false rejects.

In the following, we summarize the contributions of this dissertation, and conclude this dissertation with directions for the future.

7.1 Summary of Contributions

7.1.1 Optimization of Identification Errors and the Number of Inputs

In Chapter 3, we focused on MSPRT's (Multi-hypothesis Sequential Probability Ratio Tests) [30], and proposed two sequential fusion schemes in identification: the PPSI (Posterior Probability-based Sequential Identification) scheme and the LRSI (Likelihood Ratio-based Sequential Identification) scheme. The PPSI scheme is based on MSPRT (Test δ_a) and can optimize the trade-off between the identification error probabilities (EFRP/EFAP/NFAP) and ANI (the average number of inputs), while the LRSI scheme is a simpler one which can be carried out quickly. Then we proved that the LRSI scheme can also optimize the above trade-off by showing some properties of N-dimensional score distributions (such as "Pythagorean theorem for N-dimensional score distributions"), and proving that *this scheme is equivalent to MSPRT (Test δ_b)* based on the properties. We also discussed the conditions to achieve the optimality of the two schemes. We finally evaluated our two schemes using the

NIST BSSR1 Set1 dataset [77] (one face and two fingerprints), and showed the effectiveness of our schemes through comparison with the OR rule [14] and the posterior probability-based parallel fusion scheme in identification [76].

7.1.2 Optimization of Response Time

In Chapter 4, we proposed the PPS (Posterior Probability-based Search) scheme which normalizes pseudo-scores to the posterior probabilities of being in the answer to the range query, and uses them as probability-based pseudo-scores to search non-pivots. We proposed an algorithm which computes the probability-based pseudo-scores using the object-specific parameters in logistic regression, and learns the parameters using MAP (Maximum a Posteriori) estimation. We also proposed a technique which speeds up learning the parameters using pseudo-scores. We proved that the PPS scheme has an optimal property with regard to the number of score computations and the expected number of retrieval errors, and showed that it outperforms the standard pivot-based indexing scheme [22] and the permutation-based indexing scheme [4, 22] with regard to both the number of score computations and the CPU time in various datasets from the Metric Space Library [37].

In Chapter 5, we proposed a sequential indexing and fusion framework in biometric identification which is constructed from (I) a pseudo-score based indexing scheme, (II) a sequential search scheme, and (III) a sequential fusion scheme in identification which handles missing scores. Then we proposed the PPSS (Posterior Probability-based Sequential Search) scheme, a modification of the PPS scheme to compute posterior probabilities using *not only pseudo-scores at the current input but past pseudo-scores and scores*, as (II). We discussed the optimal property of this scheme with regard to the trade-off between the number of score computations and the retrieval error probability, and described that it does not have a problem of the learning time. We also proposed a technique which optimizes the number of pivots with regard to the retrieval error probability. We finally evaluated our proposals using a large-scale multi-modal dataset ($N = 1800$ enrollees; one face and two fingerprints) obtained by combining the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21], and showed that our proposals reduce the number of score computations to 10[%] while keeping the identification error rates (EFRR/EFAR/NFAR) and ANI of the PPSI scheme.

7.1.3 Optimization of Security against Wolves and Lambs and Convenience

In Chapter 6, we introduced a taxonomy which classifies wolves into three categories (zero-effort wolves, non-adaptive spoofing wolves, and adaptive spoofing wolves) and lambs into two categories (zero-effort lambs and spoofing lambs), and defined LAP (Lamb Accept Probability) as a security measure for lambs in the same way as WAP (Wolf Attack Probability) [116]. Then we proposed the MLRSV (Minimum Likelihood Ratio-based Sequential Verification) scheme as a sequential fusion scheme in verification which has an optimal property with regard to security against wolves and lambs and convenience. We proved that this scheme can keep WAP and LAP less than a desired value except in the case of adaptive spoofing wolves, and minimize FRP (False Reject Probability) and ANI. We also discussed the conditions to achieve the security and optimality. We then proposed an input order decision scheme based on the KL (Kullback-Leibler) divergence to further reduce ANI of the MLRSV scheme in the case where the KL divergence differs from one modality to another. We finally evaluated our schemes using a multi-modal dataset (one face and eight fingerprints) obtained by combining the NIST BSSR1 Set3 dataset [77] and the CASIA-FingerprintV5 dataset [21], and showed the effectiveness of our schemes through comparison with the LRSV scheme [107].

7.2 Future Work

In this dissertation, we included false accepts, wolves, and lambs as factors which affect security, and false rejects, the number of inputs, and response time as factors which affect convenience in biometric identification. There are still some challenges which need to be addressed to optimize all of these factors:

Firstly, we proposed a countermeasure against wolves and lambs only in the verification mode in Chapter 6. As described at the end of Chapter 6, one of our major future work is to define security measures for wolves and lambs in identification, and extend our countermeasure to the identification scenario. Secondly, although we proved a number of optimal properties of our proposals with regard to security and convenience (Theorem 3.1, Proposition 4.1, Proposition 5.1, Theorem 6.1, 6.2, 6.3), the conditions to achieve the optimality are not satisfied in reality, as discussed in Section 3.5.3 and 6.4.5. The proof of the optimality without the conditions, or developing such algorithms is another challenging future work.

In Chapter 1, we described that a spoofing attack and a leakage of biometric information are also major security problems which are common to verification and identification. One of the advantages of our proposals is that they can be used in conjunction with a template protection scheme which outputs scores [93, 106, 111, 112], as described in Section 2.4. Our proposals can also be combined with anti-spoofing measures such as liveness detection. In Section 6.7, we also described that this combination is important to increase total security against wolves and lambs. It is desirable to use our proposals, anti-spoofing measures, and template protection to have total security in biometric identification.

Recall that we need to design a system which is simultaneously secure and convenient to overcome the *security vs. convenience dilemma*, as described in Chapter 1. Biometric identification, which does not require a user ID, password, nor card but recognizes a user based on “something you are” (an inherence factor), has a potential to provide the best solution with regard to security and convenience. We believe this dissertation, which studied on optimization of security and convenience in biometric identification, plays a significant role in achieving this ultimate goal.

Bibliography

[1] Anne Adams and Martina Angele Sasse. Users are not the enemy. *Communications of the ACM*, 42(12):40–46, 1999.

[2] Lorene Allano, Bemadette Dorizzi, and Sonia Garcia-Salicetti. Tuning cost performance in multi-biometric systems: A novel and consistent view of fusion strategies based on the sequential probability ratio test (SPRT). *Pattern Recognition Letters*, 31(9):884–890, 2010.

[3] Shunichi Amari and Andrzej Cichocki. Information geometry of divergence functions. *Bulletin of the Polish Academy of Sciences, Technical Sciences*, 58(1):183–195, 2010.

[4] Giuseppe Amato and Pasquale Savino. Approximate similarity search in metric spaces using inverted files. In *Proceedings of the 3rd international conference on Scalable information systems (InfoScale'08)*, pages 1–10, 2008.

[5] Iván Arce. The weakest link revisited. *IEEE Security & Privacy*, 1(2):72–76, 2003.

[6] Vassilis Athitsos, Jonathan Alon, Stan Sclaroff, and George Kollios. Boostmap: An embedding method for efficient nearest neighbor retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(1):89–104, 2008.

[7] Roland Auckenthaler, Michael Carey, and Harvey Lloyd-Thomas. Score normalization for text-independent speaker verification systems. *Digital Signal Processing (DSP)*, 10(1–3):42–54, 2000.

[8] Enrique Bailly-baillire, Samy Bengio, Frederic Bimbot, Miroslav Hamouz, Josef Kittler, Johnny Mariéthoz, Jiri Matas, Kieron Messer, Fabienne Porée, Belen Ruiz, and Jean-Philippe Thiran. The BANCA database and evaluation protocol. In *Proceedings of 4th International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'03)*, volume 2688 of *Lecture Notes in Computer Science*, pages 625–638, 2013.

[9] Dirk Balfanz, Glenn Durfee, D.K. Smetters, and Rebecca E. Grinter. In search of usable security: Five lessons from the field. *IEEE Security & Privacy*, 2(5):19–24, 2004.

[10] Glenn Becker and Mark Potts. Non-metric biometric clustering. In *Proceedings of Biometrics Symposium*, pages 1–6, 2007.

[11] Souheil Ben-Yacoub, Yousri Abdeljaoued, and Eddy Mayoraz. Fusion of face and speech data for person identity verification. *IEEE Transactions on Neural Networks*, 10(5):1065–1074, 1999.

[12] Christopher. M. Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

[13] David Blackwell. On an equation of Wald. *The Annals of Mathematical Statistics*, 17:84–87, 1946.

[14] Ruud M. Bolle, Jonathan H. Connell, Sharath Pankanti, Nalini K. Ratha, and Andrew W. Senior. *Guide to Biometrics.* Springer, 2003.

[15] Sergey Brin. Near neighbor search in large metric spaces. In *Proceedings of the 21st Conference on Very Large Databases (VLDB'95)*, pages 574–584, 1995.

[16] W.A. Burkhard and R.M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973.

[17] Benjamin Bustos and Gonzalo Navarro. Probabilistic proximity searching algorithms based on compact partitions. *Journal of Discrete Algorithms*, 2(1):115–134, 2004.

[18] Brent Cantafio. Security vs. convenience. Is RSA SecurID the answer? *GIAC Security Essentials Certification, Practical Assignment, Version 1.4b*, 2004.

[19] Raffaele Cappelli, Matteo Ferrara, and Davide Maltoni. Fingerprint indexing based on minutia cylinder-code. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):1051–1057, 2011.

[20] Raffaele Cappelli, Alessandra Lumini, Dario Maio, and Davide Maltoni. Fingerprint classification by directional image partitioning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):402–420, 1999.

[21] CASIA-FingerprintV5. `http://biometrics.idealtest.org/`.

[22] Edgar Chávez, Karina Figueroa, and Gonzalo Navarro. Effective proximity retrieval by ordering permutations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(9):1647–1658, 2008.

[23] Edgar Chávez, Gonzalo Navarro, Ricardo Baeza-Yates, and José L. Marroquín. Searching in metric spaces. *ACM Computing Surveys*, 33(3):273–321, 2001.

[24] Paolo Ciaccia and Marco Patella. PAC nearest neighbor queries: Approximate and controlled search in high-dimensional and metric spaces. In *Proceedings of the 16th International Conference on Data Engineering (ICDE'00)*, pages 244–255, 2000.

[25] Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory, Second Edition.* Wiley-Interscience, 2006.

[26] Simona G. Crihalmeanu, Arun Ross, Stephanie Schuckers, and Lawrence A. Hornak. A protocol for multibiometric data acquisition, storage and dissemination. Technical report, West Virginia University, LDCSEE, 2007.

[27] John Daugman. Probing the uniqueness and randomness of iriscodes: Results from 200 bilion iris pair comparisons. *Proceedings of the IEEE*, 94(11):1927–1935, 2006.

[28] Yaohui Ding and Arun Ross. A comparison of imputation methods for handling missing scores in biometric fusion. *Pattern Recognition*, 45:919–933, 2012.

[29] George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. In *Proceedings of the Fifth International Conference on Spoken Language Processing (ICSLP'98)*, pages 1351–1354, 1998.

[30] Vladimir P. Dragalin, Alexander G. Tartakovsky, and Venugopal V. Veeravalli. Multihypothesis sequential probability ratio tests, Part I: Asymptotic optimality. *IEEE Transactions on Information Theory*, 45(7):2448–2461, 1999.

[31] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience, 2000.

[32] Ole Edsberg and Magnus Lie Hetland. Indexing inexact proximity search with distance regression in pivot space. In *Proceedings of the Third International Workshop on Similarity Search and Applications (SISAP'10)*, pages 51–58, 2010.

[33] Andrea Esuli. PP-index: Using permutation prefixes for efficient and scalable approximate similarity search. In *Proceedings of the Seventh Workshop on Large-Scale Distributed Systems for Information Retrieval (LSDS-IR'09)*, pages 17–24, 2009.

[34] Omolara Fatukasi, Josef Kittler, and Norman Poh. Estimation of missing values in multimodal biometric fusion. In *Proceedings of the 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS'08)*, pages 1–6, 2008.

[35] Karina Figueroa, Edgar Chávez, Gonzalo Navarro, and Rodrigo Paredes. On the least cost for proximity searching in metric spaces. In *Proceedings of the Fifth Workshop on Efficient and Experimental Algorithms (WEA'06)*, pages 279–290, 2006.

[36] Karina Figueroa and Kimmo Fredriksson. Speeding up permutation based indexing with indexing. In *Proceedings of the Second International Workshop on Similarity Search and Applications (SISAP'09)*, pages 107–114, 2009.

[37] Karina Figueroa, Gonzalo Navarro, and Edgar Chávez. Metric spaces library. *http://www.sisap.org/Metric_Space_ Library.html*, 2007.

[38] Francis Galton. *Finger Prints*. London: Macmillan, 1892.

[39] Aglika Gyaourova and Arun Ross. Index codes for multibiometric pattern retrieval. *IEEE Transactions on Information Forensics and Security*, 7(2):518–529, 2012.

[40] Donna Harman. Overview of the third text retrieval conference. In *Proceedings of the third text retrieval conference (TREC-3)*, 1995.

[41] Edward R. Henry. *Classification and Uses of Finger Prints*. London: Routledge, 1900.

[42] C. A. R. Hoare. Algorithm65: Find. *Communications of the ACM*, 4(7):321–322, 1961.

[43] Lin Hong and Anil Jain. Integrating faces and fingerprints for personal identification. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, 20(12):1295–1307, 1998.

[44] Michael E. Houle and Jun Sakuma. Fast approximate similarity search in extremely high-dimensional data sets. In *Proceedings of the 21st International Conference on Data Engineering (ICDE'05)*, pages 619–630, 2005.

[45] Ogechukwu Iloanusi, Aglika Gyaourova, and Arun Ross. Indexing fingerprints using minutiae quadruplets. In *Proceedings of 2011 Conference on Computer Vision and Pattern Recognition Workshops (CVPRW'11)*, pages 127–133, 2011.

[46] International Telecommunication Union. The world in 2013: ICT facts and figures, 2013.

[47] Manabu Inuma, Akira Otsuka, and Hideki Imai. Theoretical framework for constructing matching algorithms in biometric authentication systems. In *Proceedings of the 3rd International Conference on Biometrics (ICB'09)*, pages 806–815, 2009.

[48] ISO/IEC 19784. Information technology - Biometric application programming interface - Part 1: BioAPI specification, 2006.

[49] ISO/IEC 19792. Information technology - Security techniques - Security evaluation of biometrics, 2009.

[50] ISO/IEC 19795-1. Information technology - Biometric performance testing and reporting - Part 1: Principles and framework, 2006.

[51] M. Hamed Izadi, Leila Mirmohamadsadeghi, and andrzej Drygajlo. Introduction of cylinder quality measure into minutia cylinder-code based fingerprint matching. In *Proceedings of the Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS'12)*, pages 353–358, 2012.

[52] Anil K. Jain, Lin Hong, and Ruud Bolle. On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4):302–314, 1997.

[53] Anil K. Jain and Stan Z. Li. *Encyclopedia of Biometrics*. Springer-Verlag, 2009.

[54] Anil K. Jain, Karthik Nandakumar, and Abhishek Nagar. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, pages 1–17, 2008.

[55] Anil K. Jain, Karthik Nandakumar, and Arun Ross. Score normalization in multimodal biometric systems. *Pattern Recognition*, 38(12):2270–2285, 2005.

[56] Anil K. Jain and Arun Ross. Fingerprint mosaicking. In *Proceedings of 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'02)*, pages 4064–4067, 2002.

[57] Anil K. Jain, Arun Ross, and Karthik Nandakumar. *Introduction to Biometrics*, volume 4. Springer, 2011.

[58] Anil K. Jain, Arun Ross, and Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.

[59] Jiming Jiang. *Large Sample Techniques for Statistics*. Springer, 2010.

[60] Xudong Jiang and Wee Ser. Online fingerprint template improvement. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1121–1126, 2002.

[61] Yoshihiro Kojima, Rie Shigetomi, Manabu Inuma, Akira Otsuka, and Hideki Imai. A matching algorithm secure against the wolf attack in biometric authentication systems. In *Proceedings of the Joint COST 2101 & 2102 International Conference on Biometric ID Management and Multimodal Communication (BioID_MultiComm'09)*, pages 293–300, 2009.

[62] Ajay Kumar, David C. M. Wong, Helen C. Shen, and Anil K. Jain. Personal verification using palmprint and hand geometry biometric. In *Proceedings of the Fourth International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'03)*, pages 668–678, 2003.

[63] Xuefeng Liang, Arijit Bishunu, and Tetsuo Asano. A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles. *IEEE Transactions on Information Forensics and Security*, 2(4):721–733, 2007.

[64] Yan Ma, Bojan Cukic, and Harshinder Singh. A classification approach to multibiometric score fusion. In *Proceedings of the Fifth International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'05)*, pages 484–493, 2005.

[65] Takuji Maeda, Masahito Matsushita, and Koichi Sasakawa. Identification algorithm using a matching score matrix. *IEICE Transactions on Information and Systems*, E84-D(7):819–824, 2001.

[66] Dario Maio, Davide Maltoni, Raffaele Cappelli, Jim L. Wayman, and Anil K. Jain. FVC2004: Third fingerprint verification competition. In *Proceedings of the First International Conference on Biometric Authentication (ICBA'04)*, pages 1–7, 2004.

[67] Davide Maltoni, Dario Maio, Anil K. Jain, and Salil Prabhakar. Fingerprint classification and indexing. In *Handbook of Fingerprint Recognition*, chapter 5, pages 235–269. Springer, second edition, 2009.

[68] Gian Luca Marcialis and Fabio Roli. Serial fusion of fingerprint and face matchers. In *Proceedings of the Seventh International Workshop on Multiple Classifiers Systems (MCS'07)*, volume 4472 of *Lecture Notes in Computer Science*, pages 151–160, 2007.

[69] Markets and Markets. Next generation biometric technologies market global forecast & analysis (2012–2017), 2012.

[70] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of artificial "gummy" fingers on fingerprint systems. In *Proceedings of SPIE*, volume 4677, pages 275–289, 2002.

[71] Theodore.K. Matthes. On the optimality of sequential probability ratio tests. *The Annals of Mathematical Statistics*, 34(1):18–21, 1963.

[72] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre. XM2VTSDB: The extended M2VTS database. In *Proceedings of the 2nd International Conference on Audio- and Video-based Biometric Person Authentication (AVBPA'99)*, pages 72–77, 1999.

[73] Luisa Micó, José Oncina, and Enrique Vidal. An algorithm for finding nearest neighbours in constant average time with a linear space complexity. In *Proceedings of the 11th International Conference on Pattern Recognition (ICPR'92)*, volume 2, pages 557–560, 1992.

[74] Kenneth R. Moses, Peter Higgins, Michael McCabe, Salil Probhakar, and Scott Swann. Fingerprint sourcebook. In *Automated Fingerprint Identification System (AFIS)*, chapter 6, pages 1–33. National Institute of Justice, 2010.

[75] Karthik Nandakumar, Yi Chen, Sarat C. Dass, and Anil K. Jain. Likelihood ratio-based biometric score fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):342–347, 2008.

[76] Karthik Nandakumar, Anil K. Jain, and Arun Ross. Fusion in multibiometric identification systems: What about the missing data? In *Proceedings of the Third International Conference on Advances in Biometrics (ICB'09)*, pages 743–752, 2009.

[77] NIST Biometric Scores Set - Release 1 (BSSR1). `http://www.nist.gov/itl/iad/ig/biometricscores.cfm`.

[78] Hideki Noda and Eiji Kawaguchi. Adaptive speaker identification using sequential probability ratio test. In *Proceedings of the 15th International Conference on Pattern Recognition (ICPR'00)*, pages 262–265, 2000.

[79] David Novak and Michal Batko. Metric index: An efficient and scalable solution for similarity search. In *Proceedings of the Second International Workshop on Similarity Search and Applications (SISAP'09)*, pages 65–73, 2009.

[80] P. Wessa, Free Statistics Software, Office for Research Development and Education, version 1.1.23-r7, 2012. `http://www.wessa.net/`.

[81] Jeffrey Paone and Patrick J. Flynn. On the consistency of the biometric menagerie for irises and iris matchers. In *Proceedings of the 2011 IEEE International Workshop on Information Forensics and Security (WIFS'11)*, pages 1–6, 2011.

[82] Marco Patella and Paolo Ciaccia. Approximate similarity search: A multi-faceted problem. *Journal of Discrete Algorithms*, 7(1):36–48, 2009.

[83] Bryan D. Payne and W. Keith Edwards. A brief introduction to usable security. *IEEE Internet Computing*, 12(3):13 – 21, 2008.

[84] P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, and Patrick J. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):1090–1104, 2000.

[85] Norman Poh. User-specific score normalization and fusion for biometric person recognition. In *Advanced Topics in Biometrics*, chapter 16, pages 401–418. World Scientific Publishing Company, 2010.

[86] Norman Poh, Thirimachos Bourlai, and Josef Kittler. A multimodal biometric test bed for quality-dependent, cost-sensitive and client-specific score-level fusion algorithms. *Pattern Recognition Journal*, 43(3):1094–1105, 2010.

[87] Norman Poh, Thirimachos Bourlai, Josef Kittler, Lorene Allano, Fernando Alonso-Fernandez, Onkar Ambekar, John Baker, Bernadette Dorizzi, Omolara Fatukasi, Julian Fierrez, Harald Ganster, Javier Ortega-Garcia, Donald Maurer, Albert Ali Salah, Tobias Scheidat, and Claus Vielhauer. Benchmarking quality-dependent and cost-sensitive score-level multimodal biometric fusion algorithms. *IEEE Transactions on Information Forensics and Security*, 4(4):849–866, 2009.

[88] Norman Poh and Josef Kittler. A methodology for separating sheep from goats for controlled enrollment and multimodal fusion. *Proceedings of Biometrics Symposium 2008 (BSYM'08)*, pages 17–22, 2008.

[89] Norman Poh and Josef Kittler. A biometric menagerie index for characterising template/model-specific variation. *Proceedings of the Third International Conference on Advances in Biometrics (ICB'09)*, pages 816–827, 2009.

[90] Norman Poh, Ajita Rattani, and Rabio Roli. Critical analysis of adaptive biometric systems. *IET Biometrics*, 1(4):179–187, 2013.

[91] Norman Poh, Arun Ross, Weifeng Lee, and Josef Kittler. A user-specific and selective multimodal biometric fusion strategy by ranking subjects. *Pattern Recognition*, 46(12):3341–3357, 2013.

[92] Norman Poh, David Windridge, Vadim Mottl, Alexander Tatarchuk, and Andrey Eliseyev. Addressing missing values in kernel-based multimodal biometric fusion using neutral point substitution. *IEEE Transactions on Information Forensics and Security*, 5(3):461–469, 2010.

[93] Nalini K. Ratha, Sharat Chikkerur, Jonathan H. Connell, and Ruud M. Bolle. Generating cancelable fingerprint templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):561–572, 2007.

[94] Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle. Image mosaicing for rolled fingerprint construction. In *Proceedings of the Fourteenth International Conference on Pattern Recognition (ICPR'98)*, pages 1651–1653, 1998.

[95] Arun Ross and Rohin Govindarajan. Feature level fusion using hand and face biometrics. In *Proceedings of the SPIE Conference on Biometric Technology for Human Identification*, pages 196–204, 2005.

[96] Arun Ross and Anil K. Jain. Multimodal biometrics: An overview. In *Proceedings of the 12th European Signal Pocessing Conference (EUSIPCO'04)*, pages 1221–1224, 2004.

[97] Arun Ross, Karthik Nandakumar, and Anil K. Jain. *Handbook of Multibiometrics*. Springer, 2006.

[98] Arun Ross, Ajita Rattani, and Massimo Tistarelli. Exploiting the "Doddington zoo" effect in biometric fusion. In *Proceedings of the 3rd IEEE International Conference on Biometrics: Theory, Applications, and Systems (BTAS'09)*, pages 264–270, 2009.

[99] Martina Angela Sasse and Ivan Flechais. Usable security: Why do we need it? how do we get it? In *Security and Usability: Designing Secure Systems That People Can Use*, chapter 2, pages 13–30. O'Reilly Media, 2005.

[100] S.A.C. Schuckers. Spoofing and anti-spoofing measures. *Information Security Technical Report*, 7(4):56–62, 2002.

[101] Xin Shuai, Chao Zhang, and Pengwei Hao. Fingerprint indexing based on composite set of reduced sift features. In *Proceedings of the 19th International Conference on Pattern Recognition (ICPR'08)*, 2008.

[102] Matthew Skala. Counting distance permutations. In *Proceedings of the First International Workshop on Similarity Search and Applications (SISAP'08)*, pages 69–76, 2008.

[103] Robert Snelick, Umut Uludag, and Alan Mink. Large-scale evaluation of multimodal biometric authentication using state-of-the-art systems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(3):450–455, 2005.

[104] SourceAFIS. `http://sourceforge.net/projects/sourceafis/`.

[105] Yagiz Sutcu, Husrev T. Sencar, and Nasir Memon. How to measure biometric information? In *Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR'10)*, pages 1469–1472, 2010.

[106] Kenta Takahashi and Shinji Hirata. Generating provably secure cancelable fingerprint templates based on correlation-invariant random filtering. In *Proceedings of the 3rd International Conference on Biometrics: Theory, Applications, and Systems (BTAS'09)*, pages 1–6, 2009.

[107] Kenta Takahashi, Masahiro Mimura, Yoshiaki Isobe, and Yoichi Seto. A secure and user-friendly multimodal biometric system. In *Proceedings of SPIE*, volume 5404, pages 12–19, 2004.

[108] Kenta Takahashi and Takao Murakami. A metric of information gained through biometric systems. In *Proceedings of the 2010 20th International Conference on Pattern Recognition (ICPR'10)*, pages 1184–1187, 2010.

[109] Xuejun Tan, Bir Bhanu, and YingQiang Linl. Fingerprint identification: Classification vs. indexing. In *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS'03)*, pages 151–156, 2003.

[110] Mohammad Nayeem Teli, J. Ross Beveridge, P. Jonathon Phillips, and Geof H. Givens. Biometric zoos: Theory and experimental evidence. In *Proceedings of the IEEE/IAPR International Joint Conference on Biometrics (IJCB'11)*, pages 1–8, 2011.

[111] Andrew B.J. teoh, Alwyn Goh, and David C.L. Ngo. Random multispace quantization as an analytic mechanism for biohashing of biometric and random identity inputs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1892–1901, 2006.

[112] Andrew B.J. Teoh and David C.L. Ngo. Biophasor: Token supplemented cancellable biometrics. In *Proceedings of the 9th International Conference on Control, Automation, Robotics and Vision (ICARCV'06)*, pages 1–5, 2006.

[113] Massimo Tistarelli, Stan Z. Li, and Rama Chellappa. *Handbook of Remote Biometrics: for Surveillance and Security.* Springer, 2009.

[114] Matthew A. Turk and Alex P. Petland. Face recognition using eigenfaces. In *Proceedings of 1991 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'91)*, pages 586–591, 1991.

[115] Jeffrey K. Uhlmann. Satisfying general proximity/similarity queries with metric trees. *Information Processing Letters*, 40:175–179, 1991.

[116] Masashi Une, Akira Otsuka, and Hideki Imai. Wolf attack probability: A theoretical security measure in biometric authentication systems. *IEICE Transactions on Information and Systems*, 91(5):1380–1389, 2008.

[117] Patrick Verlinde and Marc Acheroy. A contribution to multi-modal identity verification using decision fusion. In *Proceedings of PROMOPTICA*, pages 1–16, 2000.

[118] Patrick Verlinde and Gérard Chollet. Comparing decision fusion paradigms using k-NN based classfiers, decision trees and logistic regression in a multi-modal identity verification application. In *Proceedings of the Second International Conference on Audio- and Video-Based Biometric Person Authentication (AVBPA'99)*, pages 188–193, 1999.

[119] Enrique Vidal. An algorithm for finding nearest neighbors in (approximately) constant average time. *Pattern Recognition Letters*, 4:145–157, 1986.

[120] Abraham Wald. Some generalizations of the theory of cumulative sums of random variables. *The Annals of Mathematical Statistics*, 16, 1945.

[121] Abraham Wald. *Sequential Analysis.* Wiley & Sons, New York, 1947.

[122] Abraham Wald and Jacob Wolfowitz. Optimum character of the sequential probability ratio test. *The Annals of Mathematical Statistics*, 19(3):326–339, 1948.

[123] Zibo Wang, Abdul Serwadda, Kiran S. Balagani, and Vir V. Phoha. Transforming animals in a cyber-behavioral biometric menagerie with frog-boiling attacks. In *Proceedings of the IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS'12)*, pages 298–296, 2012.

[124] Neil Yager and Ted Dunstone. The biometric menagerie. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(2), 2010.

[125] Fan Yang, Michel Paindavoine, and Hervé Abdi. Fast panoramic face mosaicing and recognition. In *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis (ISPA'05)*, pages 197 – 202, 2005.

[126] Peter N. Yianilos. Data structures and algorithms for nearest neighbor search in general metric spaces. In *Proceedings of 4th ACM-SIAM Symposium on Discrete Algorithms (SODA'93)*, pages 311–321, 1993.

Appendix A

Related Publications

Journal Papers:

[A1] Takao Murakami, Kenta Takahashi, and Kanta Matsuura, “Toward Optimal Fusion Algorithms with Security against Wolves and Lambs in Biometrics,” IEEE Transactions on Information Forensics and Security, Vol.9, No.2, pp.259–271, 2014 (in press).

[A2] Takao Murakami, Kenta Takahashi, and Kanta Matsuura, “A General Framework and Algorithms for Score Level Indexing and Fusion in Biometric Identification,” IEICE Transactions on Information and Systems, Vol.E97-D, No.3, 2014 (in press).

[A3] 村上隆夫，高橋健太，松浦幹太，“大規模IDレス生体認証に向けた逐次索引融合判定の提案”，電子情報通信学会論文誌A（バイオメトリクス小特集），Vol.J96-A，No.12，pp.801–814，2013.

[A4] Takao Murakami, Kenta Takahashi, Susumu Serita, and Yasuhiro Fujii, “Probabilistic Enhancement of Approximate Indexing in Metric Spaces,” Information Systems Journal, Elsevier, Vol.38, No.7, pp.1007–1018, 2013.

[A5] 村上隆夫，高橋健太，“生体認証におけるWolfとLambに対する安全性の高い判定アルゴリズムの提案”，情報処理学会論文誌，Vol.51，No.12，pp.2319-2329，2010.

[A6] 村上隆夫，高橋健太，“多重仮説における逐次確率比検定を用いたIDレス生体認証の高精度化”，情報処理学会論文誌，Vol.50，No.12，pp.3186-3195，2009.

International Conference Papers:

[B1] Takao Murakami, Kenta Takahashi, and Kanta Matsuura, “Towards Optimal Countermeasures against Wolves and Lambs in Biometrics,” In Proceedings of IEEE Fifth International Conference on Biometrics: Theory, Applications and Systems (BTAS 2012), pp.69-76, 2012 **(Best Reviewed Paper)**.

[B2] Takao Murakami and Kenta Takahashi, “Fast and Accurate Biometric Identification Using Score Level Indexing and Fusion,” In Proceedings of IEEE/IAPR International Joint Conference on Biometrics (IJCB 2011), pp.1-8, 2011.

[B3] Takao Murakami, Kenta Takahashi, Susumu Serita, and Yasuhiro Fujii, “Versatile Probability-based Indexing for Approximate Similarity Search,”In Proceedings of 4th ACM International Conference on SImilarity Search and Applications (SISAP 2011), pp.51-58, 2011 **(Best Paper)**.

[B4] Takao Murakami and Kenta Takahashi, “Accuracy Improvement with High Convenience in Biometric Identification Using Multihypothesis Sequential Probability Ratio Test,” In Proceedings of First IEEE International Workshop on Information Forensics and Security (WIFS 2009), pp.66-70, 2009.

Internal Conference Papers:

[C1] 村上隆夫，高橋健太，松浦幹太，“ IDレス生体認証における最適な逐次融合判定について”，平成25年度8月バイオメトリクス研究会（BioX研究会），BioX2013-11，pp.34-39, 2013.

[C2] 村上隆夫，高橋健太，松浦幹太，“ WolfとLambに対する安全性と最適性を持つ融合判定の理論的考察”，第1回バイオメトリクス研究会（BioX研究会），BioX2012-13，pp.73-79, 2012.

[C3] 村上隆夫，高橋健太，“ 照合順序の最適化とスコア融合判定に基づくIDレス生体認証の高速・高精度化 ”，2011年 暗号と情報セキュリティシンポジウム（SCIS 2011），2011.

[C4] 村上隆夫，高橋健太，“ Wolf及びLambに対する安全性の高い生体認証の実験的評価 ”，バイオメトリックシステムセキュリティ研究会 第22回研究発表会，2010.

[C5] 村上隆夫，高橋健太，“ Wolf及びLambに対する安全性の高い生体認証の提案 ”，コンピュータセキュリティシンポジウム2009（CSS 2009），2009 **（CSS2009優秀論文賞受賞，平成22年度山下記念研究賞受賞）**.

[C6] 村上隆夫，高橋健太，“ 個人毎のスコア分布を用いた逐次的融合判定によるIDレス生体認証の高精度化 ”，2009年 暗号と情報セキュリティシンポジウム（SCIS 2009），2009.

[C7] 村上隆夫，高橋健太，“ 多重仮説における逐次確率比検定を用いたIDレス生体認証の高精度化”，コンピュータセキュリティシンポジウム2008（CSS 2008），2008 **（CSS2008優秀論文賞受賞）**.