IMPERFECT STEGOSYSTEMS –
ASYMPTOTIC LAWS AND
NEAR-OPTIMAL PRACTICAL CONSTRUCTIONS

BY

TOMÁŠ FILLER

M.S. in Computer Science, Czech Technical University, Prague, 2007

DISSERTATION

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate School of
Binghamton University
State University of New York
2011

Accepted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate School of
Binghamton University
State University of New York
2011

April 1, 2011

Jessica Fridrich, Chair and Faculty Advisor
Department of Electrical and Computer Engineering, Binghamton University

Andrew Ker, Member
Computing Laboratory, University of Oxford, UK

Scott Craver, Member
Department of Electrical and Computer Engineering, Binghamton University

Gaurav Sharma, Outside examiner
Department of Electrical and Computer Engineering, University of Rochester, NY

Abstract

Steganography is an art and science of hidden communication. Similarly as cryptography, steganography allows two trusted parties to exchange messages in secrecy, but as opposed to cryptography, steganography adds another layer of protection by hiding the mere fact that any communication takes place in a plausible cover traffic. Corresponding security goal is thus the statistical undetectability of cover and stego objects studied by steganalysis — a counterpart to steganography. Ultimately, a stegosystem is perfectly secure if no algorithm can distinguish its cover and stego objects.

This dissertation focuses on stegosystems which are not truly perfectly secure — they are imperfect. This is motivated by practice, where all stegosystems build for real digital media, such as digital images, are imperfect. Here, we present two systematic studies related to the secure payload loosely defined as the amount of payload, which can be communicated at a certain level of statistical detectability.

The first part of this dissertation describes a fundamental asymptotic relationship between the size of the cover object and the secure payload which is now recognized as the Square-root law (SRL). Contrary to our intuition, secure payload of imperfect stegosystems does not scale linearly but, instead, according to the square root of the cover size. This law, which was confirmed experimentally, is proved theoretically under very mild assumptions on the cover source and the embedding algorithm. For stegosystems subjected to the SRL, the amount of payload one is able to hide per square root of the cover size, called the root rate, leads to new definition of capacity of imperfect stegosystems.

The second part is devoted to a design of practical embedding algorithms by minimizing the statistical impact of embedding. By discovering the connection between steganography and statistical physics, the Gibbs construction provides a theoretical framework for implementing and designing such embedding algorithms. Moreover, we propose a general solution for implementing the embedding algorithms minimizing the sum of distortions over individual cover elements in practice. This solution, called the Syndrome-trellis code (STC), achieves near-optimal performance over wide class of distortion functions.

To my parents and my wife.

Acknowledgments

Completing this dissertation reminds me how much I should express thanks to people without which this work would never be finished.

The first person I would like to mention is my supervisor, Jessica Fridrich. Without her patience and interest, I would not spend the final year of my master studies at Binghamton University. This trip lead to another great 4 years in which the work presented in this dissertation was done. It was her, who came many times with interesting questions and ideas. I am grateful to her for her constant encouragement especially in designing my own graduate course on modern coding theory which I developed and taught in Fall 2010. Finally, I am grateful for the amount of support and freedom I have received in the final years of my PhD.

In my PhD, I have enjoyed collaboration with many colleagues, which helped me to realize the potential of ideas presented in this dissertation. I am indebted to Andrew Ker for sharing his enthusiasm for the Square-root law which resulted in an avalanche of my own results presented in Part I. The Gibbs construction would probably not be invented without the motivation I have received from the joint work on the HUGO algorithm I have developed with Tomáš Pevný and Patrick Bas for the Break Our Steganographic System (BOSS) contest. Finally, I have enjoyed collaboration with Jan Judas when visiting Binghamton University and working on the syndrome-trellis codes.

I am grateful to my alma mater back in Czech Republic for the way and style mathematics is being taught. Although not fully appreciated at that time, I realized the influence and broad scope of subjects they offer later while in Binghamton.

I would like to thank the Digital Watermarking Alliance for sponsoring the best paper award, which I have received in 2009 and 2010.

Many thanks belong to my parents and to my wife. Without the support and love, my parents gave me, I would not be able to realize the dreams I had. Finally, I would like to thank my wife Radka for her support and understanding when we were in Binghamton and Rye. Her love and friendship made my life in Binghamton much happier.

We can only see a short distance ahead, but we
can see plenty there that needs to be done.

— ALAN TURING, (1950)

Contents

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1 Steganography of Empirical Cover Sources

The idea of secret information exchange between trusted parties is following the human being since the very beginning. Cryptography achieves this goal by providing algorithms that make messages unintelligible to parties not possessing the proper decryption mechanism and key. In most cases, the mere fact that two parties are communicating using cryptography is obvious when the encrypted messages are intercepted. This may be inconvenient for the communicating parties due to the fact that their communication is detectable by a third party. This may result in further actions against them. This is the case when an authoritative third party controls the communication channel. In such cases, the communicating parties may be interested even in hiding the fact that they exchange encrypted messages, which is the goal of steganography. Steganography achieves this goal by hiding (possibly encrypted) messages into plausible traffic so that it is hard (if possible) to distinguish it from the original cover traffic. Similarly, as cryptography has its counterpart in cryptanalysis, steganography has its counterpart in steganalysis, the art and science of detecting hidden communication.

The central concept driving the security of steganographic systems is the *statistical detectability*, which is the ability of a third party, call it the warden, to distinguish plausible *cover* traffic from the traffic emitted by parties exchanging messages using steganography, the so-called *stego* traffic. The key role in security of steganographic systems is played by the source of cover objects, which may not be under the control of the communicating parties. To allow any form of hidden communication that is not trivially detectable and thus highly insecure, the source should have some form of randomness inside. For example, using the set of digits in the decimal expansion of π is not a good option for the cover source, since the symbols follow a specific pattern from which any deviation can be easily detected. On the other hand, a plausible cover source emitting random independently and identically-distributed (i.i.d.) bits is ideal but rarely seen for steganographic applications. When the sources of cover and stego objects are statistically identical, i.e., the probability distributions of both sources are equal, we speak of a *perfectly secure stegosystem* since then there is no space for the warden to mount any attack which may reliably accuse the communicating parties of using steganography. Although such systems are highly desirable, their existence in practice is limited to artificial cover sources[1] which can be described using a mathematical model with known probability distributions, such as the source of i.i.d. bits. Cover sources that can be described by known probability distributions are rarely plausible in practical situations.

Due to the ubiquity of real digital media available today, objects, such as digital images, audio or video streams became popular and thus plausible in many communication scenarios on the Internet. Similarly, the number of algorithms able to hide large amounts of data into a single image is also

[1] Böhme [9, Ch. 2.6] made a distinction between artificial and empirical covers. Artificial covers are “sequences of elements drawn from theoretically defined probability distribution” which lack any uncertainty about its possible parameters. Empirical covers, instead, are discretized samples of real measurements corresponding to real natural phenomena.

growing as steganography is more and more popular. Using real digital media as a source of covers is not only popular among users, but is also mathematically interesting for the research community. This is mainly because the cover source is much more complex than just a sequence of i.i.d. symbols, which has immediate consequences on statistical detectability of such stegosystems. Although it might be theoretically possible to assume that objects like digital images follow a certain probability distribution, algorithmic complexity of accurately estimating such a high-dimensional distribution forces the sender and the warden to rely on finite sample estimates, which are almost surely different from the true distribution. Böhme [9, Ch. 3] also argues that since real digital media represent an image of a real world, such distribution may be incognisable and thus never available to any party in practice. This is what makes steganography and steganalysis in complex cover sources, such as real digital media, an empirical discipline relying on different principles than when the cover source distribution is precisely known. Following [9], we call cover sources which do not permit us to obtain their exact probability distribution *empirical*.

Since the sender cannot obtain an accurate and complete statistical description of the cover source, she has to give up her hope for constructing perfectly secure stegosystems by preserving the cover source distribution. The best she can do is to utilize a finite set of genuine cover objects to learn as much as possible about the cover source and then embed the payload in such a way that the emitted stego objects are visually and *statistically* as close to the cover source as possible. This is indeed the case of *all* stegosystems designed for real digital images known until 2011. We call stegosystems that are not perfectly secure *imperfect*.

Although the loss of perfect security may seem as a big sacrifice, current steganalytic techniques are still not reliable enough to render the imperfect stegosystems unusable in practice. A closer study of imperfect stegosystems reveals many new possibilities that the sender can leverage to reduce the detectability in practice. On the other hand, new steganalytic techniques can hardly be designed without a stimulating and provocative development in steganography.

1.2 Dissertation Goal

The main focus of this dissertation is steganography – the problem of hiding messages. By studying the stegosystems, we also discover important consequences that influence the development and future design of steganalysis. By realizing the limits and by improving the practical tools one can use in steganography, we stimulate the design of new steganalytic techniques. Without such interplay, steganalysis can hardly ever reach the point when it can detect hidden messages reliably.

As for any communication system, designers should strive to maximize the amount of information to be transmitted, while satisfying given system requirements. For steganographic systems, the sender is interested in the so-called *secure payload* simply defined as the largest payload that can be embedded by a given steganographic algorithm into a specific source of cover objects without being detected. For the case of spatial domain digital images, the size of the payload is often expressed as the number of bits embedded per pixel, the so-called *relative payload*. Secure payload and its dependency on the size of the cover object is one of the key problems studied in this work.

For the case of perfectly secure stegosystems, the size of the secure payload one can embed in n independently obtained cover objects grows linearly with n. This is because no matter what the warden tries, any detector would make random guesses about the presence of any payload even when all n objects are used for detection. This supports our intuition about the relative payload as defined with respect to (w.r.t.) the size (or number) of cover objects. This also allows us to define *the capacity* of a given steganographic channel in a way which is common in information-theory, as a supremum over all relative payloads (communication rates) as n tends to infinity. Unfortunately, so far no perfectly secure stegosystem has been introduced for empirical cover sources, which motivates us to study the relationship between secure payload and the size (or number) of cover objects for imperfect stegosystems. The constraint of an imperfect stegosystem will render our result highly relevant for practice.

Surprisingly, the fact that we grant the warden a non-trivial detector dramatically changes the linear dependency of the secure payload on the size (or number) of cover objects. As shown by

Ker [67] for the problem of spreading the payload into multiple independent cover objects, the so-called batch steganography problem, the size of the payload can only grow at the order of $\sqrt{n}$ (no longer linearly) while keeping the same level of detectability. Embedding the same relative payload in every image allows us to construct detectors with arbitrary small errors as n tends to ∞. This is mainly because the more objects the warden can use to make the decision, the more reliable the decision can be. As it will turn out, the same asymptotic law holds not only w.r.t. the number of independent cover objects in the batch embedding problem, but even w.r.t. the number of locally dependent cover elements, such as pixels in a digital image, which renders the result very useful for most real digital media. This law, which is now known as the Square-Root Law (SRL), has a large impact on steganography and steganalysis both in theory and in practice. Among other consequences, this implies that only 2× larger message can be hidden in a 4× larger cover object. In other words, the same relative payload can be detected more reliably in larger images than in smaller ones. The size of the cover object is thus an important parameter one has to control when comparing steganographic methods on different cover sources.

From a pure information-theoretic point of view, the capacity of imperfect stegosystems is not very interesting since the maximum *relative payload* one can send while being at a fixed level of detectability converges to zero as the size of the cover object tends to ∞. The problem can be made much more useful and appealing if we normalize the payload by $\sqrt{n}$ instead of by n in the definition of the capacity. Since all imperfect stegosystems fall under the SRL, such quantity describes how many bits one can hide per square root of the cover size when measured at a fixed detectability level. This quantity, which we later call *the root rate,* appears to be a useful and theoretically well-founded way for comparing imperfect stegosystems.

Along with studying the asymptotic behavior of secure payload and its application for imperfect stegosystems, one may also be interested in techniques allowing us to implement imperfect stegosystems by minimizing statistical detectability in practice. As of 2011, virtually all embedding algorithms for real digital media follow a general hiding principle which calls for slightly modifying a genuine cover object when hiding a message. Although different algorithms interpret this principle differently, most of them embed a message by minimizing suitably defined distortion function. In fact, this approach leads to more secure stegosystems in practice when the distortion function is connected with statistical detectability. The advantage of the minimum-distortion framework is the fact that it can also be equipped with theoretical bounds connecting the relative payload with average distortion. Such bounds, often known as the rate–distortion bounds, inform the sender about the limits when a particular distortion function is used. The bounds can also be used for benchmarking embedding algorithms in practice.

Although the problem of embedding while minimizing a distortion function is well connected [3] to Shannon's source coding theory [108], only very special cases of distortion functions were studied in steganography literature. These limitations on the distortion function slowed down the development of new embedding methods because steganography designers had to face the same problem of how to communicate the payload by minimizing the distortion function again and again. This dissertation presents the first complete framework in which a large class of distortion functions can be used for embedding and thus removes the limitations in steganography design. We also present a practical implementation of this framework using syndrome coding, a standard technique used in information theory.

When applying the principles in practice, we focus on digital image steganography, where digital photographs of natural scenes are used as a source of cover objects. We believe that the derived principles and algorithms also apply to other empirical sources, such as audio and video streams.

1.3 Outline

This work is divided into two major parts which both study imperfect stegosystems but each from a different perspective. Both parts present the results of our own research with references to relevant works of others.

To introduce the reader to the field of digital image steganography, we present a short overview of results that will be relevant to other chapters. This overview, which is presented in Chaper 2, formulates steganography as the prisoners' problem, defines necessary terms and notation, and introduces several embedding algorithms for digital images. Although this dissertation is not targeted to advance steganalysis, we describe some of the techniques we will use to evaluate the security of practical embedding schemes that we develop later. After reading Chapter 1 and Chapter 2, both parts can be read independently since they require minimal knowledge about each other. The chapters in each part should be read in their respective order since they build on each other.

In Part 1, we study the question of how the secure payload scales with the size of the cover object for imperfect stegosystems. The general theme of this part is the Square-root law and its applications for empirical cover sources that show some level of dependency between cover elements. Chapter 3 includes the proof of the SRL for sources modeled as the first order Markov chain and precisely describes conditions under which the law does and does not hold. To prove this, we assume the embedding algorithm to perform mutually independent substitutions of individual cover elements, which covers a vast majority of embedding algorithms we know so far. As a byproduct of such analysis, we show that the set of all cover sources that results in a perfectly secure stegosystem with a fixed embedding algorithm, forms a linear space fully described by the embedding algorithm.

Under mild assumptions, all imperfect stegosystems fall under the SRL, i.e., as we increase the size of the cover object to n elements, the size of the payload that leads to the same level of detectability scales as $r\sqrt{n}$, for some positive constant r. In Chapter 4, we show that the constant r is inversely proportional to the quantity known as the Fisher information. The higher the r is, the more bits the sender can hide per square root of the size of the cover, which permits natural ordering of all imperfect stegosystems. For Markov cover sources, the Fisher information can be written in a simple closed-form expression giving us an opportunity to use it for the design of embedding algorithms.

Part 2 describes a complete framework and its practical implementation for designing imperfect stegosystems by minimizing an *arbitrary* distortion function between cover and stego images. In Chapter 5, we introduce the so-called Gibbs construction, which lays a foundation for the embedding framework by making a connection between steganography and statistical physics. This connection allows us to formalize the embedding problem, restate appropriate rate–distortion bounds and import algorithms allowing us to implement the schemes in practice. Most of the embedding schemes following the Gibbs construction can be transformed into a series of problems of embedding while minimizing distortion that is *additive* over individual cover elements – a simpler problem which was still not completely solved in steganography. The first general solution, the syndrome-trellis codes, is proposed in Chapter 6, where we describe their construction and design along with extensive experimental results.

Having such framework in our hand, a significant gain in secure payload can be achieved by placing the embedding changes adaptively w.r.t. a local neighborhood of individual cover elements. This is achieved within the framework by designing the distortion function so that the distortion corresponds with statistical detectability – a problem studied in Chapter 7. In this chapter, we present several studies of how adaptive algorithms for digital images in both spatial and DCT domain can be designed. The material presented in this chapter was motivated by the design of the HUGO algorithm [94] tested in the "Break Our Steganographic System" (BOSS) challenge organized in 2010 [4].

We believe the framework can be used by steganographers as an of-the-shelf tool when designing new embedding schemes and allow them to focus on the design of the distortion functions rather than on practical implementations of the embedding methods.

1.4 Notation

In the text, we adhere to the following general notation. We use $\mathbb{A} = (a_{i,j})$ to denote a matrix with elements $a_{i,j}$ and similarly for higher-order tensors $\mathbb{A} = (a_{i,j,k})$. Caligraphic font ($\mathcal{X}$) denotes sets, bold font ($\mathbf{x} = (x_1, \ldots, x_n)$) denotes vectors with elements x_i. Capital letters are used for random variables, bold for vector and regular for scalar variables, such that $Pr(X = x)$ denotes the probability that random variable X equals x. We write $L \triangleq R$ when we want to define the symbol L with expression R. The symbols $\mathbb{R}$, $\mathbb{N}$ denote the set of real and natural numbers, respectively.

We use $\mathbf{x} \triangleq (x_1, \ldots, x_n) \in \mathcal{X} \triangleq \mathcal{I}^n$ and $\mathbf{y} \triangleq (y_1, \ldots, y_n) \in \mathcal{X}$ exclusively for an$n$-element cover and stego object with dynamic range $\mathcal{I}$, respectively. For 8-bit grayscale images $x_i \in \mathcal{I} = \{0, \ldots, 255\}$. In general, x_i can stand not only for light intensity values in a raster image but also for scalar transform coefficients, palette indices, audio samples and even for RGB color triples, for example $\mathcal{I} = \{0, \ldots, 255\}^3$. Depending on the character of the cover source, elements $\{x_i | i \in \mathcal{S}\}$, $\mathcal{S} \triangleq \{1, \ldots, n\}$, are organized in a regular lattice with index set $\mathcal{S}$. Given $\mathcal{J} \subset \mathcal{S}$, $\mathbf{x}_{\mathcal{J}} \triangleq \{x_i | i \in \mathcal{J}\}$ and $\mathbf{x}_{\sim\mathcal{J}} \triangleq \{x_i | i \in \mathcal{S} - \mathcal{J}\}$. The image $(x_1, \ldots, x_{i-1}, y_i, x_{i+1}, \ldots, x_n)$ will be abbreviated as $y_i \mathbf{x}_{\sim i}$. When working with spatial domain digital images, we will need to address pixels by their two-dimensional coordinates. We will thus be switching between using the index set $\mathcal{S} = \{1, \ldots, n\}$ and its two-dimensional equivalent $\mathcal{S} = \{(i,j) | 1 \leq i \leq n_1, 1 \leq j \leq n_2\}$, $n = n_1 n_2$, hoping that it will cause no confusion for the reader. We reserve P and Q for probability distributions of cover and stego objects, respectively. Sometimes we write $P^{(n)}$ or $Q^{(n)}$ to denote the number of cover elements. In some cases, the distribution of stego images is parametrized by a scalar parameter describing the size of the payload or the number of changes and denoted as Q_α or Q_β.

If $\mathbf{y}$ is a vector with components $\mathbf{y} = (y_1, \ldots, y_n)$, $\mathbf{y}_k^l$ denotes the subsequence $\mathbf{y}_k^l = (y_k, \ldots, y_l)$. If $\mathbf{Y} = (Y_1, \ldots, Y_n)$ is a random vector with underlying probability distribution P, then $P(\mathbf{Y}_k^l = \mathbf{y}_k^l)$ denotes the marginal probability $P(Y_k = y_k, Y_{k+1} = y_{k+1}, \ldots, Y_l = y_l)$.

We exclusively use $\mathbf{m} = (m_1, \ldots, m_m) \in \{0,1\}^m$ to denote an m-bit message the sender wants to hide. The symbol $D : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is solely used for the distortion function, i.e., $D(\mathbf{x}, \mathbf{y})$ denotes the distortion between cover $\mathbf{x}$ and stego $\mathbf{y}$. Specific forms of D are defined in individual chapters.

We also use the Iverson bracket, $[S]$, defined as $[S] = 1$ when the statement S is true and zero otherwise. Finally, we use $\log_2 x$ for the logarithm at the base of 2 and reserve $\ln x$ for the natural base, $h(x) = -x \log_2 x - (1-x) \log_2 (1-x)$ is the binary entropy function, and $H(\boldsymbol{\pi}) = -\sum_{i=1}^{k} \pi_i \log_2 \pi_i$ is the entropy of probability distribution $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_k) \in [0,1]^k$.

1.5 Historical Context

Steganography started as an ancient art of hiding secret messages more than 4000 years ago when Egyptians started to use hieroglyphics. Hieroglyphic text consisted from a set of pictograms with different meaning for different groups of readers. Later, the Greeks report successful usage of steganography to communicate messages through enemy field which would otherwise be intercepted. Herodotus describes in “The Histories” a story of Histaiaeus, later tyrant of Miletus, who instigated a revolt in Ionia against the Persians in 499 BC. He informed Aristagoras to start the revolt by sending a message tattooed on the head of his most trusted slave. He shaved his head, tattooed the message on it, and let his hair regrow. When the slave arrived to Aristagoras, he was instructed to shaved the slave’s head to read the message. This way, the slave did not know the content of the message and did not carry anything suspicious when traveling through enemy field.

In World War II, the Germans used a technique called “microdot” allowing them to shrink a one-page document to 1 mm in size using photographic techniques. The shrunk document was then cemented to a seemingly innocuous letter as a period mark. Microdots were used to communicate messages through insecure channels. The same steganographic technique was used in Germany after the Berlin Wall was put up to pass the censors when the messages were sent through ordinary mail.

With the invention and spread of the Internet and digital media, steganography turned from being an art to actual science. Since then many news have speculated of steganography being used for plotting terrorist attacks. Unfortunately no case was officially reported by any government organization on this. The first officially reported use of steganography over the Internet was in June 2010 when the Russian spy gang was broken [107]. The FBI reported that Russian spies hid secret information in images which they posted on the Internet. As described in the news, their steganographic method was broken by revealing the keys used by the embedding method.

More detailed history of steganography can be found in the book of Kipper [75].

Chapter 2

Steganography of Digital Images

2.1 Prisoners' Problem and the Subliminal Channel

Steganography is often explained as a communication between three fictitious identities, Alice, Bob, and Eve as first introduced by Simmons [112] in his prisoners' problem. Consider Alice and Bob being two criminals closed in separate cells and suspected of committing a crime. Being convinced that Alice and Bob are guilty, Eve allows them to communicate between each other while carefully inspecting every message and looking for any piece of supporting evidence that may help her in further accusation. After Eve gathers enough evidence from such communication, she places Alice and Bob into solitary confinement and cuts off the communication channel. Alice and Bob, knowing that they are being monitored, want to cook up an escape plan without being caught by Eve. An option for Alice and Bob is to first agree on a steganographic method before being arrested and then use it to hide their communication in a traffic allowed by Eve. Simmons described a solution to this problem, the so-called subliminal channel, based on digital-signature crypto systems[1].

When designing the data-hiding algorithm, Alice and Bob should make their strategy public and rely on Eve's inability to obtain the secret keys known only to Alice and Bob. This guideline, better known as the Kerckhoffs' principle named after the Dutch cryptographer Auguste Kerckhoffs, is one of the key principles often used in modern cryptography. However, this principle may not be as straightforward to apply in steganography since steganographers may have more degrees of freedom when designing the systems. For example, this principle can be interpreted differently in batch steganography, when the sender is allowed to spread the payload in many cover objects since different levels of knowledge can be granted to the warden [70, Sec. 1.2]. In this dissertation, we mostly study stegosystems that do not follow the batch paradigm and thus by the Kerckhoffs' principle, we grant the warden complete knowledge about the cover source and the embedding algorithm.

2.1.1 Problem Formulation

Figure 2.1.1 summarizes the communication setup. Alice wants to send message $\mathbf{m}$, which she first encrypts by a pre-agreed secret key and then embeds into a randomly sampled genuine cover image. While embedding a message, she utilizes the pre-agreed stego key which may drive the embedding function when producing the stego image. The stego image is sent through the channel and inspected by Eve's detection algorithm, the steganalyzer. Eve has to gather evidence about the type of communication from a single output of the steganalyzer and has to decide whether or not to cut the communication channel. If the stego image is delivered to Bob, he extracts the message by inverting the embedding operation. By doing this, he does not need to recover the original cover image, which served as a decoy. In practice, the stego key may also be used to identify images from which Bob should extract the payload. This problem setup is known as *sequential steganalysis.*

[1] Subliminal channels based on digital signatures were of practical interest, see [75, Sec. "U.S./U.S.S.R Nuclear Arms Treaties"] and [1].

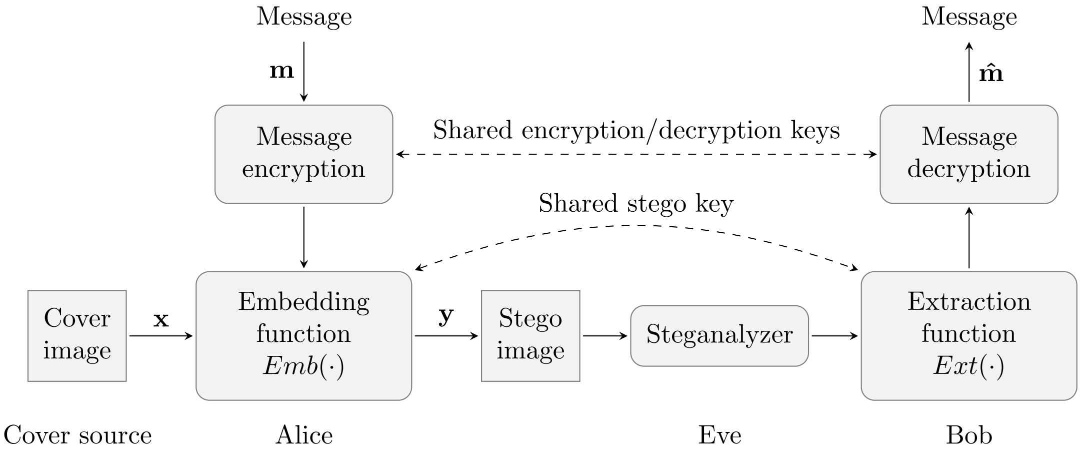

Figure 2.1.1: Model of steganographic communication channel.

There are many practically-relevant modifications of this communication setup which we omit in our discussion. For example, the already-mentioned batch steganography with *pooled steganalysis* as its counterpart. Similarly, the cover object does not need to be of size known a priori to Eve, i.e., we may be interested in working with audio or video streams which may also change the assumptions considerably.

From this perspective, the problem of sequential steganalysis can be interpreted as a binary hypothesis testing problem that decides between hypothesis H_0, a given image $\mathbf{z}$ is cover, or H_1, a given image $\mathbf{z}$ is stego. Depending on the knowledge of the size of the payload available to Eve, she can either interpret this as a simple binary hypothesis test if the size of the relative payload, say $\alpha_0 > 0$, is known

$$H_0 : \mathbf{z} \sim Q_0, \qquad H_1 : \mathbf{z} \sim Q_{\alpha_0}, \tag{2.1.1}$$

or as a binary hypothesis test with composite alternative when the payload is not known

$$H_0 : \mathbf{z} \sim Q_0, \qquad H_1 : \mathbf{z} \sim Q_{\alpha},\ \alpha > 0. \tag{2.1.2}$$

Here, Q_α denotes the distribution of images with payload α, i.e., Q_0 denotes the distribution of the cover images[2]. The Bayesian approach, where the size of the relative payload α is described by a prior distribution, is also possible, but requires the knowledge of such quantity which may not be available.

When classifying a given image, Eve can make two kinds of errors which are often of different practical importance. She can classify a stego image as cover or classify an innocent cover as stego. The corresponding probabilities are called *probability of missed detection*, P_{MD}, and *probability of false alarm*, P_{FA}, respectively. These two probabilities completely describe the error of Eve's steganalyzer and depend on each other. The graph showing the dependency between $1 - P_{\mathrm{MD}}$ and P_{FA}, known as the Receiver Operating Characteristic (ROC) curve, is the complete descriptor of steganalyzer's performance. For its two-dimensional character, ROC curves are rarely used for comparison of two different embedding methods since both curves may cross each other leading to different conclusion for different values of P_{FA}. For this reason, many ROC-based scalar measures were introduced which allow ordering of different stegosystems based on their security. Among others, the area under the curve, the detection rate $1 - P_{\mathrm{MD}}$ at a fixed value of P_{FA}, are being used. To remain consistent with current literature [37], we use the *minimum error under equal priors* P_{E}, defined as

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{P_{\mathrm{MD}} + P_{\mathrm{FA}}}{2} \tag{2.1.3}$$

[2] Since we assume empirical cover sources, neither Alice, nor Eve can obtain distribution Q_α for any $\alpha \geq 0$ in practice. This implies that, although optimal solutions to these hypothesis problems are theoretically possible (for example from the Neyman-Pearson lemma), they are not feasible in practice.

to measure the level of security. Obviously, $0 \leq P_{\mathrm{E}} \leq 0.5$, with $P_{\mathrm{E}} = 0.5$ being a perfectly secure stegosystem w.r.t. a given steganalyzer. Using the above language, a stegosystem is perfectly secure iff the Kullback-Leibler divergence between Q_0 and Q_α,

$$D_{\mathrm{KL}}(P||Q_\alpha) \triangleq \sum_{\mathbf{z}\in\mathcal{X}} Q_0(\mathbf{X}=\mathbf{z}) \ln \frac{Q_0(\mathbf{X}=\mathbf{z})}{Q_\alpha(\mathbf{Y}=\mathbf{z})} \tag{2.1.4}$$

is zero, since only then $Q_0 = Q_\alpha$ [11, 17]. A stegosystem satisfying $D_{\mathrm{KL}}(Q_0||Q_\alpha) \leq \epsilon$, for some $\epsilon > 0$, is called *ϵ-secure*.

By bounding the KL divergence from above, we force Eve to make non-zero errors in the following sense [11]. Let $f : \mathcal{X} \rightarrow \{0,1\}$ be a deterministic function describing Eve's steganalyzer calling $\mathbf{z}$ cover if $f(\mathbf{z}) = 0$ and stego otherwise. Since any deterministic processing does not increase the KL divergence, we have $D_{\mathrm{KL}}(Q'_0||Q'_\alpha) \leq D_{\mathrm{KL}}(Q_0||Q_\alpha) \leq \epsilon$, where Q'_0 and Q'_α are distributions of $f(\mathbf{X})$ and $f(\mathbf{Y})$, respectively. Using this notation, $P_{\mathrm{FA}} = Pr(f(\mathbf{X}) = 1) = Q'_0(1)$, $P_{\mathrm{MD}} = Pr(f(\mathbf{Y}) = 0) = Q'_\alpha(0)$, and thus

$$D_{\mathrm{KL}}(Q'_0||Q'_\alpha) = P_{\mathrm{FA}} \ln \frac{P_{\mathrm{FA}}}{1-P_{\mathrm{MD}}} + (1-P_{\mathrm{FA}}) \ln \frac{1-P_{\mathrm{FA}}}{P_{\mathrm{MD}}} \leq \epsilon. \tag{2.1.5}$$

For example, enforcing $P_{\mathrm{FA}} = 0$ leads to $P_{\mathrm{MD}} \geq e^{-\epsilon}$.

In all of the descriptions above, we have assumed that Eve only inspects the images traveling over the channel and does not maliciously change any of them. This regime of operation, often called *passive warden*, is usually assumed in situations, where it may be practically infeasible for the warden to make any malicious changes, such as when the digital media is spread over the Internet. The opposite form of operation, the *active warden*, assumes modifications which may or may not be under Eve's control. For example, the timing covert channel, when the sender modifies delays between computer packets sent over the network, naturally exhibits some form of noise due to different physical conditions each packet encounters while being routed. The case of an active warden requires slightly different tools for constructing the stegosystems and the problem more resembles the digital watermarking problem with slightly different embedding criteria. In this work, we specifically deal with passive-warden scenario only.

In the rest, we briefly discuss some basic algorithms and practices in steganography and steganalysis which will be used to motivate our further development.

2.2 Steganography

2.2.1 Embedding Paradigms

A large number of different embedding algorithms can be divided into the following 3 categories.

- **Steganography by cover selection:** To communicate one bit of information, Alice and Bob can agree on a keyed cryptographic hash function which partitions the space of all cover images into 2 halves such that 0's and 1's can be seen with probabilities as close to 0.5 as possible. To send a specific bit, Alice samples her cover source until she finds an image whose hash equals the desired payload and sends it to Bob. By doing this, her method is undistinguishable from regularly exchanged images as long as bits extracted from innocent cover images are undistinguishable from a random bit stream [11, Sec. 4]. Unfortunately, as the size of the payload grows, this method becomes impractical due to he exponential number of images Alice has to discard before seeing the right one.

- **Steganography by cover synthesis:** The largest number of bits Alice can ever send, while still being perfectly secure is the entropy of the cover source distribution, $H(Q_0)$. If such distribution is available to both Alice and Bob, they can, in theory, construct perfectly secure stegosystem as follows [1]. Alice and Bob first constructs an optimal compression algorithm which can compress their cover source to $H(Q_0)$. When Alice wants to send a message to

Bob, she feeds the message to the decompressor and obtains a perfectly plausible object. Bob reads the message by compressing the object. Many other theoretical methods [124, 101] allow construction of perfectly secure stegosystems under the crucial assumption that the cover source distribution is known.
The cover synthesis approach may also be used when multiple images of one scene are available to create a conditional statistical model. This model can then be used to obtain new stego images by merely sampling from the model. Such a technique, however, is not expected to be perfectly secure, although it may give some advantage to Alice over Eve who has access to only one image from a given scene. This idea has been studied by Franz in the context of multiple images obtained from a scanner [33, 34].

- **Steganography by cover modification:** The most commonly used approach for embedding messages is to slightly modify certain pixels or transform coefficients of a genuine cover image.

Since this dissertation deals exclusively with the last paradigm for image steganography, we describe some of the algorithms proposed in the literature.

2.2.2 Simple Embedding Operations

Perhaps the simplest algorithm for image steganography, the so-called Least Significant Bit (LSB) embedding, works by replacing the least significant bits of pixels along a key-dependent path with the message bits, i.e., the ith cover pixel in a given color channel, $x_i \in \mathbb{N}$, is replaced by

$$y_i = x_i + m_i - (x_i \bmod 2), \tag{2.2.1}$$

where m_i is the corresponding message bit. This method is particularly popular for its very easy implementation[3] and visual imperceptibility in most natural-scene images. Basic reasoning behind this algorithm is the fact that every image contains some form of natural noise and thus individual least significant bits, when placed on a separate plane, appears to be random. This situation may change dramatically when the image contains contiguous regions of pixels saturated at their dynamic range.

The same embedding operation can be easily used on any numerical data. When used on JPEG images, DCT coefficients $x_i \in \{0, 1\}$ are left unmodified because embedding into 0s would lead to highly disturbing artifacts. When skipping 0s, 1s are skipped as well for easier message extraction, exactly as in the Jsteg algorithm [116]. The same embedding operation can also be applied to higher order bit planes [69].

Since even colors can only be increased by (2.2.1) and odd colors can only be decreased, the operation of replacing LSBs introduces a strong asymmetry which leads to a large number of accurate detectors in both spatial-domain [21, 40, 62, 63, 66, 68, 72] and DCT-domain images [8, 78, 81, 82]. By this, the sum of two neighboring histogram bins, $|\{y_i|y_i = 2k\}| + |\{y_i|y_i = 2k+1\}|$, remains unchanged after embedding for any $k \in \mathbb{N}$.

An embedding operation which changes the pixel value by $+1$ or -1 uniformly at random if $x_i \bmod 2 \neq m_i$ leads to a method, called ± 1 embedding, which is much harder to detect than the LSB embedding. This is mainly because the randomization breaks embedding invariants which LSB embedding has. By changing the non-matching pixels by ± 1, the steganographer can also potentially embed ternary symbols by making the same number of changes and increasing the available payload from 1 bit per pixel (bpp) to $\log_2 3 \approx 1.53$ bpp.

2.2.3 Minimize Number of Changes - Matrix Embedding

When embedding m bits into an n-pixel cover, each pixel's LSB matches the required message bit with probability $1/2$ and thus $m/2$ changes are required on average. If $m < n$, a much smaller

[3] LSB embedding can be implemented using one line of Perl code as shown by Ker [62, p. 99]. This fact may be useful for criminals who do not want to leave any traces of a special-purpose software on their computers since Perl is generaly available on most platforms.

number of changes are required when the whole cover image can be used. This allows us to minimize the statistical detectability of the method, while embedding the same payload and thus makes the method more efficient. We illustrate this on a method called matrix embedding, which was first introduced by Crandall [19] in 1998.

Suppose we want to hide an m-bit message $\mathbf{m} = (m_1, \ldots, m_m)$ in an n-pixel cover using any of the above-described embedding algorithm and let $n = 2^m - 1$. Instead of just using m pseudorandomly-chosen pixels for embedding, Alice uses all $2^m - 1$ of them with the advantage of changing at most one instead of $m/2$ on average. Alice first constructs a matrix $\mathbb{H} \in \{0,1\}^{m \times 2^m - 1}$ containing all m-bit vectors except the one with all 0's as its columns. If, by any chance, $\mathbb{H}(\mathbf{x}^T \bmod 2) = \mathbf{m}^T$ when calculated using binary arithmetic and element-wise modulo operation, no change in the cover is necessary. If not, she starts with $\mathbf{y} = \mathbf{x}$ and applies her embedding operation to change the LSB of one stego pixel such that $\mathbb{H}(\mathbf{y}^T \bmod 2) = \mathbf{m}^T$ holds. This is always possible to do by changing exactly one LSB since the matrix $\mathbb{H}$ contains all possible combinations of 0's and 1's except for the all-zero vector. Upon receiving $\mathbf{y}$, Bob simply calculates $\mathbb{H}(\mathbf{y}^T \bmod 2)$ to obtain the message.

This trick was recognized by Crandal as a specific instance of the covering problem known in coding theory and later analyzed by Bierbrauer [6]. In fact, the above matrix $\mathbb{H}$ is a parity-check matrix of a binary Hamming code [84, Ch. 1.2]. The same algorithm can be used with the ± 1 embedding operation over ternary alphabet if all $\bmod 2$'s are replaced with $\bmod 3$'s, message is represented in ternary $\{0,1,2\}$ alphabet, and if we use a ternary Hamming code. Bierbrauer [6] also described finite and asymptotic bounds on the average number of changes required to communicate a given payload — the so-called rate–distortion bounds. For a given relative payload $0 \leq \alpha = m/n \leq 1$, the normalized number of changes one has to make using a binary embedding operation, such as LSB replacement, and averaged over different messages embedded into $\mathbf{x}$ as n tends to ∞,

$$d = \lim_{\substack{n \to \infty \\ m = \alpha n}} E_{\mathbf{M} \in \{0,1\}^m} \left[\sum_{i=1}^{n} [x_i = y_i(\mathbf{M})]/n \right],$$

satisfies $d \geq h(\alpha)$. Since 1998, many practical algorithms have improved the performance [117, 106, 48, 7, 38, 129].

2.2.4 Avoid Changing Forbidden Cover Elements - Wet Paper Codes

Matrix embedding was first put into use by Westfeld [125] in his F5 embedding algorithm for JPEG images. F5 was designed from Jsteg by removing several of its weaknesses. It communicates the message in LSBs of non-zero AC DCT coefficients. By avoiding changes in zero AC DCT coefficients, the algorithm avoids strong statistical artifacts that may arise due to strong peak in the center of the histogram of AC DCT coefficients. By removing all zero AC DCT coefficients, more coefficients would represent message bit 1 than 0 and thus, instead of extracting LSBs directly, F5 extracts each bit from the coefficient using a modified bit-extraction function p_{F5} defined as $p_{\mathrm{F5}}(y_i) = y_i \bmod 2$ if $y_i > 0$ and as $p_{\mathrm{F5}}(y_i) = (1 - y_i) \bmod 2$ otherwise. The embedding operation was also changed to better fit the histogram shape. If the bit extracted from x_i does not match the message bit, it is changed by decreasing its absolute value by one, i.e., $y_i = x_i - 1$ if $x_i > 0$ and to $y_i = x_i + 1$ otherwise. The histogram of AC DCT coefficients calculated from the stego image embedded with F5 appears as if the original image was compressed with a slightly lower quality factor. Unfortunately the fact that all zero AC DCT coefficients are omitted by Bob when reading the message, causes slight difficulties in practical implementation know as the *shrinkage problem*. When a bit 0 needs to be embedded in $x_i = 1$ or bit 1 in $x_i = -1$, the coefficient needs to be changed into $y_i = 0$ and is thus omitted by the reader. In practice, the number of these cases is not negligible due to the relatively high number of coefficients with $|x_i| = 1$.

The original implementation of the F5 algorithm solves the shrinkage problem by re-embedding the same bit again as shown in Figure 2.2.1 and thus loses on capacity. The shrinkage problem was removed completely in the non-shrinkage F5 (nsF5) algorithm by Fridrich [47] by using the so-called Wet Paper Codes (WPC). WPCs allow the sender to specify a set of cover element indices $\mathcal{W} \subset \{1, \ldots, n\}$, called *wet elements*, which are not allowed to be modified when embed-

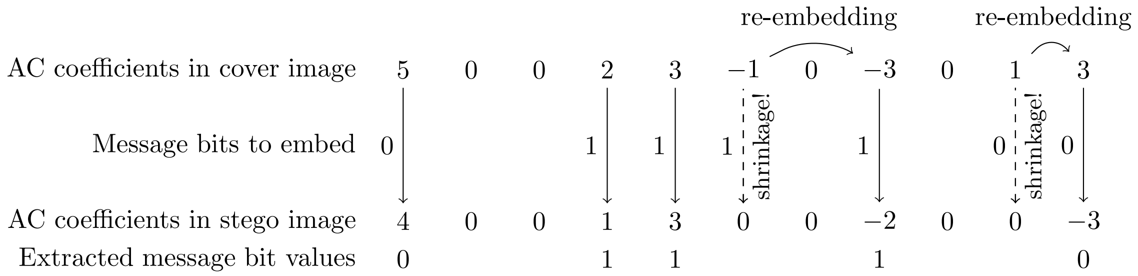

Figure 2.2.1: Example of embedding message using F5 algorithm.

ding the message. The index set $\mathcal{W}$ is not shared with Bob, which is what makes WPCs relevant for removing the shrinkage problem since the original positions of zero AC DCT coefficients are not known and many new zero AC DCT coefficients may appear after embedding. We set $\mathcal{W} = \{i|x_i \text{ is AC DCT coefficient and } x_i = 0\}$. The only quantity allowed to be shared is the number of wet elements $|\mathcal{W}|$ which is often communicated aside using a few bits.

WPCs follow a similar approach as matrix embedding, i.e., the message is communicated as a syndrome of the stego object calculated w.r.t. a pre-agreed parity-check matrix $\mathbb{H} \in \{0,1\}^{m\times n}$. This technique, called syndrome coding or binning, will be described in Chapter 6 in great detail. Here we show a very basic implementation as originally described in [45]. Let $\mathbb{H} \in \{0,1\}^{m\times n}$ be a parity-check matrix with elements generated uniformly at random from $\{0,1\}$ using the shared stego key. Alice, given a sequence of all AC DCT coefficients $\mathbf{x}$, wants to find *any* solution to

$$\mathbb{H}p_{\text{F5}}(\mathbf{y})^T = \mathbf{m}^T, \tag{2.2.2}$$

where $p_{\text{F5}}(\mathbf{y}) = (p_{\text{F5}}(y_1), \ldots, p_{\text{F5}}(y_n))$ is the sequence of modified stego AC DCT coefficients while satisfying the value of wet elements, i.e., $y_i = x_i$ for all $i \in \mathcal{W}$. Since wet elements cannot be changed, they can be substituted into (2.2.2) and thus the system of linear equations can be reduced to

$$\mathbb{H}'p_{\text{F5}}(\mathbf{y}')^T = \mathbf{m}^T - \mathbf{m}'^T, \tag{2.2.3}$$

where $\mathbb{H}' \in \{0,1\}^{m\times n-|\mathcal{W}|}$ and $\mathbf{y}'$ do not contain columns corresponding to wet elements and $\mathbf{m}'$ represents the part from these missing columns. As long as $\mathbb{H}'$ is of full rank (in binary arithmetic), then there is at least one solution to (2.2.3). This solution can be found using Gaussian elimination as long as the cubic complexity allows this. Larger cover objects can either be processed in a block-by-block manner or more efficient implementations, such as the one proposed in [42], can be used. If the number of changeable coefficients $n - |\mathcal{W}| > m$, then more than one solution to (2.2.3) exists and the solution requiring fewer number of changes than $m/2$ can be found. Such WPCs were proposed in [44] and tested with the nsF5 algorithm in [47].

2.2.5 Place the Embedding Changes Adaptively

All embedding algorithms introduced so far treat all embedding changes in allowed coefficients as having equal impact on statistical detectability. One can easily imagine that certain pixels or transform coefficients, such as those in heavy textured areas, can be changed more frequently than those in saturated or flat areas. Similarly, when Alice obtains her cover images by a process requiring any form of quantization, such as JPEG compression, the quantization errors provide a guideline about errors made by changing such coefficients. The knowledge of quantization errors may give some advantage to Alice, since this form of side information is not available to Eve. The rationale behind this is that by changing coefficients which are close to the quantization boundary should lead to smaller errors when the image is represented in spatial domain. Perhaps, such changes are harder to detect than changes which make larger spatial-domain errors. Coefficients that are exactly on the quantization boundary are thus the best for embedding since the direction where they should be quantized is implementation specific and very sensitive to noise.

This methodology of constructing embedding schemes, called Perturbed Quantization (PQ), was put forward by Fridrich [43] and, in fact, was the first application of wet paper codes in steganography. In the original implementation of PQ using WPCs, only m AC DCT coefficients[4] with the smallest quantization errors were allowed to be modified, whereas all other coefficients were assumed to be wet. To artificially increase the number of coefficients close to the quantization boundary for JPEG images, the embedding algorithm can embed the message into images that were first pre-compressed using a smaller quality factor [41].

In [74], Kim proposed a different way of embedding a message while minimizing the quantization error for JPEG images by using a modification of the original matrix embedding technique. Instead of making at most one change when embedding the message, the Modified Matrix Embedding (MME) algorithm gives Alice a list of different alternatives communicating the same message. By weighting each possibility, Alice can pick the one leading to the smallest rounding error or distortion even if this happens with more changes. For example, when $\mathbf{m} = (1, 0, 0)$, $\mathbf{x} = (0, 0, 0, 0, 0, 0, 0)$, and

$$\mathbb{H} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix},$$

Alice can choose either $\mathbf{y} = (1, 0, 0, 0, 0, 0, 0)$ with one change, or $\mathbf{y} \in \{(0, 1, 1, 0, 0, 0, 0), (0, 0, 0, 1, 1, 0, 0)\}$ with two changes, or vectors $\mathbf{y}$ with 3 changes to satisfy $\mathbb{H}\mathbf{y}^T = \mathbf{m}^T$. Such freedom leads to an improved algorithm for JPEG images.

The MME algorithm requires a never-compressed image on its input and embeds the message into LSBs of non-zero AC DCT coefficients. Let $\mathbf{c} = (c_1, \ldots, c_n) \in \mathbb{R}^n$ be a sequence of real AC DCT coefficients which, in ordinary JPEG compression, are quantized into cover elements $x_i = round(c_i/q_i) \neq 0$ using quantization table entries $q_i > 0$ and rounding function $round(x) = \lfloor x + 0.5 \rfloor$. If the cover element x_i needs to be modified, it is increased by 1 whenever $c_i/q_i > x_i$ and decreased by 1 otherwise. To resolve the shrinkage problem when bit 0 needs to be communicated in $x_i = 1$, the element is changed to $y_i = 2$ and similarly $x_i = -1$ is changed to $y_i = -2$ instead of $y_i = 0$. While embedding the message, the algorithm minimizes the total distortion $\sum_{i=1}^{n} r_i$ due to individual rounding errors

$$r_i = \begin{cases} 1 + |c_i/q_i - y_i| & \text{if } c_i/q_i \in [-1, -0.5] \cup [0.5, 1] \\ 1 - |c_i/q_i - y_i| & \text{otherwise.} \end{cases} \tag{2.2.4}$$

Depending on the number of allowed changes, MME will be denoted as MM1, MM2, or MM3. The MM3 algorithm placed among the best available for steganography in JPEG images in 2007 [47]. Since then, different modifications were proposed, such as the work of Zhang [128] and Sachnev [102], where they proposed the use of BCH codes along with a modified distortion function. According to the reported results, the improvement over MM3 was significant. These results have partly motivated the work we present in Part II.

2.3 Steganalysis

Depending on the amount of information Eve has about the stegosystem, different sequential steganalytic attacks can be divided into the following categories. All categories require the knowledge of the embedding algorithm by the Kerckhoffs' principle.

- **Targeted steganalysis:** Targeted attacks exploit specific weaknesses of the embedding algorithm for which they are designed. Such attacks often cannot be extended beyond the scope of the embedding algorithm. Many targeted attacks were described on the LSB embedding in both spatial [40, 21, 62, 63, 66, 68, 72] and DCT domain [82, 81, 8, 78].
- **Blind steganalysis:** Blind steganalysis more resembles the empirical notion of steganalysis since it represents cover and stego objects in a lower-dimensional feature space where cover and

[4] In practice, the number of dry coefficients has to be slightly larger than m to assure that the matrix $\mathbb{H}'$ in (2.2.3) will be of full rank.

stego objects can be well separated by a machine learning algorithm. Features should represent quantities, which are mostly content independent and change significantly after embedding. Different features can be combined together, which represents more general and today's most-often-used approach for detecting steganography. When Eve knows the size of the payload, the hypothesis testing problem is implemented using binary classifiers such as Fisher linear discriminant, or perhaps most often using Support Vector Machine (SVM) with either linear or Gaussian kernel [57]. If no information about the payload is available, Eve has several options [91] among which she can try estimate the payload using statistical regression, i.e., use quantitative steganalysis.

- **Quantitative steganalysis:** Quantitative attacks allow Eve to estimate the size of the payload. Methods can either be targeted (such as for Jsteg [78]) or feature-based [97]. Feature-based methods often perform statistical regression to learn the mapping from the feature space to payload size allowing us to reuse the feature mappings developed for blind steganalysis.

We have deliberately omitted the pooled steganalysis [65] designed to detect messages split into multiple cover objects. Such a technique did not receive much attention so far although the batch steganography paradigm is highly relevant.

In the rest of this section, we describe a commonly-used experimental setup, which will be further used to evaluate different embedding schemes discussed in this dissertation. We follow the blind steganalysis approach to evaluate the security of the stegosystem assuming Eve knows the size of the embedded payload. To train and test the steganalyzer, we use a large database of images, which we evenly split into training and testing parts. In each experiment, we will report a specific image database that was used. Depending on the image domain, we extract features from both cover and stego image obtained by embedding a pseudo-random payload using a given embedding algorithm. The SPAM features [93] and the CCPev [77][5] features were used for grayscale spatial-domain and JPEG images, respectively. We also combine these features together and use them in both image domains which results in the so-called Cross-Domain Features [79] (CDF) by either decompressing the JPEG image to spatial domain or by compressing the spatial-domain image using JPEG with 100% a quality factor. Both SPAM and CCPev features are described below for reference since both feature sets will serve as a starting point for designing new embedding algorithms in Chapter 7.

We chose the soft-margin support-vector machine with Gaussian kernel as implemented in the LIBSVM package[14] to perform binary classification. The kernel width γ and the penalty parameter C were determined using five-fold cross validation on the grid $(C,\gamma) \in \big\{(10^k, 2^j) | k \in \{-3,\dots,4\}, j \in \{-L-3,\dots,-L+3\}\big\}$, where $L \triangleq \log_2 d$ is the binary logarithm of the number of features. Only the training images were used for optimizing C and γ.

We report the results using a measure frequently used in steganalysis – the minimum average classification error (2.1.3), where P_{FA} and P_{MD} are the false-alarm and missed-detection probabilities measured w.r.t. the testing set of images. Smaller values of P_{E} correspond to better steganalysis and thus larger statistical detectability (worse security).

2.3.1 SPAM Features for Grayscale Spatial Domain Digital Images

The main idea behind the SPAM feature set (acronym for Subtractive Pixel Adjacency Matrix) is to model dependencies among neighboring pixels without being influenced by the image content. It is well known that the values of neighboring pixels in natural images are not independent. This is not only caused by the inherent smoothness of natural images, but also by the image processing (de-mosaicking, sharpening, etc.) in the image acquisition device. This processing makes the noise, which is independent in the raw sensor output, dependent in the final image. The latter source of dependencies is very important for steganalysis because steganographic changes try to hide themselves within the image noise.

The SPAM features [93] model dependencies between neighboring pixels by means of higher-order Markov chains. They have been designed to provide a low-dimensional model of image noise that

[5] CCPev stands for 548-dimensional cartesian-calibrated version of Pevný's 274-dimensional merged features [95].

can be used for steganalytic purposes. The calculation of differences can be viewed as an application of high-pass filtering, which effectively suppresses the image content and exposes the noise. The success of SPAM features in detecting a wide range of steganographic algorithms [79] suggests that this model is reasonable for steganalysis and steganography.

The SPAM features model the transition probabilities between neighboring pixels along 8 directions $\{\leftarrow, \rightarrow, \downarrow, \uparrow, \nwarrow, \searrow, \swarrow, \nearrow\}$. Below, the calculation of the features is explained on the horizontal left-to-right direction, because for the other directions the calculations differ only by different indexing. All direction-specific variables are denoted by a superscript showing the direction.

Let $\mathbb{Z} = (z_{i,j}) \in \mathcal{X}$ be an image at hand (either cover or stego) of size $n_1 \times n_2$ pixels represented in a matrix form. The calculation starts by computing the difference array $\mathbb{D}^{\bullet} = (d^{\bullet}_{i,j})$, which is for a horizontal left-to-right direction

$$d^{\rightarrow}_{i,j} = z_{i,j} - z_{i,j+1},$$

for $i \in \{1, \dots, n_1\}$, $j \in \{1, \dots, n_2 - 1\}$. Here we describe a specific version of features based on the second-order Markov process with transition-probability array $\mathbb{M}^{\rightarrow} = (m^{\rightarrow}_{k_1,k_2,k_3})$ defined as,

$$\begin{aligned} m^{\rightarrow}_{k_1,k_2,k_3} &= Pr(d^{\rightarrow}_{i,j+2} = k_1 | d^{\rightarrow}_{i,j+1} = k_2, d^{\rightarrow}_{ij} = k_3) \\ &= \frac{n_1(n_2-2)\left|\{(i,j)|d^{\rightarrow}_{i,j+2} = k_1 \wedge d^{\rightarrow}_{i,j+1} = k_2 \wedge d^{\rightarrow}_{i,j} = k_3\}\right|}{n_1(n_2-3)\left|\{(i,j)|d^{\rightarrow}_{i,j+1} = k_2 \wedge d^{\rightarrow}_{i,j} = k_3\}\right|}. \end{aligned}$$

To reduce the number of features, we restrict only to the central portion of $\mathbb{M}^{\rightarrow}$ corresponding to $k_1, k_2, k_3 \in \{-T, \dots, T\}$ for some small $T \in \mathbb{N}$. The calculation of the features is finished by separate averaging of the horizontal and vertical arrays and the diagonal arrays to form the final feature sets. With a slight abuse of notation, the final feature vector $\mathbf{f} = (f_1, \dots, f_{2(2T+1)^3}) \in \mathbb{R}^{2(2T+1)^3}$ can be written as

$$f_k = \begin{cases} \frac{1}{4}\left(m^{\rightarrow}_{k_1,k_2,k_3} + m^{\leftarrow}_{k_1,k_2,k_3} + m^{\downarrow}_{k_1,k_2,k_3} + m^{\uparrow}_{k_1,k_2,k_3}\right) & \text{if } k \leq (2T+1)^3 \\ \frac{1}{4}\left(m^{\rightarrow}_{k_1,k_2,k_3} + m^{\leftarrow}_{k_1,k_2,k_3} + m^{\downarrow}_{k_1,k_2,k_3} + m^{\uparrow}_{k_1,k_2,k_3}\right) & \text{otherwise,} \end{cases}$$

where $k \in \{1, \dots, 2(2T+1)^3\}$ uniquely indexes 2 sets of triples $\{(k_1, k_2, k_3)| - T \leq k_1, k_2, k_3 \leq T\}$. We follow the authors' recommendation and use $T = 3$ leading to 686 features.

2.3.2 CCPev Features for Digital Images in JPEG Format

Among many other feature sets [109, 55, 15], Cartesian Calibrated (CCPev) version of Pevný's merged features [95] (denoted here as Pev) achieves stable performance in detecting different embedding algorithms [77, 79] in JPEG images. Originally, the 274-dimensional Pev features were designed by merging features capturing different dependencies one may find among DCT coefficients in a JPEG image and which are disturbed by embedding. The feature set contains elements describing first-order statistics, such as histograms, second-order statistics describing inter- and intra-block dependencies using sample transition probability matrices and quantities calculated from spatial-domain representation of the image, such as the blockiness.

Although the quantities change considerably after embedding, they have high variance on clean cover images and thus it is hard to separate the set of cover from stego features. The reason for this is the cover image content which increases the variance. To solve this problem, the process of calibration [35], where the same quantities are calculated from a slightly cropped image, allows us to differentiate embedding changes from innocuous image content. In the original implementation [35], this calibration process was implemented by calculating differences between quantities calculated from the original and the cropped image. Later [77] the difference was replaced by a Cartesian product giving the machine-learning algorithm freedom in choosing the best transformation. This results in $2 \cdot 274 = 548$ features — the CCPev.

Part I

The Square-Root Law of Imperfect Steganography

Thanks to the Central Limit Theorem,
the more covertext we give the Warden,
the better he will be able to estimate its statistics,
and so the smaller the rate at which [Alice] will be able to tweak bits safely.
The rate might even tend to zero...

— ROSS ANDERSON, (1996)

Chapter 3

The Square-Root Law of Imperfect Stegosystems

In the first part of this dissertation, we study the very fundamental relationship between the size of the secure payload and the size of the cover objects used by Alice and Bob. This study allows us to advise Alice and Bob on how fast they need to increase the size of the cover objects in order to increase the payload in their imperfect stegosystem. We describe and prove the so-called Square-Root Law (SRL) of imperfect stegosystems for sources with locally-dependent cover elements. The SRL is further used for comparison of stegosystems in Chapter 4. The work presented in this part is based on our own research [24, 25, 32].

This chapter is structured as follows. The original motivation for the SRL is described in Section 3.1. In Section 3.2, we present a very simple proof of the SRL of imperfect stegosystems with cover sources emitting i.i.d. bits. This section gives us a guideline we will follow when proving the SRL for cover sources in the form of a Markov chain. Section 3.3 describes the assumptions and their relationship to known stegosystems under which we will prove the SRL. Since the theorem deals exclusively with imperfect stegosystems, we characterize the set of all possible cover sources which form a perfectly secure stegosystem with a given embedding algorithm in Section 3.4. This analysis allows us to state and prove the main theorem of this part, the SRL of imperfect stegosystems with Markov covers. To improve the flow of arguments in the proof of the SRL in Section 3.5, several important but rather technical results were formulated as lemmas and moved to Appendix A. We finish this chapter in Section 3.6 by discussing the results and their direct consequences to steganography and steganalysis.

3.1 Introduction

It is a well-established fact that the maximal secure payload Alice can embed using perfectly secure stegosystem equals the entropy of the cover source she is using. Methods able to achieve this, at least in theory, can be constructed by following the cover-synthesis approach described in Section 2.2. For most cover sources of interest, the entropy (and thus the secure payload) scales linearly w.r.t. the size of the objects. From this we can conclude that perfectly secure stegosystems are able to achieve positive communication rates [16, 87], a fact that is common to many communication and compression problems in information theory.

In view of the absence of provably secure steganographic methods for empirical cover sources, it makes sense to investigate steganographic capacity of imperfect embedding methods for which detectors exist and inquire about the largest payload that can be embedded using their ϵ-secure versions in the sense of Cachin [11].

The fact that the secure payload is most likely sublinear was already suspected by Anderson [1] in 1996:

> “Thanks to the Central Limit Theorem, the more covertext we give the Warden, the

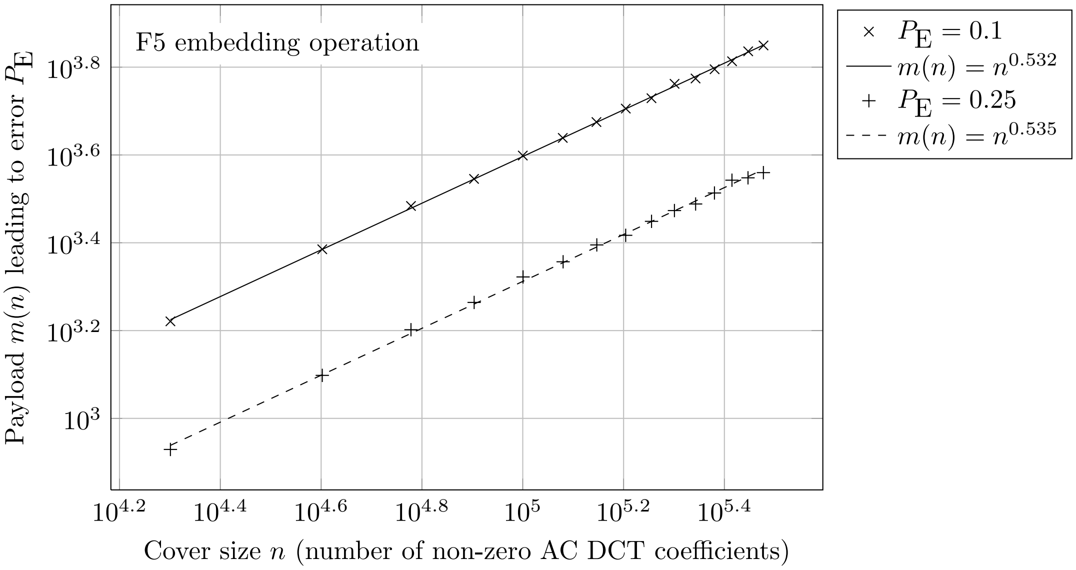

Figure 3.1.1: The largest payload $m(n)$ embedded using the embedding operation of F5 that produces a fixed steganalyzer error, P_{E}, for images with n non-zero AC DCT coefficients. The straight lines are corresponding linear fits. The slope of the lines is 0.53 and 0.54, which is in good agreement with the Square-root law.

> better he will be able to estimate its statistics, and so the smaller the rate at which [Alice] will be able to tweak bits safely. The rate might even tend to zero...”

Recent analysis of batch steganography and pooled steganalysis by Ker [67] tells us that the secure payload in imperfect stegosystems only grows as the square root of the number of communicated covers. This result could be interpreted as the asymptotic law describing the secure payload in a single image by dividing it into smaller blocks. Ker's result, however, was obtained with the assumption that the individual images (blocks) form a sequence of independent random variables, which is clearly false not only for images but also other digital media files. Our goal here is to study the asymptotic law for the simplest form of dependence that enables analytical reasoning—we assume that individual elements of the cover follow stationary Markov chain. The reason why we expect that the SRL will hold is its experimental verification [73] for various embedding methods in both spatial and DCT domains. In particular, the maximal payload that leads to the same fixed detection accuracy of the steganalyzer is proportional to the square root of the cover size. A sample result of these experiments on JPEG images for the embedding operation of F5 is reprinted in Figure 3.1.1. There, the accuracy of the detector is represented using the P_{E} error (**??**). For each set of images with a given number of non-zero AC DCT coefficients, n, the largest payload, $m(n)$, was iteratively found for which the steganalyzer obtained a fixed value of P_{E}. A linear fit through the experimental data displayed in a log-log plot confirms the SRL.

To study the SRL, we assume the worst possible interpretation (for Alice) of the Kerckhoffs' principle. We assume the warden is not only familiar with the embedding algorithm, but has complete knowledge of the cover source distribution. This may seem to be an unreasonably strong assumption, which may be impossible to achieve when the warden is working with empirical covers. It will turn out later, that Alice falls under the SRL whenever Eve is able to construct a non-trivial detector — a case for which the complete knowledge of the cover distribution is not necessary.

3.2 The SRL of Stegosystems with Independent Cover Sources

Perhaps the simplest way of explaining and understanding the reason for the form of the asymptotic law is to assume a stegosystem where cover objects are n-bit binary vectors i.i.d. following a

Bernoulli(p) distribution, i.e., $Pr(X_i = 1) = p$, $0 < p < 1$. The only way for Alice to be perfectly secure is to emit stego objects sampled from the same distribution. Since we are interested in imperfect stegosystems, we assume that Alice replaces each cover bit by a random bit obtained according to Bernoulli(q) distribution with $q \in [0, 1]$, and in order to be imperfect, $p \neq q$. Singular cases of $p = 0$ or $p = 1$ are not of our interest since any detector looking for the missing symbols will be able to detect any communication.

Since Alice does not know p and $q \neq p$, she will embed her payload by replacing βn bits in n-bit cover, for some $0 \leq \beta \leq 1$, i.e., she follows the cover-modification strategy. First, assume that Alice does not use any form of source coding (for example the matrix embedding algorithm discussed in Section 2.2.3) and thus, based on the stego key, she selects βn cover bits pseudo-randomly and replaces them with her stego bits. By doing so, she can embed up to $\beta n h(q)$ bits. Later, we discuss the version of the SRL when Alice is allowed to use the source coding.

Stego objects are n-bit vectors i.i.d. according to Bernoulli($(1-\beta)p + \beta q$) distribution. Let $(\beta_n)_{n=1}^{\infty}$ be a sequence describing Alice's strategy for setting the value of parameter β when embedding in n-bit covers. By (2.1.5), Alice should choose $(\beta_n)_{n=1}^{\infty}$ such that the KL divergence between n-bit cover distribution, $P^{(n)}$, and n-bit stego distribution $Q_{\beta_n}^{(n)}$ embedded with parameter β_n, $D_{\mathrm{KL}}(P^{(n)}||Q_{\beta_n}^{(n)})$, tends to zero in order to be sure that no possible detectors can be constructed for large n. Since the bits are independent,

$$\begin{aligned}
D_{\mathrm{KL}}\left(P^{(n)}||Q_{\beta_n}^{(n)}\right) &= nD_{\mathrm{KL}}\left(P^{(1)}||Q_{\beta_n}^{(1)}\right) \\
&= n\left((1-p)\ln\frac{1-p}{1-(1-\beta_n)p-\beta_n q} + p\ln\frac{p}{(1-\beta_n)p+\beta_n q}\right) \\
&= -n\left((1-p)\ln\left(1-\beta_n\frac{q-p}{1-p}\right) + p\ln\left(1-\beta_n\frac{p-q}{p}\right)\right) \\
&= n\beta_n^2\frac{(p-q)^2}{2(1-p)p} + n\sum_{i=3}^{\infty}\frac{\beta_n^i}{i}\left(\frac{(q-p)^i}{(1-p)^{i-1}} + \frac{(p-q)^i}{p^{i-1}}\right),
\end{aligned}$$

where we used the Taylor expansion $\ln(1-x) = -\sum_{i=1}^{\infty} x^i/i$ valid for small enough β_n. This shows the main reason for the form of the law, since $D_{\mathrm{KL}}(P^{(n)}||Q_{\beta_n}^{(n)})$ converges to 0 as n tends to ∞ if and only if $n\beta_n^2 \to 0$ which forces Alice to decrease β_n (and thus the relative payload) faster than $1/\sqrt{n}$.

If she keeps decreasing β_n slower than $1/\sqrt{n}$ (for example keeps β_n constant), a simple detector counting the number of bits in the observed bit string would be able to achieve arbitrarily small P_{FA} and P_{MD} errors. Consider $T(\mathbf{z}) = \sum_{i=1}^{n} z_i$ to be a test statistic and without loss of generality $p < q$. The test classifying $\mathbf{z} \in \{0,1\}^n$ as cover if $T(\mathbf{z}) < np + n\beta_n(p-q)/2$ and stego otherwise achieves the following errors

$$\begin{aligned}
P_{\mathrm{FA}} = Pr\left(T(\mathbf{X}) \geq np + \frac{n\beta_n(p-q)}{2}\right) &\leq Pr\left(|T(\mathbf{X}) - np| \geq \frac{n\beta_n(p-q)}{2}\right) \\
\leq \frac{4Var(T(\mathbf{X}))}{(n\beta_n(p-q))^2} = \frac{1}{n\beta_n^2}\frac{4p(1-p)}{(p-q)^2}
\end{aligned}$$

$$\begin{aligned}
P_{\mathrm{MD}} = Pr\left(T(\mathbf{Y}) < np + \frac{n\beta_n(p-q)}{2}\right) &\leq Pr\left(|T(\mathbf{Y}) - np| \geq \frac{n\beta_n(p-q)}{2}\right) \\
\leq \frac{4Var(T(\mathbf{Y}))}{(n\beta_n(p-q))^2} = \frac{1}{n\beta_n^2}\frac{4(p+\beta_n(q-p))(1-p+\beta_n(q-p))}{(p-q)^2}
\end{aligned}$$

which converge to zero as $n\beta_n^2 \to \infty$. This allows us to summarize the derivations of the first SRL theorem for i.i.d. binary covers.

Theorem 3.1 (The Square-root law for binary i.i.d. covers)**.** *Let $(S_n)_{n=1}^{\infty}$ be a sequence of stegosystems each with n-bit binary cover source i.i.d. according to Bernoulli(p) distribution and replacing $\beta_n n$ bits when embedding a message using i.i.d. Bernoulli(q) trials with $p \neq q$. The following holds:*

1. *If the sequence of embedding parameters* $(\beta_n)_{n=1}^{\infty}$ *increases faster than* $1/\sqrt{n}$ *in the sense that* $\lim_{n\to\infty} \frac{\beta_n}{1/\sqrt{n}} = \infty$*, then, for sufficiently large* n*, the warden can construct detectors achieving arbitrarily small* P_{FA} *and* P_{MD} *errors.*

2. *If* β_n *increases slower than* $1/\sqrt{n}$*,* $\lim_{n\to\infty} \frac{\beta_n}{1/\sqrt{n}} = 0$*, then the stegosystem can be made* ε*-secure for any* $\varepsilon > 0$ *for sufficiently large* n*.*

3. *Finally, if* β_n *grows as fast as* $1/\sqrt{n}$*,* $\lim_{n\to\infty} \frac{\beta_n}{1/\sqrt{n}} = \epsilon$ *for some* $0 < \epsilon < \infty$*, then the stegosystem is asymptotically* $C\epsilon^2$*-secure for some constant* C*.*

If Alice is allowed to use the best possible source coding algorithm, then she communicates the payload not only by changing βn cover elements, but also by choosing which one to change out of n possible. There are

$$\sum_{i=0}^{\lfloor \beta_n n \rfloor} \binom{n}{i} \le 2^{h(\beta_n)n}$$

possibilities of how to modify at most $\lfloor \beta_n n \rfloor$ cover bits out of n and thus up to $h(\beta_n)n$ bits can be communicated when all possibilities are utilized. Since $h(\beta_n)n$ is dominated by $\beta_n n \log_2 \beta_n$ for small β_n, the order of $\sqrt{n}$ from Theorem 3.1 changes to $\sqrt{n}\log_2 n$ possible bits—a choice which is still sublinear. Simple Hamming codes described in Section 2.2.3 achieve this rate.

3.3 Basic Assumptions

In the rest of this chapter, we formulate and prove the SRL for cover sources, which can be modeled by a stationary Markov chain allowing us to model simple dependencies between cover elements. We also generalize the form of stegosystems by increasing the size of the alphabet. We first formulate and discuss three basic assumptions under which we prove the SRL. The first assumption concerns the impact of embedding. We postulate that the stego object is obtained by applying a mutually independent embedding operation to each cover element. This type of embedding can be found in majority of practical embedding methods (see, e.g., [47] and the references therein). The second assumption is our model of covers. We require the individual cover elements to form a first-order Markov chain because this model is analytically tractable while allowing study of more realistic cover sources with memory. Finally, the third assumption essentially states that the steganographic method is not perfectly secure.

A stegosystem is a triple $S_n = (\mathbf{X}_1^n, \Phi^{(n)}, \Psi^{(n)})$ consisting of the random variable describing the cover source, embedding mapping $\Phi^{(n)}$, and extraction mapping $\Psi^{(n)}$. The embedding mapping $\Phi^{(n)}$ applied to $\mathbf{X}_1^n$ induces another random variable $\mathbf{Y}_1^n \triangleq (Y_1, \dots, Y_n)$ with probability distribution $Q_\beta^{(n)}$ over $\mathcal{X} \triangleq \mathcal{I}^n \triangleq \{1, \dots, N\}^n$. Here, $\beta \ge 0$ is a scalar parameter of embedding whose meaning will be explained shortly. The specific details of the embedding (and extraction) mappings are immaterial for our study. We only need to postulate the probabilistic *impact* of embedding.

Assumption 1 (Mutually independent embedding)**.** *The embedding algorithm visits every cover element* X_k *and modifies it to a corresponding element of the stego object* Y_k *with probability*

$$Q_\beta(Y_k = j | X_k = i) \triangleq b_{i,j}(\beta) = \begin{cases} 1 + \beta c_{i,i} & \text{if } i = j \\ \beta c_{i,j} & \text{otherwise,} \end{cases} \tag{3.3.1}$$

for some constants $c_{i,j} \ge 0$ *for* $i \ne j$*. Note that because* $\sum_{j=1}^{N} b_{i,j} = 1$*, we must have* $c_{i,i} = -\sum_{j\ne i} c_{i,j}$ *for each* $i \in \mathcal{I}$*. The matrix* $\mathbb{C} \triangleq (c_{i,j})$ *reflects the inner workings of the embedding algorithm, while the parameter* β *captures the* extent *of embedding changes. It will be useful to think of* β *as the relative number of changes (change rate) or some function of the change rate. Also note that we can find sufficiently small* β_0 *such that* $b_{i,i}(\beta) > 0$ *for* $\beta \in [0, \beta_0]$ *and all* $i \in \mathcal{I}$*. In matrix form,* $\mathbb{B}_\beta \triangleq (b_{i,j}(\beta)) = \mathbb{I} + \beta\mathbb{C}$*, where* $\mathbb{I}$ *is identity matrix of appropriate size.*

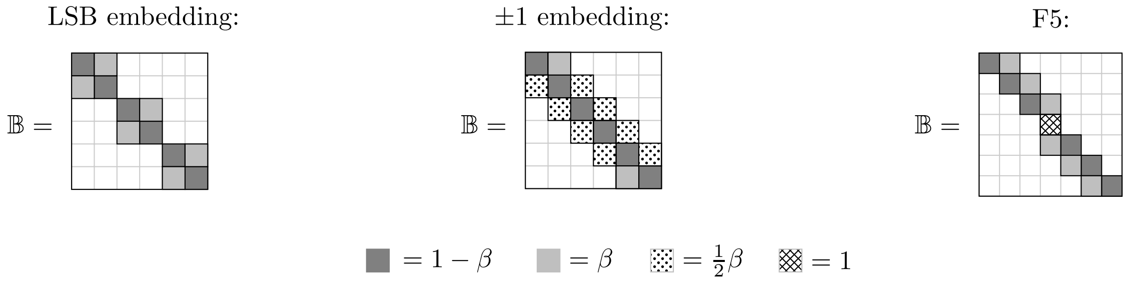

Figure 3.3.1: Examples of several embedding methods in the form of a functional matrix $\mathbb{B}$.

Because the matrix $\mathbb{B}_\beta$ does not depend on pixel location $k \in \{1, \dots, n\}$ or the history of embedding changes, one can say that the stego object is obtained from the cover by applying to each cover element a Mutually Independent embedding operation (we speak of *MI embedding*). The independence of embedding modifications implies that the conditional probability of stego object given the cover object can be factorized, i.e., $Q_\beta^{(n)}(\mathbf{Y}_1^n|\mathbf{X}_1^n) = \prod_{i=1}^n Q_\beta(Y_i|X_i)$.

Many embedding algorithms across different domains use MI embedding. Representative examples are LSB embedding, ±1 embedding, stochastic modulation [39], Jsteg, MMx [74], and various versions of the F5 algorithm [47]. Examples of matrix $\mathbb{B}_\beta$ for three selected embedding methods are shown in Figure 3.3.1.

Next, we formulate our assumption on the cover source.

Assumption 2 (Markov cover source)**.** *We assume the cover source $\mathbf{X}_1^n$ is a first-order stationary Markov Chain (MC) over $\mathcal{I} \triangleq \{1, \dots, N\}$, to which we will often refer as just Markov chain for brevity. This source is completely described by its stochastic transition probability matrix $\mathbb{A} \triangleq (a_{i,j}) \in \mathbb{R}^{N \times N}$, $a_{i,j} = Pr(X_k = j|X_{k-1} = i)$, and by the initial distribution $Pr(X_1)$. The probability distribution induced by the MC source generating n-element cover objects satisfies $P^{(n)}(\mathbf{X}_1^n = \mathbf{x}_1^n) = P^{(n-1)}(\mathbf{X}_1^{n-1} = \mathbf{x}_1^{n-1})a_{x_{n-1},x_n}$, where $P^{(1)}(X_1)$ is the initial distribution. We further assume that the transition probability matrix of the cover source satisfies $a_{i,j} \geq \delta > 0$, for some δ and thus the MC is irreducible. The stationary distribution of the MC source is a vector $\boldsymbol{\pi} \triangleq (\pi_1, \dots, \pi_N)$ satisfying $\boldsymbol{\pi}\mathbb{A} = \boldsymbol{\pi}$. In this chapter, we will always assume that the initial distribution $P^{(1)}(X_1) = \boldsymbol{\pi}$, which implies $P^{(n)}(X_k) = \boldsymbol{\pi}$ for every n and k. This assumption simplifies the analysis without loss of generality because the marginal probabilities $P^{(n)}(X_k)$ converge to $\boldsymbol{\pi}$ with exponential rate w.r.t. k (see Doob [20], Equation (2.2) on page 173). In other words, MCs are "forgetting" their initial distribution with exponential rate.*

Under the above assumption and the class of MI embedding, the source of stego images no longer exhibits the Markov property and forms a Hidden Markov Chain (HMC) instead [111]. The HMC model is described by its hidden states (cover elements) and output transition probabilities (MI embedding). Hidden states are described by the cover MC and the output probability transition matrix $\mathbb{B}$ is taken from the definition of MI embedding.

Unless stated otherwise, in the rest of this chapter $Q_\beta^{(n)}$ denotes the probability measure induced by the HMC source embedded with parameter β into n-element MC cover objects. By the stationarity of the MC source, the marginal probabilities $P^{(n)}(\mathbf{X}_k^{k+1}) = P^{(2)}(\mathbf{X}_1^2)$ and $Q_\beta^{(n)}(\mathbf{Y}_k^{k+1}) = Q_\beta^{(n)}(\mathbf{Y}_1^2)$ for all k. Sometimes we will omit the number of elements, n, and denote as P and Q_β the probability distribution over cover and stego images, respectively.

The third assumption we formulate concerns the entire stegosystem S_n which we require to be imperfect.

Assumption 3 (Imperfect stegosystem)**.** *We assume that the stegosystem $S_n = (\mathbf{X}_1^n, \Phi^{(n)}, \Psi^{(n)})$ is not perfectly secure in the sense of Cachin [11], i.e., the KL divergence $d(\beta) \triangleq D_{KL}(P^{(n)}||Q_\beta^{(n)}) > 0$.*

Finally, we would like to stress that Assumptions 1–3 are not overly restrictive and will likely be satisfied for all practical steganographic schemes in some appropriate representation of the cover.

For example, a stegosystem that preserves the Markov model is likely to be detectable by computing higher-order dependencies among pixels. Thus, the stegosystem will become imperfect when representing the cover as pairs or groups of pixels/coefficients or some other quantities computed from the cover.

3.4 Perfectly Secure Stegosystems with MI Embedding

Unfortunately, direct application of Assumption 3 to Markov cover sources and MI embedding requires further study which we present in this section. To simplify the language in this section, we will speak of security of a cover source w.r.t. a given MI embedding meaning that the *cover source is perfectly secure w.r.t.* $\mathbb{B}$, if the resulting stegosystem is perfectly secure. It does then make sense to inquire about all possible perfectly secure cover sources w.r.t. MI embedding with matrix $\mathbb{B}_\beta$. Such analysis is required to satisfy Assumption 3 for the proof of the SRL. Results presenting equivalent conditions for validating the Assumption 3 and summarized in Corollary 3.2 will be utilized in the proof.

Combination of MC cover source and MI embedding requires results from the theory of ergodic classes which we borrowed from [20]. We will apply them to the stochastic matrix $\mathbb{B}_\beta$. For states $i, j \in \mathcal{I}$, we call j *a consequent* of i (of order k) $(i \to j)$ iff $\exists k$, $(\mathbb{B}_\beta^k)_{i,j} \neq 0$. State $i \in \mathcal{I}$ is *transient* if it has a consequent of which it is not itself a consequent, i.e., $\exists j \in \mathcal{I}$ such that $(i \to j) \Rightarrow (j \not\to i)$. We say $i \in \mathcal{I}$ is *non-transient* if it is a consequent of every one of its consequents, $\forall j \in \mathcal{I}$, $(i \to j) \Rightarrow (j \to i)$. The set $\mathcal{I}$ can be decomposed as $\mathcal{I} = \mathcal{F} \cup \mathcal{E}_1 \cup \cdots \cup \mathcal{E}_k$, where $\mathcal{F}$ is the set of all transient states and $\mathcal{E}_a$, $a \in \{1, \ldots, k\}$, are so-called ergodic classes. We put two non-transient states into one ergodic class if they are consequents of each other.

Let matrix $\mathbb{B}_\beta$ have k ergodic classes. Then, there exist k linearly independent left eigenvectors with non-negative elements, denoted as $\boldsymbol{\pi}^{(1)}, \ldots, \boldsymbol{\pi}^{(k)}$, of matrix $\mathbb{B}_\beta$ corresponding to eigenvalue 1, called *invariant distributions*. If $\boldsymbol{\pi}^{(a)}\mathbb{B}_\beta = \boldsymbol{\pi}^{(a)}$, for some $a \in \{1, \ldots, k\}$, then $\pi_i^{(a)} > 0$ for all $i \in \mathcal{E}_a$, and $\pi_i^{(a)} = 0$ otherwise. Every other $\boldsymbol{\pi}$ with non-negative elements satisfying $\boldsymbol{\pi}\mathbb{B}_\beta = \boldsymbol{\pi}$ is obtained by a convex linear combination of $\{\boldsymbol{\pi}^{(a)} | a \in \{1, \ldots, k\}\}$. For a complete reference, see [20, Chapter V, §2]. The set of ergodic classes for matrix $\mathbb{B}_\beta$ depends only on the set $\{(i,j) | b_{i,j}(\beta) \neq 0\}$. Since $b_{i,j}(\beta) = 0$ iff $c_{i,j} = 0$ for $i \neq j$ and $b_{i,i}(\beta) > 0$ for $\beta \in (0, \beta_0]$, the structure of ergodic classes does not depend on β. Moreover, if $\boldsymbol{\pi}\mathbb{B}_\beta = \boldsymbol{\pi}$ for some $\beta > 0$, then $\boldsymbol{\pi}\mathbb{C} = 0$ and thus all invariant distributions are independent of β, because $\boldsymbol{\pi}\mathbb{B}_{\beta'} = \boldsymbol{\pi}\mathbb{I} + \beta'\boldsymbol{\pi}\mathbb{C} = \boldsymbol{\pi}\mathbb{I} = \boldsymbol{\pi}$. By this reason, we frequently omit the index β.

3.4.1 Perfectly Secure Cover Sources under MI Embedding Operation

In this section, we let matrix $\mathbb{B}$ represent an arbitrary MI embedding with k ergodic classes $\mathcal{E}_a$ and invariant distributions $\boldsymbol{\pi}^{(a)}$, $a \in \{1, \ldots, k\}$. The following example describes a construction of perfectly secure cover sources w.r.t. $\mathbb{B}$.

Example 3.1 (Perfectly secure cover sources)**.** Let $P^{(2)}$ be a probability distribution on 2-element cover objects defined as $P^{(2)}(\mathbf{X}_1^2 = (i,j)) = \pi_i^{(a)}\pi_j^{(b)}$ for some $a, b \in \{1, \ldots, k\}$. Then $P^{(2)}$ is a perfectly secure cover source w.r.t. $\mathbb{B}$ because

$$\begin{aligned} Q_\beta^{(2)}(\mathbf{Y}_1^2 = (i,j)) &= \left(\sum_{i'} b_{i',i} P(X_1 = i')\right)\left(\sum_{j'} b_{j',j} P(X_2 = j')\right) \\ &= \left(\boldsymbol{\pi}^{(a)}\mathbb{B}\right)_i \left(\boldsymbol{\pi}^{(b)}\mathbb{B}\right)_j = \pi_i^{(a)}\pi_j^{(b)} = P^{(2)}\left(\mathbf{X}_1^2 = (i,j)\right), \end{aligned}$$

and thus both distributions $P^{(2)}$, and $Q_\beta^{(2)}$ are identical, which implies perfect security. Since this construction does not depend on the particular choice of $a, b \in \{1, \ldots, k\}$, we can create k^2 perfectly secure cover sources w.r.t. $\mathbb{B}$. The probability distributions $P^{(2)}$ obtained from this construction are linearly independent and form a k^2-dimensional linear vector space. By a similar construction, we can construct k^n n-element linearly independent perfectly secure cover sources w.r.t. $\mathbb{B}$.

We next show that there are no other linearly independent perfectly secure cover sources w.r.t. $\mathbb{B}$.

Theorem 3.2 (Mutually independent embedding)**.** *There are exactly k^n linearly independent perfectly secure probability distributions P on n-element covers. Every perfectly secure probability distribution P w.r.t. $\mathbb{B}$ can be obtained by a convex linear combination of k^n linearly independent perfectly secure distributions described in Example 3.1.*

Proof. It is sufficient to prove that there cannot be more than k^n linearly independent perfectly secure probability distributions P on n-element covers. We show the proof for $n = 2$ and later present its generalization.

We define the following matrices $\mathbb{P} \triangleq (p_{i,j})$, $p_{i,j} = P(\mathbf{X}_1^2 = (i,j))$, and $\mathbb{Q} \triangleq (q_{i,j})$, $q_{i,j} = Q_\beta(\mathbf{Y}_1^2 = (i,j))$. By definition of MI embedding, we have

$$\begin{aligned} q_{ij} &= \sum_{(v,w)\in\mathcal{I}^2} Q_\beta\big(\mathbf{Y}_1^2 = (i,j)|\mathbf{X}_1^2 = (v,w)\big)P\big(\mathbf{X}_1^2 = (v,w)\big) \\ &= \sum_{v,w\in\mathcal{I}} b_{v,i}b_{w,j}p_{v,w}. \end{aligned}$$

Define matrix $\mathbb{D} \triangleq (d_{\mathbf{u}_1^2,\mathbf{v}_1^2})$ of size $N^2 \times N^2$, where $d_{\mathbf{u}_1^2,\mathbf{v}_1^2} = b_{u_1,v_1}b_{u_2,v_2}$. If $\mathbf{p}$ is defined as one big row vector of elements $p_{i,j}$ and similarly $\mathbf{q}$, then assuming perfect security of cover source w.r.t. $\mathbb{B}$ ($\mathbb{P} = \mathbb{Q}$), we have $\mathbf{q} = \mathbf{p}\,\mathbb{D} = \mathbf{p}$ and thus $\mathbf{p}$ is left eigenvector of $\mathbb{D}$ corresponding to 1. Matrix $\mathbb{D}$ is stochastic and thus it is sufficient to show that it has k^2 ergodic classes.

We first show that

$$\mathbf{u}_1^2 \overset{(m)}{\to} \mathbf{v}_1^2 \Leftrightarrow (u_1 \overset{(m)}{\to} v_1) \text{ and } (u_2 \overset{(m)}{\to} v_2),\quad \mathbf{u}_1^2,\mathbf{v}_1^2 \in \mathcal{I}^2. \tag{3.4.1}$$

By $\mathbf{u}_1^2 \overset{(m)}{\to} \mathbf{v}_1^2$ we mean that $\mathbf{v}_1^2$ is a consequent of $\mathbf{u}_1^2$ of order m in terms of matrix $\mathbb{D}$. If $\mathbf{u}_1^2 \overset{(m)}{\to} \mathbf{v}_1^2$, then there exist $m-1$ intermediate states ${}_1\mathbf{w}_1^2, \ldots, {}_{m-1}\mathbf{w}_1^2$, such that $d_{\mathbf{u},{}_1\mathbf{w}}d_{{}_1\mathbf{w},{}_2\mathbf{w}}\cdots d_{{}_{m-1}\mathbf{w},\mathbf{v}} > 0$. Since $d_{\mathbf{u}_1^2,\mathbf{v}_1^2} = b_{u_1,v_1}b_{u_2,v_2}$, this implies the existence of both paths $u_i \overset{(m)}{\to} v_i$ of order m, $i = 1,2$. The converse is true by the same reason.

We show that $\mathcal{E}_a \times \mathcal{E}_b$, $a,b \in \{1,\ldots,k\}$ are the only ergodic classes. If $u_1 \overset{(m_1)}{\to} v_1$ and $u_2 \overset{(m_2)}{\to} v_2$, then $\mathbf{u}_1^2 \overset{(m_1+m_2)}{\to} \mathbf{v}_1^2$ for all $u_1,v_1 \in \mathcal{E}_a$ and $u_2,v_2 \in \mathcal{E}_b$, because the path from u_i to v_i can be arbitrarily extended by adding self loops of type $j \to j$ since all diagonal terms $b_{j,j}$ are positive and thus by (3.4.1) we have $\mathbf{u}_1^2 \overset{(m_1+m_2)}{\to} \mathbf{v}_1^2$. Finally by $u_1,v_1 \in \mathcal{E}_a$ and $u_2,v_2 \in \mathcal{E}_b$, $v_i \to u_i$ and by the same argument $\mathbf{v}_1^2 \to \mathbf{u}_1^2$, and therefore $\mathcal{E}_a \times \mathcal{E}_b$ are ergodic classes. Any other state $\mathbf{u}_1^2 \in (\mathcal{E}_a \times \mathcal{F}) \cup (\mathcal{F} \times \mathcal{E}_a) \cup (\mathcal{F} \times \mathcal{F})$ must be transient w.r.t. $\mathbb{D}$, otherwise by (3.4.1) we obtain contradiction with $u_i \in \mathcal{F}$ for some i.

This proof can be generalized for $n \geq 3$ by proper definition of matrices $\mathbb{P}$, $\mathbb{Q}$, and $\mathbb{D}$. In general, matrix $\mathbb{D}$ has size $N^n \times N^n$. By similar construction we obtain k^n ergodic classes of generalized matrix $\mathbb{D}$, however we know k^n linearly independent distributions. □

3.4.2 Perfect Security and Fisher Information

In this sub-section, we show that for stegosystems with MI embedding perfect security can be captured using Fisher information. From Taylor expansion of KL divergence, for small β,

$$d(\beta) = D_{\mathrm{KL}}\Big(P^{(n)}||Q_\beta^{(n)}\Big) = \frac{1}{2}\beta^2 \underbrace{\frac{d^2}{d\beta^2}d(\beta)\Big|_{\beta=0}}_{\triangleq I(0)} + O(\beta^3),$$

where $I(0)$ is the Fisher information w.r.t. β at $\beta = 0$. If for some stegosystem $d(\beta) = 0$ for $\beta \in [0,\beta_0]$, then $I(0) = 0$ from the Taylor expansion. Even though the opposite does not hold in general, we will prove that for MI embedding zero Fisher information implies perfect security. In other words, a stegosystem with MI embedding is perfectly secure for $\beta \in [0, \beta_0]$ if and only if $I(0) = 0$. This provides us with a simpler condition for verifying perfect security than the KL divergence. Fisher information also provides a connection to quantitative steganalysis because $1/I(\beta)$ is the lower bound on variance of unbiased estimators of β. Moreover, $I(0)$ will be used for comparing (benchmarking) stegosystems.

We start by reformulating the condition $I(0) = 0$.

Proposition 3.1. *Let P and Q_β be probability distributions of cover and stego objects with n elements embedded with parameter β. The Fisher information is zero if and only if the FI condition is satisfied*

$$\forall \mathbf{y}_1^n \in \mathcal{I}^n \quad \Big(P\big(\mathbf{X}_1^n = \mathbf{y}_1^n\big) > 0\Big) \Rightarrow \Big(\frac{d}{d\beta}Q_\beta\big(\mathbf{y}_1^n\big)\big|_{\beta=0} = 0\Big). \tag{3.4.2}$$

Proof. The second derivative of $d(\beta)$ at β, $d''(\beta)$, can be written as

$$I(\beta) = -\sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P(y_1^n)\left(\frac{Q''_\beta(\mathbf{y}_1^n)}{Q_\beta(\mathbf{y}_1^n)} - \left(\frac{Q'_\beta(\mathbf{y}_1^n)}{Q_\beta(\mathbf{y}_1^n)}\right)^2\right), \tag{3.4.3}$$

where $Q'_\beta(\mathbf{y}_1^n) = \frac{d}{d\beta}Q_\beta(\mathbf{y}_1^n)$. By $P(\mathbf{y}_1^n) = Q_{\beta=0}(\mathbf{y}_1^n)$, the first term in the bracket in (3.4.3) sums to zero at $\beta = 0$, and thus $I(0)$ is zero iff $Q'_\beta(\mathbf{y}_1^n)\big|_{\beta=0} = 0$ is zero for all $\mathbf{y}_1^n \in \mathcal{I}^n$ for which $P^{(n)}(\mathbf{y}_1^n) > 0$ as was to be proved. Here, we assume the KL divergence $d(\beta)$ to be continuous w.r.t. β which is valid by the construction of the matrix $\mathbb{B}$. □

The next theorem shows that the FI condition (3.4.2) is equivalent with perfect security for MI embedding.

Theorem 3.3 (Fisher information condition)**.** *There are exactly k^n linearly independent probability distributions P on n-element covers satisfying the FI condition (3.4.2). These distributions are perfectly secure w.r.t. $\mathbb{B}$. Every other probability distribution P satisfying (3.4.2) can be obtained by a convex linear combination of k^n linearly independent perfectly secure distributions.*

Proof. From Example 3.1, we know k^n linearly independent perfectly secure distributions. By Taylor expansion of $d(\beta)$, these distributions satisfy the FI condition, because $d(\beta) = 0 \Rightarrow I(0) = 0$. It is sufficient to show that there cannot be more linearly independent distributions satisfying the FI condition.

Similarly as in the previous proof, we reformulate the theorem as an eigenvector problem and use ergodic class theory to give the exact number of left eigenvectors corresponding to 1. Again, we present the proof for the case $n = 2$ and then show how to generalize it.

If P satisfies (3.4.2), then the linear term in the Taylor expansion of $Q_\beta(\mathbf{y}_1^2)$ w.r.t. β is zero. By the independence property, $(Q(\mathbf{y}_1^n|\mathbf{x}_1^n) = \prod_{i=1}^n Q(y_i|x_i))$, and the form of matrix $\mathbb{B}$ ($\mathbb{B}_\beta = \mathbb{I} + \beta\mathbb{C}$), condition (3.4.2) has the following form

$$\begin{aligned}\frac{d}{d\beta}Q_\beta(\mathbf{y}_1^2)\Big|_{\beta=0} &= \lim_{\beta\to 0}\sum_{\mathbf{x}_1^2 \in \mathcal{I}^2} P(\mathbf{x}_1^2)\frac{d}{d\beta}\prod_{i=1}^{2} Q_\beta(y_i|x_i)\\ &= \sum_{x_1 \in \mathcal{I}} c_{x_1,y_1}P(x_1, y_2) + \sum_{x_2 \in \mathcal{I}} c_{x_2,y_2}P(y_1, x_2) = 0.\end{aligned} \tag{3.4.4}$$

We define matrix $\mathbb{P} \triangleq (p_{i,j})$ as $p_{i,j} = P(\mathbf{X}_1^2 = (i,j))$ and represent it as a row vector $\mathbf{p}$. If we define matrix $\mathbb{D} \triangleq (d_{\mathbf{u}_1^2,\mathbf{v}_1^2})$ of size $N^2 \times N^2$ as

$$d_{\mathbf{u}_1^2,\mathbf{v}_1^2} = \begin{cases} c_{u_1,v_1} & \text{if } u_1 \neq v_1 \text{ and } u_2 = v_2\\ c_{u_2,v_2} & \text{if } u_1 = v_1 \text{ and } u_2 \neq v_2\\ 0 & \text{otherwise,}\end{cases} \tag{3.4.5}$$

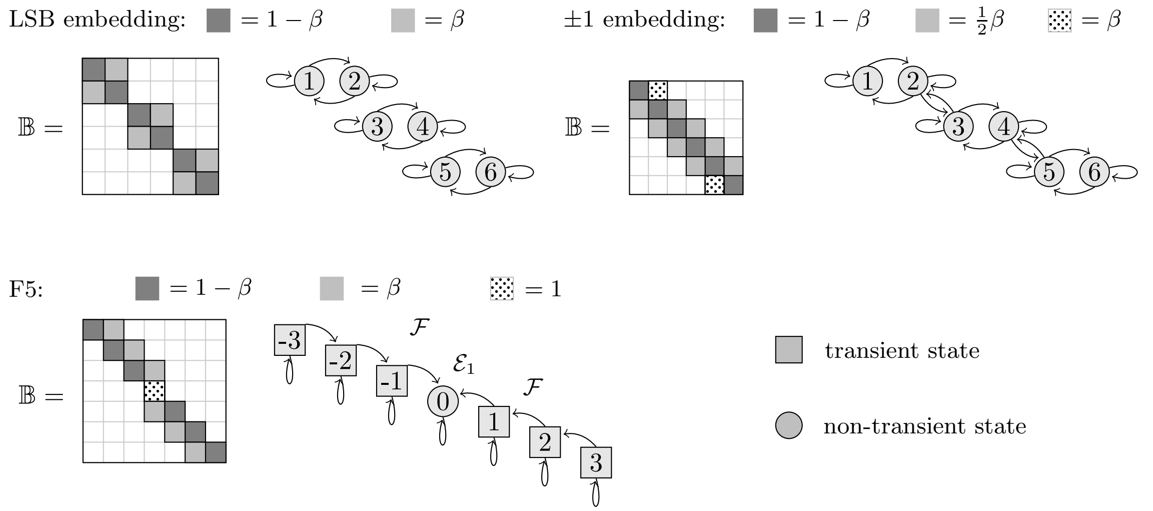

Figure 3.4.1: Examples of several embedding methods and their ergodic classes.

and diagonal matrix $\mathbb{G} \triangleq (g_{\mathbf{u}_1^2,\mathbf{v}_1^2})$ of size $N^2 \times N^2$ as $g_{\mathbf{u}_1^2,\mathbf{u}_1^2} = -c_{u_1,u_1} - c_{u_2,u_2}$, then Equation (3.4.4) can be written in a compact form as $\mathbf{p}\,\mathbb{D} = \mathbf{p}\,\mathbb{G}$. Both matrices $\mathbb{D}$ and $\mathbb{G}$ are non-negative by their definitions. Let $\mathbb{H} = \mathbb{I} + \gamma(\mathbb{D} - \mathbb{G})$ with $\mathbb{I}$ being identity matrix of appropriate size. If we put $\gamma = (\max_{\mathbf{u}_1^2 \in \mathcal{I}^2} g_{\mathbf{u}_1^2,\mathbf{u}_1^2})^{-1}$, then matrix $\mathbb{H}$ is stochastic and $\mathbf{p}\,\mathbb{H} = \mathbf{p}$ iff $\mathbf{p}\,\mathbb{D} = \mathbf{p}\,\mathbb{G}$ and thus (3.4.2) is equivalent with an eigenvalue problem for matrix $\mathbb{H}$.

First, we observe that for $i \neq j$ $c_{i,j} > 0$ iff $h_{(i,a),(j,a)} > 0$ for all $a \in \mathcal{I}$, because by (3.4.5) $h_{(i,a),(j,a)} = \gamma d_{(i,a),(j,a)} = \gamma c_{i,j}$ (the first case when $u_2 = v_2$). Similarly, for $i \neq j$ $c_{i,j} > 0$ iff $h_{(a,i),(a,j)} > 0$ for all $a \in \mathcal{I}$ (the second case when $u_1 = v_1$). This means that $i \to j$ iff $(i,a) \to (j,a)$ w.r.t. $\mathbb{H}$ for all $a \in \mathcal{I}$ and similarly $i \to j$ iff $(a,i) \to (a,j)$ w.r.t. $\mathbb{H}$ for all $a \in \mathcal{I}$. This can be proved by using the previous statement. By this rule used for a given $\mathbf{u}_1^2 \in \mathcal{E}_a \times \mathcal{E}_b$, we obtain $\mathbf{u}_1^2 \to \mathbf{v}_1^2$ and $\mathbf{v}_1^2 \to \mathbf{u}_1^2$ for all $\mathbf{v}_1^2 \in \mathcal{E}_a \times \mathcal{E}_b$ and thus $\mathcal{E}_a \times \mathcal{E}_b$ is an ergodic class w.r.t. $\mathbb{H}$. We show that there can not be more ergodic classes and thus we have all k^2 of them. If $\mathbf{u}_1^2 \in \mathcal{F} \times \mathcal{E}$, then $\mathbf{u}_1^2$ has to be transient w.r.t. $\mathbb{H}$, otherwise we will obtain contradiction with $u_1 \in \mathcal{F}$. This is because the only consequents of order 1 are of type $(i,a) \to (j,a)$ or $(a,i) \to (a,j)$, therefore if $\mathbf{u}_1^2 \in \mathcal{F} \times \mathcal{E}$, we choose $\mathbf{v}_1^2 \in \mathcal{I} \times \mathcal{E}$, such that $v_1 \not\to u_1$ (u_1 is transient and thus such v_1 must exist). State $\mathbf{u}_1^2$ must be transient otherwise $\mathbf{u}_1^2 \leftrightarrow \mathbf{v}_1^2$ implies $u_1 \leftrightarrow v_1$ which results in contradiction with $v_1 \not\to u_1$. Similarly for $\mathbf{u}_1^2 \in \mathcal{E} \times \mathcal{F} \cup \mathcal{F} \times \mathcal{F}$. This proof can be generalized for $n \geq 3$ by assuming larger matrices $\mathbb{P}$, $\mathbb{D}$, $\mathbb{G}$, and $\mathbb{H}$, obtaining exactly k^n linearly independent perfectly secure distributions satisfying the FI condition. □

Next, we discuss the structure of the set of invariant distributions for a given MI embedding and show how to find ergodic classes from matrix $\mathbb{B}$ in practice. By Theorem 2.1 from [20, Chapter V, page 175], this can be done by inspecting the matrix limit $\mathbb{M} = (m_{i,j}) = \lim_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{B}^i$. According to this theorem, state i is non-transient iff $m_{i,i} > 0$ and is transient otherwise. We put two non-transient states $i, j \in \mathcal{I}$ into one ergodic class if $m_{i,j} > 0$. All rows of the matrix $\mathbb{M}$ corresponding to states in one ergodic class $\mathcal{E}_a$ are the same and equal to the invariant distribution of this class, $\boldsymbol{\pi}^{(a)}$.

This sub-section is closed with a short discussion of two practical embedding algorithms. For the F5 embedding algorithm [125], the set of states $\mathcal{I} = \{-1024, \ldots, 1024\}$. By the nature of the embedding changes (flip towards 0), there is only one ergodic set $\mathcal{E}_1 = \{0\}$ and $\mathcal{F} = \mathcal{I} \setminus \{0\}$. Thus, there is only one invariant distribution, $\pi_0 = 1$ and zero otherwise. Obviously, no message can be embedded in covers with this singular distribution.

For the case of LSB embedding over $\mathcal{I} = \{0, \ldots, 255\}$, we have $\mathcal{E}_a = \{2a, 2a+1\}$ for $a \in \{0, \ldots, 127\}$, $\mathcal{F} = \{\}$ and $\pi_{2a}^{(a)} = \pi_{2a+1}^{(a)} = 1/2$ and zero otherwise (LSB embedding cannot be detected in images with evened out histogram bins). Thus, sources realized as a sequence of mutually independent random variables with such a distribution are the only perfectly secure sources w.r.t. LSB embedding. Figure 3.4.1 shows examples of matrices $\mathbb{B}$ and ergodic classes of several known algorithms with MI embedding operation.

3.4.3 Application to Markov cover sources

In this section, we reformulate the results obtained so far for a special type of cover sources that can be modeled as first-order stationary Markov Chains (MC). The results play a key role in proving the SRL of steganographic capacity of imperfect stegosystems for Markov covers and substitute Assumption 3.

First, for stationary cover sources Theorem 3.2 leads to this immediate corollary.

Corollary 3.1. *There are exactly k (instead of k^n) linearly independent perfectly secure stationary cover sources. These sources are i.i.d. with some invariant distribution $\boldsymbol{\pi}_a$, $a \in \{1, \ldots, k\}$.*

The next corollary states that in order to study perfect security of n-element stationary MC covers, it is enough to study only 2-element covers.

Corollary 3.2. *Let P be a first-order stationary MC cover distribution and Q_β its corresponding stego distribution after MI embedding with parameter β. For a given $n \geq 2$, the following statements are equivalent.*

1. *An n-element stegosystem is not perfectly secure.*
2. *Corresponding stegosystem narrowed to 2-element cover source is not perfectly secure:*

$$\forall \beta > 0,\ \exists \mathbf{y}_1^2 \in \mathcal{I}^2 \quad P^{(2)}\big(\mathbf{X}_1^2 = \mathbf{y}_1^2\big) \neq Q_\beta^{(2)}\big(\mathbf{X}_1^2 = \mathbf{y}_1^2\big). \tag{3.4.6}$$

3. *The pair $(P^{(2)}, Q_\beta^{(2)})$ does not satisfy the FI condition,*

$$\forall \mathbf{y}_1^2 \in \mathcal{I}^2 \quad \Big(P^{(2)}(\mathbf{X}_1^2 = \mathbf{y}_1^2) > 0\Big) \quad \Rightarrow \quad \left(\frac{d}{d\beta} Q_\beta^{(2)}(y_1^2)\big|_{\beta=0} = 0\right). \tag{3.4.7}$$

Proof. We prove the equivalences in the order (1)⇒(2)⇒(3)⇒(1) by contradiction. For (1)⇒(2), assume 2-element stegosystem is perfectly secure. According to Theorem 3.2, this implies that the cover source is i.i.d. according to some invariant distribution which contradicts (1) since the stegosystem extended to n elements with stationary cover source distribution must be perfectly secure as well. For (2)⇒(3), assume the FI condition does hold. Then, by Theorem 3.3, the 2-element stegosystem must be perfectly secure which contradicts (2). Similarly for (3)⇒(1), if n-element stegosystem is perfectly secure, then Fisher information is zero which would contradict (3). This completes the proof since 2-element marginal is sufficient statistics for a first-order stationary MC. □

3.5 The SRL for Markov Cover Sources

In this section, we formulate and prove the main result of this chapter, which states that the secure payload of imperfect stegosystems with Markov covers and MI embedding only grows with the square root of the number of cover elements.

For the formulation of the SRL theorem, we borrow the term used in [67, Cor. 7]. We will say that the Steganographer is *at risk* (w.r.t. some fixed tuple $(P_{\mathrm{FA}}^*, P_{\mathrm{MD}}^*)$, with $0 < P_{\mathrm{FA}}^* < 1$ and $0 < P_{\mathrm{MD}}^* < 1 - P_{\mathrm{FA}}^*$) if the warden has a detector with probability of false alarms and missed detection $P_{\mathrm{FA}}, P_{\mathrm{MD}}$ satisfying $P_{\mathrm{FA}} < P_{\mathrm{FA}}^*$ and $P_{\mathrm{MD}} < P_{\mathrm{MD}}^*$.

Theorem 3.4 (The Square-root law of imperfect stegosystems with Markov covers)**.** *For the sequence of stegosystems $(S_n)_{n=1}^{\infty}$ satisfying Assumptions 1–3, the following holds:*

1. *If the sequence of embedding parameters β_n increases faster than $1/\sqrt{n}$ in the sense that $\lim_{n\to\infty}\frac{\beta_n}{1/\sqrt{n}}=\infty$, then, for sufficiently large n, the Steganographer is at risk for arbitrary tuple (P^*_{FA},P^*_{MD}).*
2. *If β_n increases slower than $1/\sqrt{n}$, $\lim_{n\to\infty}\frac{\beta_n}{1/\sqrt{n}}=0$, then the stegosystem can be made ε-secure for any $\varepsilon>0$ for sufficiently large n.*
3. *Finally, if β_n grows as fast as $1/\sqrt{n}$, $\lim_{n\to\infty}\frac{\beta_n}{1/\sqrt{n}}=\epsilon$ for some $0<\epsilon<\infty$, then the stegosystem is asymptotically $C\epsilon^2$-secure for some constant C.*

Proof. We prove each part of the theorem separately. We remind that under the Kerckhoffs' principle, Eve knows the distribution of cover images $P^{(n)}=Q_0^{(n)}$.
Part 1 [Steganographer at risk] Here, we prove that the Steganographer is at risk w.r.t. any (P^*_{FA},P^*_{MD}) for all sufficiently large n. This means that we need to construct a sequence of detectors, D_n, for the following composite binary hypothesis testing problem

$$\begin{aligned}\mathrm{H}_0 &:\ \beta=0\\ \mathrm{H}_1 &:\ \beta>0\end{aligned}$$

based on observing one stego image (one realization of a random sequence with distribution $Q_\beta^{(n)}$). The error probabilities of these detectors are required to satisfy $P_{\mathrm{FA}}<P^*_{\mathrm{FA}}$ and $P_{\mathrm{MD}}<P^*_{\mathrm{MD}}$ for all sufficiently large n. We now describe the test statistic for each detector D_n.

Equation (3.4.6) in Corollary (3.2) guarantees the existence of an index pair (i,j) such that $P(\mathbf{X}_1^2=(i,j))\neq Q_\beta(\mathbf{Y}_1^2=(i,j))$ for all $\beta>0$. Thus, we define the test statistic $\nu_{\beta,n}$ for detector D_n as

$$\nu_{\beta,n}=\sqrt{n}\left|\frac{1}{n-1}h_\beta(i,j)-P\big(\mathbf{X}_1^2=(i,j)\big)\right|, \tag{3.5.1}$$

where $\frac{1}{n-1}h_\beta(i,j)$ is the relative count of the number of consecutive pixel pairs (i,j) in an n-element stego image embedded using parameter β (In terms of Iverson bracket, $\frac{1}{n-1}h_\beta(i,j)=\frac{1}{n-1}\sum_{k=1}^{n-1}[Y_k=i][Y_{k+1}=j]$). Note that due to stationarity of the cover source, $E[h_\beta(i,j)]=(n-1)Q_\beta(\mathbf{Y}_1^2=(i,j))$ for all β.

We prove the following for the difference between the means of $\nu_{\beta_n,n}$ under both hypotheses

$$\lim_{n\to\infty}E[\nu_{\beta_n,n}]-E[\nu_{0,n}]=\infty\text{ when }\sqrt{n}\beta_n\to\infty. \tag{3.5.2}$$

By contradiction, there exists $K>0$, such that for all n_0 there exists $n\geq n_0$ such that $|E[\nu_{\beta_n,n}]-E[\nu_{0,n}]|<K$. We can thus obtain a strictly increasing sequence of integers $(n_m)_{m=1}^\infty$ for which

$$|E[\nu_{\beta_{n_m},n_m}]-E[\nu_{0,n_m}]|<K\text{ for all }m. \tag{3.5.3}$$

If $\limsup_{m\to\infty}\beta_{n_m}=\beta_0>0$, then there exists a subsequence of $(n_m)_{m=1}^\infty$, which we denote the same to keep the notation simple, such that $\lim_{m\to\infty}\beta_{n_m}=\beta_0$. For this subsequence, however, the difference

$$E[\nu_{\beta_{n_m},n_m}]-E[\nu_{0,n_m}]\quad=\quad\sqrt{n_m}\Big|Q_{\beta_{n_m}}\big(\mathbf{Y}_1^2=(i,j)\big)-P\big(\mathbf{X}_1^2=(i,j)\big)\Big|$$

tends to ∞ with $m\to\infty$ because by (3.4.6) the absolute value converges to a positive value independent of m. This is, however, a contradiction with (3.5.3).

If $\lim_{m\to\infty}\beta_{n_m}=0$, we find the contradiction in a different manner. By the FI condition from Corollary (3.2), there must exist indices (i,j) such that $\frac{d}{d\beta}Q_{\beta=0}(\mathbf{Y}_1^2=(i,j))\neq 0$. From Taylor expansion[1] of $Q_\beta(\mathbf{Y}_1^2=(i,j))$ at $\beta=0$ with Lagrange remainder and $0<\xi<1$

$$E[\nu_{\beta_{n_m},n_m}]-E[\nu_{0,n_m}]=\sqrt{n_m}\beta_{n_m}\left|\frac{d}{d\beta}Q_{\beta=0}\big(\mathbf{Y}_1^2=(i,j)\big)+\frac{1}{2}\beta_{n_m}\frac{d^2}{d\beta^2}Q_{\xi\beta_{n_m}}\big(\mathbf{Y}_1^2=(i,j)\big)\right|, \tag{3.5.4}$$

[1] The Taylor expansion is valid since by its form the function $Q_\beta(\mathbf{Y}_k^{k+1}=(i,j))$ is analytic.

which tends to ∞ as $m \to \infty$ when $\sqrt{n_m}\beta_{n_m} \to \infty$, which is again a contradiction with (3.5.3). We summarize that $E[\nu_{\beta_n,n}] - E[\nu_{0,n}] \to \infty$ holds for any sequence β_n for which $\sqrt{n}\beta_n \to \infty$.

Lemma A.1 in Appendix A shows that exponential forgetting of Markov chains guarantees that

$$Var[\nu_{\beta,n}] < C \tag{3.5.5}$$

for some constant C independent of n and β. Equations (3.5.2) and (3.5.5) are all we need to construct detectors D_n that will put the Steganographer at risk for all sufficiently large n. The detector D_n has the following form

$$\begin{aligned} \nu_{\beta,n} > T \quad &\text{decide stego } (\beta > 0) \\ \nu_{\beta,n} \le T \quad &\text{decide cover } (\beta = 0), \end{aligned}$$

where T is a fixed threshold. We now show that T can be chosen to make the detector probability of false alarms and missed detections satisfy

$$\begin{aligned} P_{\text{FA}} &< P^*_{\text{FA}} \\ P_{\text{MD}} &< P^*_{\text{MD}} \end{aligned}$$

for sufficiently large n. The threshold $T(P^*_{\text{FA}})$ will be determined from the requirement that the probability of the right tail, $x \ge T(P^*_{\text{FA}})$, under H_0 is at most P^*_{FA}. Using Chebyshev's inequality,

$$P_{\text{FA}} = Pr(\nu_{0,n} \ge T) \le Pr(|\nu_{0,n}| \ge T) \le \frac{Var[\nu_{0,n}]}{T^2} < \frac{C}{T^2}.$$

Setting $T = \sqrt{C/P^*_{\text{FA}}}$ gives us $P_{\text{FA}} < P^*_{\text{FA}}$.

Because of the growing difference between the means (3.5.2), we can find n large enough so that the probability of the left tail, $x \le T(P^*_{\text{FA}})$, under H_1 is less than or equal to P^*_{MD}. Again, we use the Chebyshev's inequality with the bound on the variance of $\nu_{\beta,n}$ to prove this

$$\begin{aligned} P_{\text{MD}} &= Pr\big(\nu_{\beta,n} < T(P^*_{\text{FA}})\big) = Pr\big(\nu_{\beta,n} - E[\nu_{\beta,n} - \nu_{0,n}] < T(P^*_{\text{FA}}) - E[\nu_{\beta,n} - \nu_{0,n}]\big) \\ &\le Pr\big(|\nu_{\beta,n} - E[\nu_{\beta,n} - \nu_{0,n}]| > E[\nu_{\beta,n} - \nu_{0,n}] - T(P^*_{\text{FA}})\big) < \frac{C}{\left(E[\nu_{\beta,n} - \nu_{0,n}] - T(P^*_{\text{FA}})\right)^2}, \end{aligned}$$

which can be made arbitrarily small for sufficiently large n because $E[\nu_{\beta_n,n}] - E[\nu_{0,n}] \to \infty$. This establishes the first part of the Square-root law.

Part 2 [Asymptotic undetectability] Now we prove that when $\sqrt{n}\beta_n \to 0$, then the KL divergence between the distributions of cover and stego objects

$$d_n(\beta_n) = D_{\text{KL}}\left(P^{(n)} || Q^{(n)}_{\beta_n}\right) = \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P^{(n)}(\mathbf{X}_1^n = \mathbf{y}_1^n) \ln \frac{P^{(n)}(\mathbf{X}_1^n = \mathbf{y}_1^n)}{Q^{(n)}_{\beta_n}(\mathbf{Y}_1^n = \mathbf{y}_1^n)} \to 0, \tag{3.5.6}$$

which will establish that the steganography is ε-secure for any $\varepsilon > 0$ for sufficiently large n.

Using Taylor expansion of $d_n(\beta)$ with Lagrange remainder at $\beta = 0$ we have $d_n(\beta) = d_n(0) + d'_n(0)\beta + \frac{d''_n(\upsilon\beta)}{2!}\beta^2$, where $0 < \upsilon < 1$. This step is valid since, by Lemma A.2 from Appendix A, all derivatives of (normalized) KL divergence are continuous w.r.t. β. The term $d_n(0)$ is zero because both distributions are the same when $\beta = 0$. The term $d'_n(0)$ is also zero because

$$\begin{aligned} d'_n(0) = \lim_{\beta \to 0} d'_n(\beta) &= \lim_{\beta \to 0} - \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P^{(n)}(\mathbf{X}_1^n = \mathbf{y}_1^n) \frac{\frac{d}{d\beta} Q^{(n)}_\beta(\mathbf{Y}_1^n = \mathbf{y}_1^n)}{Q^{(n)}_\beta(\mathbf{Y}_1^n = \mathbf{y}_1^n)} \\ &= - \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} \frac{d}{d\beta} Q^{(n)}_{\beta=0}(\mathbf{Y}_1^n = \mathbf{y}_1^n) \\ &= \lim_{\beta \to 0} -\frac{d}{d\beta} \underbrace{\Big(\sum_{\mathbf{y}_1^n \in \mathcal{I}^n} Q^{(n)}_\beta(\mathbf{Y}_1^n = \mathbf{y}_1^n) \Big)}_{=1} = 0. \end{aligned}$$

Finally, by Lemma A.2 from Appendix A there exists a constant $\tilde{C} < \infty$, such that $\frac{1}{n}d''_n(\beta) < \tilde{C}$ for $\beta \in [0, \beta_0]$ and all n. Thus, $d_n(\beta_n) \leq \frac{1}{2}\tilde{C}n\beta_n^2 \to 0$ when $\sqrt{n}\beta_n \to 0$.

Part 3 [Asymptotic ε-security] To prove the third part of the Square-root law, we again expand the KL divergence $d_n(\beta)$ at $\beta = 0$ up to the third order with the Lagrange form of the remainder

$$d_n(\beta) = \frac{1}{2!}\Big(\frac{d''_n(0)}{n}\Big)n\beta^2 + \frac{1}{3!}\Big(\frac{d'''_n(\upsilon\beta)}{n}\Big)n\beta^3 \tag{3.5.7}$$

for some $0 < \upsilon < 1$. According Lemma A.2 from Appendix A, both normalized derivatives of the KL divergence, $\frac{1}{n}d''_n(0)$ and $\frac{1}{n}d'''_n(\upsilon\beta)$, are upper bounded by the same finite constant $\tilde{C}$ for all $\beta \in [0, \beta_0]$. Since $\beta_n\sqrt{n} \to \epsilon$ with $n \to \infty$, $\beta_n \to 0$ and thus the expansion is valid. By the same reason, the second term in (3.5.7) converges to zero as $n \to \infty$. From this result, we obtain the asymptotic bound on KL divergence in the form $d_n\beta_n \leq \frac{1}{2}\tilde{C}\epsilon^2$ as was to be shown. □

3.6 Summary and Discussion

It is now clear that the theory of *hidden* information is quite unlike the traditional theory of information. Whether steganography is performed in a large batch of cover objects or a single large object, there is a wide range of situations in which *secure payload grows according to the square root of the cover size.* Such results will likely hold for all stegosystems that are not perfectly secure in the sense of Cachin for which the warden is able to obtain a detector — warden is not forced to know the cover source exactly.

The results presented in this chapter proved the so-called Square-root law of imperfect stegosystems under two different assumptions on the cover source. In the more realistic one, we have assumed imperfect stegosystems with cover source represented in the form of a stationary Markov chain with the embedding algorithm performing mutually-independent substitutions of individual cover elements according to a given embedding operation. We argue that this applies to a very wide range of popular steganographic algorithms, in spatial and transform domains. The fact that we constraint on imperfect stegosystems is important because it is known that the secure payload scales linearly w.r.t. the cover size for perfectly secure stegosystems. Such systems can always be constructed if the cover source is perfectly understood [16, 124].

The Square-root law has some important implications in steganography and steganalysis. For example, it explains why the same relative payload can be detected more accurately in large images. Thus, when benchmarking steganography, the distribution of image sizes in the database influences the reliability of steganalysis and makes it more difficult to compare the results on two different databases. To resolve this issue, one might switch to measuring the payload in bits per square root of pixel which is the topic of the next chapter.

Finally, we emphasize that the Square-root law relates to the number of changes caused by the embedding process, and not to the size of the information transmitted. The latter can gain an additional logarithmic factor, if adaptive source coding is used, but information capacity remains sublinear in the absence of perfect steganography.

Chapter 4

Fisher Information

The key concept in essentially all communication systems is the amount of information one can send through them. In many channel-coding problems, the amount of such information scales linearly with the number of samples one can use for transmission. This is paralleled by the secure payload in perfectly secure stegosystems. In all these cases, maximal achievable communication rate is positive and often termed as *the capacity.*

For imperfect stegosystems, the communication rate is not a good descriptor of the channel because it approaches zero with increasing n. Alice, however, still needs to know what level of risk she is exposing herself to when sending a message to Bob. It is critical for her to know how much information she can send using her stegosystem in an n-element cover, while keeping the KL divergence between cover and stego objects below some chosen ε. As shown in Chapter 3, under fairly general assumptions, the amount of information that she can hide scales as $r\sqrt{n}$, with r constant.

In this chapter, we propose to use the proportionality constant r from the SRL as a more refined measure of steganographic capacity of imperfect stegosystems. By the form of the law, the constant r, for which we coin the term *the root rate,* essentially expresses the capacity per square root of cover size. We derive a closed form expression for the root rate under the assumption that covers form a Markov chain and embedding is realized by applying a sequence of independent embedding operations to individual cover elements. The root rate depends on the Fisher information rate w.r.t. the the change rate, which was shown to be a perfect security descriptor equivalent to the KL divergence between distributions of cover and stego objects (see Section 3.4). Expressing the Fisher information rate analytically as a quadratic form allows us to evaluate, compare, and optimize security of stegosystems. To this end, we derive an analytic cover model from a large database of natural images represented in the spatial domain and show that the ±1 embedding operation is asymptotically optimal among all mutually independent embedding operations that modify cover elements by at most 1. Finally, using the Fisher information rate, we compare security of several practical stegosystems, including LSB embedding and ±1 embedding. Our findings appear to be consistent with results previously obtained experimentally using steganalyzers.

In [71], Ker used the same asymptotic behavior of the secure payload and coined the term *Steganographic Fisher Information (SFI)* for the quadratic term in Taylor expansion of the Kullback-Leibler divergence. Instead of measuring this term analytically as done in this chapter, he measured the SFI from a very large corpus of images when represented as small pixel groups (pairs, triples, ...). Althought both approaches are based on different assumptions about the cover model, they provide comparable results for LSB and ±1 embedding. On the other hand, the results differ on image sets with dependencies not covered by our analytical model, such as decompressed JPEG images [31].

This chapter is structured as follows. In the next section, we introduce the concept of the root rate as a measure of steganographic capacity of imperfect stegosystems. At the same time, we derive a closed-form expression for the Fisher information rate on which the root rate depends. Section 4.2 contains the theoretical foundation for comparing stegosystems and for maximizing the root rate with respect to the embedding operation for a fixed cover source. In Section 4.3, we present comparison of several known embedding operations for three spatial domain analytic cover models derived from databases of raw, JPEG, and scanned images. Also, we prove that ternary ±1 embedding has the highest root rate among all stegosystems that modify cover elements by at most 1. The chapter is concluded in Section 4.4.

4.1 Capacity of Imperfect Stegosystems

In this section, we introduce the concept of root rate as a measure of capacity of imperfect stegosystems. As in Chapter 3, we assume Assumptions 1–3 from Section 3.3 to hold. Under these assumptions we will work with the Fisher information defined for n-element covers w.r.t. the parameter β

$$I_n(0) = E_P\left[\left(\frac{d}{d\beta}\ln Q_\beta^{(n)}(\mathbf{Y}_1^n)\Big|_{\beta=0}\right)^2\right]. \tag{4.1.1}$$

4.1.1 Root Rate

As discussed in Section 2.3, the problem of steganalysis can be formulated as the following hypothesis testing problem

$$\begin{aligned} \mathrm{H}_0 &: \beta = 0 \\ \mathrm{H}_1 &: \beta = \beta_0 > 0. \end{aligned} \tag{4.1.2}$$

We show that for small (and known) β_0 and large n, the likelihood ratio test with test statistic

$$\frac{1}{\sqrt{n}}T_{\beta_0}^{(n)}(\mathbf{X}_1^n) = \frac{1}{\sqrt{n}}\ln\left(\frac{Q_{\beta_0}^{(n)}(\mathbf{X}_1^n)}{P^{(n)}(\mathbf{X}_1^n)}\right), \tag{4.1.3}$$

is a mean-shifted Gauss-Gauss problem.[1] This property, usually called the Local Asymptotic Normality (LAN) of the detector, allows us to quantify and correctly compare security of embedding algorithms operating on the same MC cover model for small values of β.

In this case, the detector performance can be completely described by the deflection coefficient d^2, which parametrizes the ROC curve as it binds the probability of detection, $P_\mathrm{D} \triangleq 1 - P_\mathrm{MD}$, as a function of the false alarm probability, P_FA,

$$P_\mathrm{D} = \mathcal{Q}\big(\mathcal{Q}^{-1}(P_\mathrm{FA}) - \sqrt{d^2}\big).$$

Here, $\mathcal{Q}(x) = 1 - \Phi(x)$ and $\Phi(x)$ is the cdf of a standard normal variable $N(0,1)$. Large value of the deflection coefficient implies better detection or weaker steganography.

First, we state the LAN property for the HMC model w.r.t. the embedding parameter β and then extend this result with respect to the relative payload α.

Theorem 4.1 (LAN of the LLRT)**.** *Under Assumptions 1–3 from Section 3.3, the likelihood ratio (4.1.3) satisfies the local asymptotic normality (LAN), i.e., under both hypotheses and for values of β up to order β^2*

$$\sqrt{n}\left(\frac{T_\beta^{(n)}}{n} + \frac{\beta^2 I}{2}\right) \xrightarrow{d} N(0, \beta^2 I) \text{ under } \mathrm{H}_0 \tag{4.1.4}$$

$$\sqrt{n}\left(\frac{T_\beta^{(n)}}{n} - \frac{\beta^2 I}{2}\right) \xrightarrow{d} N(0, \beta^2 I) \text{ under } \mathrm{H}_1, \tag{4.1.5}$$

where I is the Fisher information rate, $I = \lim_{n\to\infty}\frac{1}{n}I_n(0)$, and $\xrightarrow{d}$ is the convergence in distribution. The detection performance is thus completely described by the deflection coefficient

$$d^2 = \frac{(\sqrt{n}\beta^2 I/2 + \sqrt{n}\beta^2 I/2)^2}{\beta^2 I} = n\beta^2 I.$$

[1] In hypothesis testing, the problem of testing $N(\mu_0, \sigma^2)$ vs. $N(\mu_1, \sigma^2)$ is called the mean-shifted Gauss-Gauss problem and its detection performance is completely described by the deflection coefficient $d^2 = (\mu_0 - \mu_1)^2/\sigma^2$ [61, Chapter 3].

Proof. By a simple algebra, the leading term in the Taylor expansion of the mean and variance of the likelihood ratio (4.1.3) w.r.t. β is quadratic and consists of the Fisher information rate. This is valid under both H_0 and H_1.

The Gaussianity of the leading terms of the test statistic follows from a variant of the Central Limit Theorem (CLT), which is discussed in the rest of the proof. The standard proof of the CLT uses a moment generating function and shows that it can be factorized and thus converges to the moment generating function of a Gaussian random variable for large n. Finally, by using Lévy's continuity theorem, we obtain the convergence in distribution. In our case, the assumption of independence is missing and is replaced by so called "exponential forgetting," which can be used to prove a similar result. This approach was used to prove the CLT for functions of Markov chains [20], because samples far enough can be seen as "almost" independent, which allows us to use the approach from the i.i.d. case (see [20, §V, Theorem 7.5 on page 228] for an application of this idea.).

In our case, we use the prediction filter (see the discussion before Lemma A.7 in Appendix A) to write the statistic as a sum of terms that satisfy exponential forgetting. This type of description is classical in the theory of hidden Markov chains [22, p. 1538]. The exponential forgetting of the prediction filter and its derivatives, which are key to our approach, were proved in Lemma A.8 in Appendix A. □

We now reformulate the conclusion of the theorem in terms of the payload rather than the parameter β. Matrix embedding (syndrome coding) employed by the stegosystem may introduce a non-linear relationship $\beta = f(\alpha)$ between both quantities. In general, the payload embedded at each cover element may depend on its state (or color) $i \in \mathcal{I}$ (e.g., see the last two matrices in Figure 3.3.1). Thus, the expected value of the relative payload that can be embedded in each cover is $\alpha(\beta) = \sum_{i\in\mathcal{I}} \pi_i \alpha_i(\beta)$, where $\alpha_i(\beta)$ stands for the number of bits that can be embedded into state $i \in \mathcal{I}$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_N)$ is the stationary distribution of the MC. The value of β for which α is maximal will be denoted as β_{MAX}

$$\beta_{MAX} = \arg\max_{\beta} \alpha(\beta).$$

For example, for ternary ±1 embedding $\beta_{MAX} = 2/3$ and $\alpha_i(\beta_{MAX}) = \log_2 3$, while for binary ±1 embedding $\beta_{MAX} = 1/2$ and $\alpha_i(\beta_{MAX}) = 1$ (see Figure 3.3.1 for the corresponding matrices). Notice that the matrix $\mathbb{C}$ is the same for both embedding methods. The only formal difference is the range of the parameter β. We also remark that unless all α_i are the same, the maximal payload will depend on the distribution of individual states π_i.

To simplify our arguments, we assume a linear relationship between β and α (e.g., we do not consider in this chapter the effects of matrix embedding). Therefore, we can write

$$\beta = f(\alpha) = \frac{\beta_{MAX}}{\alpha_{MAX}}\alpha, \tag{4.1.6}$$

where $\alpha \in [0, \alpha_{MAX}]$ and $\alpha_{MAX} = \alpha(\beta_{MAX})$ denotes the average number of bits that can be embedded into cover element while embedding with $\beta = \beta_{MAX}$ (maximum change rate).

From (4.1.6), the deflection coefficient can be expressed in terms of the relative payload α by substituting $\beta = f(\alpha)$ from (4.1.6) into the stego distribution Q_β

$$d^2 = n\alpha^2 \left(\frac{\beta_{MAX}}{\alpha_{MAX}}\right)^2 I. \tag{4.1.7}$$

In practice, Alice can control statistical detectability by bounding $d^2 < \varepsilon$ for some fixed ε, obtaining thus an upper bound on the total number of bits (payload) αn that can be safely embedded (this requires rearranging the terms in (4.1.7))

$$\alpha n \leq \frac{\alpha_{MAX}}{\beta_{MAX}} \sqrt{\frac{\varepsilon}{I} n}. \tag{4.1.8}$$

In analogy to the communication rate, it is natural to define *the root rate*

$$r \triangleq \frac{\alpha_{MAX}}{\sqrt{I}\beta_{MAX}} \tag{4.1.9}$$

as the quantity that measures steganographic security of imperfect stegosystems in bits per square root of cover size per square root of KL divergence. We use the root rate for comparing stegosystems with a MC cover model.

In the next theorem, we establish the existence of the main component of the root rate, the Fisher information rate I, and express it in a closed form.

Theorem 4.2 (Fisher information rate)**.** *As in Section 3.3, let $\mathbb{A} = (a_{i,j})$ define the MC cover model with stationary distribution $\boldsymbol{\pi}$ and $\mathbb{B}$, defined by matrix $\mathbb{C} = (c_{i,j})$, capture the embedding algorithm. Then, the normalized Fisher information $I_n(0)/n$ approaches a finite limit I as $n \to \infty$. This limit can be written as $I = \mathbf{c}^T\mathbb{F}\mathbf{c}$, where $\mathbf{c}$ is obtained by arranging $\mathbb{C}$ into a column vector of size N^2 with elements $c_{i,j}$.*[2] *The matrix $\mathbb{F}$ of size $N^2 \times N^2$ is defined only in terms of matrix $\mathbb{A}$ and does not depend on the embedding algorithm. The elements of matrix $\mathbb{F}$ are*

$$f_{(i,j),(k,l)} = [j = l]V(i,j,k) - U(i,j,k,l), \tag{4.1.10}$$

where by the Iverson notation $[j = l]$ is one if $j = l$ and zero otherwise and

$$V(i,j,k) = \left(\sum_{z\in\mathcal{I}} \pi_z a_{z,i}\frac{a_{z,k}}{a_{z,j}}\right)\left(\sum_{z\in\mathcal{I}} a_{i,z}\frac{a_{k,z}}{a_{j,z}}\right) \tag{4.1.11}$$

$$U(i,j,k,l) = \pi_i\left(a_{i,k} - a_{i,l}\frac{a_{j,k}}{a_{j,l}}\right) + \pi_k\left(a_{k,i} - a_{k,j}\frac{a_{l,i}}{a_{l,j}}\right). \tag{4.1.12}$$

Moreover, $|I_n(0)/n - I| \le C/n$ for some constant C. This constant depends only on the elements of matrix $\mathbb{A}$ and not on the embedding algorithm. The quadratic form $I(\mathbf{c}) = \mathbf{c}^T\mathbb{F}\mathbf{c}$ is semidefinite, in general.

Proof. Here, we only present the main idea of the proof, leaving all technical details to Appendix B. The decomposition of the sequence $I_n(0)/n$ to a quadratic form and its properties can be obtained directly from the definition of Fisher information

$$\begin{aligned}\frac{1}{n}I_n(0) &= \frac{1}{n}\frac{\partial^2}{\partial\beta^2}d_n(\beta)\Big|_{\beta=0}\\ &= -\sum_{(i,j)}\sum_{(k,l)}\frac{1}{n}E_P\Bigg[\underbrace{\left(\frac{\partial^2}{\partial b_{i,j}b_{k,l}}\ln Q_\beta(\mathbf{Y}_1^n)\Big|_{\mathbb{B}=\mathbb{I}}\right)}_{\triangleq g(\mathbf{Y}_1^n,i,j,k,l)}\Bigg]\underbrace{\left(\frac{\partial b_{i,j}}{\partial\beta}\Big|_{\beta=0}\right)}_{=c_{i,j}}\underbrace{\left(\frac{\partial b_{k,l}}{\partial\beta}\Big|_{\beta=0}\right)}_{=c_{k,l}}.\end{aligned}$$

The derivatives of the log-likelihood are evaluated at $\mathbb{B} = \mathbb{I}$ because $\mathbb{B}_\beta = \mathbb{I}+\beta\mathbb{C}$ and $\beta = 0$. By using $Q_\beta(\mathbf{y}_1^n) = \sum_{\mathbf{x}_1^n\in\mathcal{I}^n} P(\mathbf{x}_1^n)Q_\beta(\mathbf{y}_1^n|\mathbf{x}_1^n)$, the random variable $g(\mathbf{Y}_1^n,i,j,k,l)$ does not depend on the embedding method. This is because the derivatives are evaluated at $\mathbb{B} = \mathbb{I}$ and thus only contain the elements of the cover source transition matrix $\mathbb{A}$. The proof of the convergence of $-\frac{1}{n}E_P[g(\mathbf{Y}_1^n,i,j,k,l)]$ to $f_{(i,j),(k,l)}$ and its closed form is more involved and is presented in Lemma B.2 and Lemma B.3 in Appendix B. The semidefinitness of the quadratic form follows from semidefiniteness of the Fisher information matrix $\mathbb{F}$. It is not positively definite because for an i.i.d. cover source all rows of matrix $\mathbb{F}$ coincide and are thus linearly dependent. □

By inspecting the proof of the theorem, the matrix $\mathbb{F}$ can be seen as the Fisher information rate matrix w.r.t. the parameters $\{b_{i,j}|1 \le i,j \le N\}$. It describes the natural sensitivity of the cover source to MI embedding. The quadratic form then combines these sensitivities with coefficients given by the specific embedding method and allows us to decompose the intrinsic detectability caused by the cover source from the detectability caused by the embedding algorithm.

[2] The order of elements in $\mathbb{C}$ is immaterial as far as the same ordering is used for pairs (i,j) and (k,l) in matrix $\mathbb{F}$.

Corollary 4.1. *For the special case when the MC degenerates to an i.i.d. cover source with distribution $P = \boldsymbol{\pi}$, the Fisher information rate simplifies to*

$$I = \sum_{i,j,k \in \mathcal{I}} c_{i,j} \frac{\pi_i \pi_k}{\pi_j} c_{k,j}.$$

4.2 Maximizing the Root Rate

In the previous section, we established that the steganographic capacity of imperfect stegosystems should be measured as the root rate (4.1.9) defined as the payload per square root of the cover size and per square root of KL divergence. The most important component of the root rate is the stegosystem's Fisher information rate, for which an analytic form was derived in Theorem 4.2. The steganographer is interested in designing stegosystems (finding $\mathbb{C}$) with the highest possible root rate. This can be achieved by minimizing the Fisher information rate or by embedding symbols from a larger alphabet, i.e., increasing the ratio α_{MAX}/β_{MAX}. In this section, we describe two general strategies for maximizing the root rate that are applicable to practical stegosystems. In Section 4.3, we draw conclusions from experiments when these strategies are applied to real cover sources formed by digital images.

Before proceeding with further arguments, we point out that the highest root rate is obviously obtained when the Fisher information rate is zero, $I = 0$. This can happen for non-trivial embedding ($\mathbb{C} \neq 0$) in certain sources because the Fisher information rate is a semidefinite quadratic form. Such stegosystems, however, would be perfectly secure and thus by Assumption 3 from Section 3.3 are excluded from our consideration.[3]

The number of bits, α_i, that can be embedded at each state $i \in \mathcal{I}$ is bounded by the entropy of the ith row of $\mathbb{B} = \mathbb{I} + \beta\mathbb{C}$, $H(\mathbb{B}_{i,\bullet})$. Thus, in the most general setting, we wish to maximize the root rate

$$\frac{\sum \pi_i H\left(\mathbb{B}_{i\bullet}(\beta_{MAX})\right)}{\beta_{MAX}} \frac{1}{\sqrt{I}}$$

w.r.t. matrix $\mathbb{C}$. The nonlinear objective function makes the analysis rather complicated and the result may depend on the distribution of individual states $\boldsymbol{\pi}$.

In the rest of this section, we present two different approaches how to optimize the embedding algorithm under different conditions.

4.2.1 Optimization by Convex Combination of Known Methods

One simple and practical approach to optimize the embedding method is obtained by combining existing stegosystems $S^{(1)}$ and $S^{(2)}$. Suppose Alice and Bob embed a portion of the message into λn elements, $0 < \lambda < 1$, using $S^{(1)}$ and use the remaining $(1-\lambda)n$ elements to embed the rest of the message using $S^{(2)}$. If both parties select the elements pseudo-randomly based on a stego key, the impact on a single cover element follows a distribution obtained as a convex combination of the noise pmfs of both methods. Note that the methods are allowed to embed a different number of bits per cover element since Bob knows which symbol to extract from each part of the stego object. Let $S^{(i)}$ represent the ith embedding method with matrix $\mathbb{C}^{(i)}$, or its vector representation $\mathbf{c}^{(i)}$, with ratio $\rho^{(i)} = \alpha^{(i)}_{MAX}/\beta^{(i)}_{MAX}$ for $i \in \{1,2\}$. The root rate $r(\lambda)$ of the method obtained by the above approach (convex embedding) with parameter λ can be written as

$$\begin{aligned} r(\lambda) &= \frac{\lambda\rho^{(1)} + (1-\lambda)\rho^{(2)}}{\sqrt{(\lambda\mathbf{c}^{(1)} + (1-\lambda)\mathbf{c}^{(2)})^T \mathbb{F} (\lambda\mathbf{c}^{(1)} + (1-\lambda)\mathbf{c}^{(2)})}} \\ &= \frac{\lambda\rho^{(1)} + (1-\lambda)\rho^{(2)}}{\sqrt{\lambda^2 I^{(1)} + (1-\lambda)^2 I^{(2)} + 2\lambda(1-\lambda) I^{(1,2)}}}, \end{aligned} \tag{4.2.1}$$

where $I^{(i)}$ is the Fisher information rate of $S^{(i)}$ and $I^{(1,2)} = \left(\mathbf{c}^{(1)}\right)^T \mathbb{F}\mathbf{c}^{(2)}$. Here, we used the symmetry of $\mathbb{F}$ to write $I^{(1,2)} = I^{(2,1)}$.

[3] An example of such a stegosystem is LSB embedding in i.i.d. covers with $\pi_{2i} = \pi_{2i+1}$ for all i.

4.2.2 Minimizing the Fisher Information Rate

In an alternative setup, we deal with the problem of optimizing the shape of the additive noise pmf under the assumption that the number of bits, α_i, embedded at each state $i \in \mathcal{I}$ is constant. For example, we may wish to determine the optimal pmf that would allow us to communicate 1 bit per element ($\alpha_i = 1$, $\forall i \in \mathcal{I}$) by changing each cover element by at most 1. In this problem, the ratio α_{MAX}/β_{MAX}, as well as the cover model (matrix $\mathbb{A}$), are fixed and known. The task is to minimize the Fisher information rate I.

We formulate our optimization problem by restricting the form of the matrix $\mathbb{C} = (c_{i,j})$, or its vector representation $\mathbf{c} = (c_{i,j}) \in \mathbb{R}^{N^2 \times 1}$, to the following linear parametric form

$$\mathbf{c} = \mathbb{D}\mathbf{v} + \mathbf{e}, \tag{4.2.2}$$

where $\mathbb{D} = (d_{i,j})$ is a full-rank real matrix of size $N^2 \times k$, $\mathbf{e}$ is a real column vector of size N^2, and $\mathbf{v} = (v_1, \ldots, v_k)^T$ is a k-dimensional column vector. We assume $\mathbf{v} \in \mathcal{V} \subset \mathbb{R}^{k \times 1}$, where $\mathcal{V}$ is bounded by a set of linear inequalities[4] and the constraint $\sum_j c_{i,j} = 0$ for all $i \in \{1, \ldots, N\}$. In other words, we decompose the matrix $\mathbb{C}$ into k real parameters v_i, $i \in \{1, \ldots, k\}$. The following example shows one such representation for a stegosystem whose embedding changes are at most 1.

Example 4.1 (Tridiagonal embedding)**.** We set $c_{i,i} = -1$, $c_{i,i-1} = v_{i-1}$, and $c_{i,i+1} = 1 - v_{i-1}$ for $i \in \{2, \ldots, N-1\}$ (and suitably defined at the boundaries). This allows us to model ±1 embedding, LSB embedding, and all possible MI embedding methods that modify every element by at most 1. By setting $c_{i,i} = -1$ for all i, we constrain ourselves to stegosystems that embed the same payload into every state $i \in \mathcal{I}$ for all $\beta \geq 0$. This model has $k = N - 2$ parameters and the set $\mathcal{V}$ is formed by $v_j \in [0, 1]$, $j \in \{1, \ldots, k\}$.

Our task is to minimize the Fisher information rate for embedding methods given by (4.2.2). The function $I(\mathbf{v}) = (\mathbb{D}\mathbf{v} + \mathbf{e})^T \mathbb{F} (\mathbb{D}\mathbf{v} + \mathbf{e})$ can attain its minimum either at a point with a zero gradient[5] (a critical point) or on the boundary of $\mathcal{V}$. We now derive a set of linear equations for the set of all possible critical points. This approach will be used in Section 4.3 to prove that ternary ±1 embedding is asymptotically optimal within the class of tridiagonal embedding in spatial domain.

For our parametrization, the gradient w.r.t. every parameter v_j can be expressed as

$$\frac{\partial}{\partial v_j} I(\mathbf{v}) = \frac{\partial}{\partial v_j} (\mathbb{D}\mathbf{v} + \mathbf{e})^T \mathbb{F} (\mathbb{D}\mathbf{v} + \mathbf{e}) = 2(\mathbb{D}_{\bullet j})^T \mathbb{F} (\mathbb{D}\mathbf{v} + \mathbf{e}),$$

where $\mathbb{D}_{\bullet j}$ is the jth column of matrix $\mathbb{D}$. Because every possible candidate $\mathbf{v}_0$ for the optimal parameters must satisfy $(\partial/\partial v_j) I(\mathbf{v})|_{\mathbf{v}=\mathbf{v}_0} = 0$ for every $j \in \{1, \ldots, k\}$, all critical points are solutions of the following linear system

$$\mathbb{D}^T \mathbb{F} \mathbb{D} \mathbf{v} = -\mathbb{D}^T \mathbb{F} \mathbf{e}. \tag{4.2.3}$$

If this system has a unique solution $\mathbf{v}_0 \in \mathcal{V}$, then $\mathbf{v}_0$ corresponds to matrix $\mathbb{C}$ achieving the global minimum of the Fisher information rate, which corresponds to the best MI embedding method w.r.t. $\mathcal{V}$ and a given MC cover source.

4.3 Experiments

In the previous section, we outlined two strategies for maximizing the root rate for practical stegosystems. This section presents specific results when these strategies are applied to stegosystems operating on 8-bit gray-scale images represented in the spatial domain. Although images are two dimensional objects with spatial dependencies in both directions, we represent them in a row-wise fashion as a first-order Markov Chain over $\mathcal{I} = \{0, \ldots, 255\}$. The MC model represents the first and simplest step of capturing pixel dependencies while still retaining the important advantage of being analytically tractable. Then, we adopt a parametric model for the transition probability matrix

[4] E.g., we must have $\mathbb{B} \geq 0$.

[5] Note that the semidefiniteness of $\mathbb{F}$ guarantees that the extremum must be a minimum.

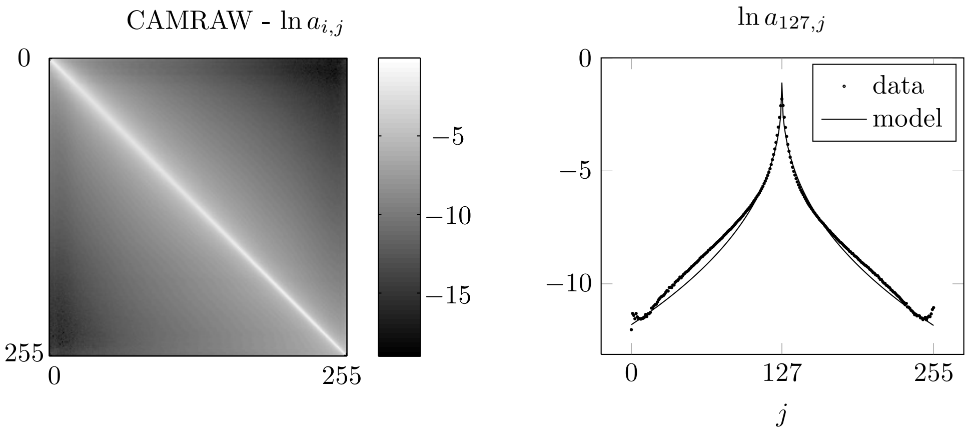

Figure 4.3.1: Left: plot of the empirical matrix $\mathbb{A}$ estimated from CAMRAW database in log domain. Right: comparison of the 128th row of matrix $\mathbb{A}$ estimated from the same database with the analytic model (4.3.1).

of this Markov cover source and show that it is a good fit for the empirical transition probability matrix $\mathbb{A}$ estimated from a large number of natural images. We use the analytic model to evaluate the root rate (4.1.9) of several stegosystems obtained by a convex combinations of known methods. Finally, we show that the optimal embedding algorithm that modifies cover elements by at most 1 is very close to ±1 embedding.

In principle, in practice we could calculate the Fisher information rate using Equation (4.1.10) with an empirical matrix $\mathbb{A}$ estimated from a large number of images. However, this approach may give misleading results because (4.1.10) is quite sensitive to small perturbations of $a_{i,j}$ with a small value (observe that $I = +\infty$ if $a_{i,j} = 0$). We do not expect this to be an issue in practice since rare transitions between distant states are probable but content dependent, which makes them difficult to be utilized for steganalysis. Because small values of $a_{i,j}$ can not be accurately estimated in practice, we represent the matrix $\mathbb{A}$ with the following parametric model

$$a_{i,j} = \frac{1}{z_i} e^{-(|i-j|/\tau)^\gamma}, \tag{4.3.1}$$

where $z_i = \sum_{j=0}^{255} e^{-(|i-j|/\tau)^\gamma}$ is the normalization constant. The parameter γ controls the shape of the distribution, whereas τ controls its "width." The model parameters were found in the logarithmic domain using the least square fit between (4.3.1) and its empirical estimate. To validate this model, we carried out the least square fit separately for three image databases: never compressed images taken by several digital cameras[6] (CAMRAW), digital scans[7] (NRCS), and decompressed JPEG images[8] (NRCS-JPEG). Figure 4.3.1 shows the comparison between the empirical matrix $\mathbb{A}$ estimated from the CAMRAW database by scanning each image in a row-wise fashion and the corresponding fit. Although this model cannot capture some important macroscopic properties of natural images, such as pixel saturations, it remains analytically tractable and is valid for many natural images.

The left part of Figure 4.3.2 shows the root rate (4.2.1), $r(\lambda)$, for a convex combination of LSB and ±1 embedding, $\lambda \in [0, 1]$, for different image sources. The higher the root rate $r(\lambda)$, the better the stegosystem. The results are consistent with the thesis that ±1 embedding is less detectable than LSB embedding. Similarly, the capacity of stegosystems with covers from NRCS (scans) is believed to be higher than the capacity of stegosystem with decompressed JPEGs or images from digital cameras. This fact is in agreement with our result obtained for all values of the convex combination

[6] Expanded version of CAMERA_RAW database from [52] with 4547 8-bit images.

[7] Contains 2375 raw scans of negatives coming from the USDA Natural Resources Conservation Service (http://photogallery.nrcs.usda.gov).

[8] Images from NRCS database compressed with JPEG quality factor 70.

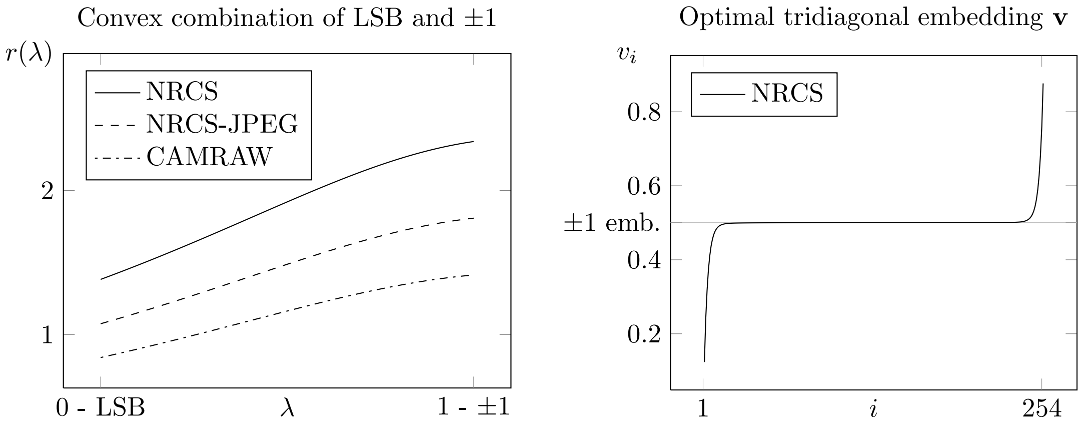

Figure 4.3.2: Left: the root rate $r(\lambda) = \alpha_{MAX}/(\beta_{MAX}\sqrt{I})$ of a convex combination of LSB and ±1 embedding for different image sources. Right: optimal parameters $\mathbf{v} = (v_1, \ldots, v_{254})$ of MI embedding (4.2.2) minimizing the Fisher information rate while modifying cover elements by at most 1. The difference between ±1 embedding and optimal MI embedding is due to boundary effects that vanish as $N \to \infty$.

of LSB and ±1 embedding and we attribute it to the fact that scans contain a higher level of noise that masks embedding changes. In contradiction with our expectations, decompressed JPEGs from NRCS-JPEG have a higher root rate than raw images from digital cameras (CAMRAW). This phenomenon is probably caused by the simplicity of the MC model, which fails to capture JPEG artifacts because they span across larger distances than neighboring pixels.

We now use the methodology described in Section 4.2.2 and maximize the root rate with respect to stegosystems that modify each cover element by at most 1. We do so for the cover model fit obtained from the NRCS database. Assuming the embedding operation is binary, it can embed one bit per cover element. Thus, it is sufficient to find the MI embedding that attains the minimum Fisher information rate. We use the parametrization from Example 4.1 and solve the system of equations (4.2.3). This system has only one solution $\mathbf{v} = (v_1, \ldots, v_{254}) \in \mathcal{V} = [0, 1]^{254}$ and thus it represents MI embedding with minimum Fisher information rate. This solution is shown in the right part of Figure 4.3.2 along with the representation of the ±1 embedding operation. The optimal MI embedding differs from ±1 embedding only at the boundary of the dynamic range. This is due to the finite number of states in the MC model. We experimentally verified that the relative number of states with $|v_i - 0.5| \geq \delta$ tends to zero for a range of $\delta > 0$ as $N \to \infty$ for fixed parameters of the analytic model.[9] Thus, the boundary effect is negligible for large N. This suggests that the loss in capacity when using ±1 embedding algorithm is negligible for large N or, in other words, ±1 embedding is asymptotically optimal.

4.4 Conclusion and Outlook

In sharp contrast with the well established fact that the secure payload of perfectly secure stegosystems increases linearly with the number of cover elements, n, the Square-root law states that the secure payload of a quite wide class of imperfect stegosystems is only proportional to $\sqrt{n}$. The communication rate of imperfect stegosystems is thus non-informative because it tends to zero with n. Instead, an appropriate measure of capacity is the constant of proportionality in front of $\sqrt{n}$, for which we coin the term the *root rate* whose unit is bit per square root of cover size per square root of KL divergence. The root rate is shown to be inversely proportional to the square root of the Fisher information rate of the stegosystem. Adopting a Markov model for the cover source, we derive an analytic formula for the root rate with Fisher information rate expressible as a quadratic

[9] We believe the same to be true for all $\delta > 0$.

form defined by the cover transition probability matrix evaluated at a vector fully determined by the embedding operation. This analytic form is important as it enables us to compare the capacity of imperfect stegosystems as well as optimize their embedding operation (maximize the root rate). We fit a parametric model through the empirical transition probability matrix for neighboring pixels of real images and use this model to compute and compare the root rate of known steganographic schemes and their convex combinations. In agreement with results previously established experimentally using blind steganalyzers, our analysis indicates that ternary ±1 embedding is more secure than LSB embedding and it is also optimal among all embedding methods that modify pixels by at most 1. Furthermore, by analyzing image databases of raw images from different sources, we established that the root rate is larger for images with higher noise level as is to be expected.

Part II

Minimum-Distortion Framework for Near-Optimal Practical Stegosystems

Conceptually, the encoder examines an area of the image and
weights each of the options that allow it to embed the desired bits in that area.
It scores each option for how conspicuous it is and
chooses the option with the best score.

— RON CRANDALL, (1998)

Chapter 5

Gibbs Construction for Steganography

The second part of this dissertation is devoted to practical methods for implementing imperfect stegosystems by minimizing the impact of embedding (also called distortion) in empirical cover sources. The following 3 chapters present a complete framework which Alice and Bob can use when designing new embedding schemes with near-optimal performance.

We start with Chapter 5, where the theoretical part of the embedding framework is described by drawing a connection between steganography and statistical physics. This connection allows us to import many algorithms and thus study embedding schemes minimizing an *arbitrary* distortion function. Due to this relationship, we call the framework *the Gibbs construction*. We show that most embedding methods based on this construction can be realized in practice if Alice and Bob know how to communicate messages by minimizing an arbitrary *additive* distortion function between cover and stego objects. Although this problem is essential and has been known for a long time in steganography, only some of its special forms were solved in the literature. In Chapter 6, we propose the first general solution to this problem which we call the Multi-Layered Syndrome-Trellis Codes (ML-STCs). Finally, in Chapter 7, we combine all the results and propose tools for optimizing the distortion function w.r.t. statistical detectability.

5.1 Introduction

There exist two general and widely used principles for designing steganographic methods for empirical cover objects, such as digital images. The first one is model-preserving steganography in which the designer adopts a model of the cover source and then designs the embedding to either completely or approximately preserve the model [59, 99, 103, 105, 114]. This way, one can provide mathematical guarantee that the embedding is perfectly secure (or ϵ-secure) within the chosen model. A problem is that empirical cover objects are notoriously difficult to model accurately, and, as history teaches us, the model mismatch can be exploited by an attacker to construct a sensitive detection scheme. Even worse, preserving an oversimplified model could introduce a security weakness [10, 76, 126]. An obvious remedy is to use more complicated models that would better approximate the cover source. The major obstacle here is that most current model-preserving steganographic constructions are specific to a certain model and do not adapt easily to more complex models.

The second, quite pragmatic, approach avoids modeling the cover source altogether and, instead, minimizes a heuristically-defined embedding distortion (impact). Matrix embedding [19], wet paper codes [46], and minimal embedding distortion steganography [30, 38, 43, 74, 102] are examples of this philosophy. Despite its heuristic nature, the principle of minimum embedding distortion has produced the most secure steganographic methods for digital media known today, at least in terms of low statistical detectability as measured using blind steganalyzers [47, 74, 79, 102]. Most of these schemes, however, use a distortion function that is additive – the total distortion is a sum of individual pixel distortions *computed from the cover image*. Fundamentally, such a distortion function cannot capture interactions among embedding changes, which leads to suboptimality in practice. This deficiency affects especially adaptive schemes for which the embedding changes have a tendency to form clusters because the pixel distortion is derived from local content or some content-dependent side-information. For example, the embedding changes might follow edges or be concentrated in textured regions.

One discovers a relationship between both embedding principles when the distortion function is defined as a weighted norm of the difference between feature vectors of cover and stego objects in some properly chosen feature space [76, 94], an example of which are spaces utilized by blind steganalyzers. The projection onto the feature space is essentially equivalent to modeling the objects in a lower-dimensional Euclidean space. Consequently, minimizing the distortion between cover and stego objects in the feature space now becomes closely tied to model preservation. Yet again, in this case the distortion cannot be written as a sum of individual pixel distortions also because the features contain higher-order statistics, such as sample transition probability matrices of pixels or DCT coefficients modeled as Markov chains [15, 92, 95, 109].

The importance of modeling interactions among embedding changes in steganography has been indirectly recognized by the designers of MPSteg [13] (Matching Pursuit Steganography) and YASS [104, 113]. In MPSteg, the authors use an overcomplete basis and embed messages by replacing small blocks with other blocks with the hope of preserving dependencies among neighboring pixels. The YASS algorithm taught us that a high embedding distortion may not directly manifest as a high statistical detectability, a curious property that can most likely be attributed to the fact that the embedding modifications are content driven and mutually correlated. Recently, the authors of [121] proposed a modification of ±1 embedding and a heuristic algorithm that minimizes a non-additive distortion function defined as the sum of squared differences between neighboring pixels. All approaches are heuristic in nature and leave many important issues unanswered, including establishing performance bounds, evaluating the methods' performance w.r.t. to these bounds, and creating a methodology for achieving near-optimal performance.

The above discussion underlines the need for a more systematic approach to steganography that can consider mutual interaction of embedding modifications, which is the topic of this chapter. The main contribution is a general framework for embedding using arbitrary distortion functions and a complete practical methodology for minimizing embedding distortion in steganography. The approach is flexible as well as modular and allows the steganographer to work with non-additive distortion functions. We provide algorithms for computing the proper theoretical bounds expressing the maximal payload embeddable with a bounded distortion, for simulating the impact of a stegosystem operating on the bound, and for designing practical steganographic algorithms that operate near the bound. The algorithms leverage standard tools used in statistical physics, such as Markov chain Monte Carlo samplers or the thermodynamic integration.

The technical part of this chapter starts in the next section, where we recall the basic result that embedding changes made by a steganographic method that minimizes embedding distortion must follow a particular form of Gibbs distribution. The main purpose of this section is to establish terminology and make connections between the concepts used in steganography and those in statistical physics. In Section 5.3, we introduce the so-called separation principle, which includes several distinct tasks that must be addressed when developing a practical steganographic method. In particular, to design and evaluate practical schemes one needs to establish the relationship between the maximal payload embeddable using bounded distortion (the rate–distortion bound) and be able to simulate the impact of a scheme operating on the bound. In the special case when the embedding distortion can be expressed as a sum of distortions at individual pixels computed from the cover image (the so-called non-interacting embedding changes), the design of near-optimal embedding algorithms has been successfully resolved in the past and is one of the key contributions of this dissertation covered in Chapter 6. We briefly review this special case in Section 5.4 since it will later allow us to implement the discussed approaches in practice. Continuing with the case of a general distortion function, in Section 5.5 we describe two useful tools for steganographers – the Gibbs sampler and the thermodynamic integration. The Gibbs sampler can be used to simulate the impact of optimal embedding and to construct practical steganographic schemes (in Sections 5.6

and 5.7). The thermodynamic integration is a method for estimating the entropy and partition function in statistical physics and we use it for computing the rate–distortion bound in steganography. The design of practical embedding schemes begins in Section 5.6, where we study distortion functions that can be written as a sum of local potentials defined on cliques. In Section 5.7, we first discuss various options the new framework offers to the steganography designer and then make a connection between local potentials and image models used in blind steganalysis. The proposed framework is experimentally validated in Section 5.8, where we also discuss various implementation issues. Finally, the chapter is concluded in Section 5.9.

5.2 Gibbs Distribution Minimizes Embedding Distortion

We first recall a well-known and quite general fact that, for a given expected embedding distortion, the maximal payload is embedded when the embedding changes follow a Gibbs distribution. This establishes a connection between steganography and statistical physics, which, later in this chapter, will enable us to compute rate–distortion bounds, simulate the impact of optimal embedding, and construct practical embedding algorithms.

Every steganographic embedding scheme considered in this chapter will be associated with a mapping that assigns to each cover $\mathbf{x} \in \mathcal{X}$ the pair $\{\mathcal{Y}, \pi\}$. Here, $\mathcal{Y} \subset \mathcal{X}$ is the set of all stego images $\mathbf{y}$ into which $\mathbf{x}$ is allowed to be modified by embedding and π is a probability mass function on $\mathcal{Y}$ that characterizes the actions of the sender. The embedding algorithm is such that, for a given cover $\mathbf{x}$, the stego image $\mathbf{y} \in \mathcal{Y}$ is sent with probability $\pi(\mathbf{y})$. The stego image is thus a random variable $\mathbf{Y}$ over $\mathcal{Y}$ with the distribution $P(\mathbf{Y} = \mathbf{y}) = \pi(\mathbf{y})$. Technically, the set $\mathcal{Y}$ and all concepts derived from it in this chapter depend on $\mathbf{x}$. However, because $\mathbf{x}$ is simply a parameter that we *fix in the very beginning*, we simplify the notation in this chapter and do not make the dependence on $\mathbf{x}$ explicit. Finally, we note that the maximal expected payload that the sender can communicate to the receiver in this manner is the entropy

$$H(\pi) \triangleq H(\mathbf{Y}) = -\sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) \log_2 \pi(\mathbf{y}). \tag{5.2.1}$$

To put it another way, we define a steganographic method from the point of view of how it modifies the cover and only then we deal with the issues of how to use it for communication and how to optimize its performance. The optimization will involve finding the distribution π for given $\mathbf{x}$, $\mathcal{Y}$, and payload (distortion).

We will consider the following special form of the set $\mathcal{Y}$: $\mathcal{Y} = \mathcal{I}_1 \times \mathcal{I}_2 \times \cdots \times \mathcal{I}_n$, where $\mathcal{I}_i \subset \mathcal{I}$. For example, in Least Significant Bit (LSB) embedding, $\mathcal{I}_i = \{x_i, \overline{x}_i\}$, where the bar denotes the operation of flipping the LSB. In ±1 embedding (also called LSB matching [64]) in an 8-bit grayscale image $\mathbf{x}$, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$ whenever $x_i \notin \{0, 255\}$ and $\mathcal{I}_i$ is appropriately modified for the boundary cases. When $|\mathcal{I}_i| = 2$ or 3 for all i, we will speak of binary and ternary embedding, respectively. In general, however, we allow the size of every set $\mathcal{I}_i$ to be different. For example, pixels not allowed to be modified during embedding (the so-called wet pixels [46]) have $\mathcal{I}_i = \{x_i\}$.

By sending a slightly modified version $\mathbf{y}$ of the cover $\mathbf{x}$, the sender introduces a distortion, which will be measured using a distortion function

$$D : \mathcal{Y} \to \mathbb{R}, \tag{5.2.2}$$

that is bounded, i.e., $|D(\mathbf{y})| < K$, for all $\mathbf{y} \in \mathcal{Y}$ for some sufficiently large K. Note that D also depends on $\mathbf{x}$. Allowing the distortion to be negative does not cause any problems because an embedding algorithm minimizes D if and only if it minimizes the non-negative distortion $D + K$. The need for negative distortion will become apparent later in Section 5.6.1.

The expected embedding distortion introduced by the sender is

$$E_\pi[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}). \tag{5.2.3}$$

An important premise we now make is that the sender is able to define the distortion function so that it is related to statistical detectability – problem we study later in Chapter 7. This assumption is motivated by a rather large body of experimental evidence, such as [47, 79], that indicates that even simple distortion measures that merely count the number of embedding changes correlate well with statistical detectability in the form of decision error of steganalyzers trained on cover and stego images. In general, steganographic methods that introduce smaller distortion disturb the cover source less than methods that embed with larger distortion.

Distortion-limited sender. To maximize the security, the so-called distortion-limited sender attempts to find a distribution π on $\mathcal{Y}$ that has the highest entropy and whose expected embedding distortion does not exceed a given D_ϵ:

$$\underset{\pi}{\text{maximize}} \quad H(\pi) = -\sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) \log_2 \pi(\mathbf{y}) \tag{5.2.4}$$

$$\text{subject to} \quad E_\pi[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}) = D_\epsilon. \tag{5.2.5}$$

By fixing the distortion, the sender fixes the security and aims to communicate as large payload as possible at this level of security. The maximization in (5.2.4) is carried over all distributions π on $\mathcal{Y}$. We will comment on whether the distortion constraint should be in the form of equality or inequality shortly.

Payload-limited sender. Alternatively, in practice it may be more meaningful to consider the payload-limited sender who faces a complementary task of embedding a *given* payload of m bits with minimal possible distortion. The optimization problem is to determine a distribution π that communicates a required payload while minimizing the distortion:

$$\underset{\pi}{\text{minimize}} \quad E_\pi[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \pi(\mathbf{y}) D(\mathbf{y}) \tag{5.2.6}$$

$$\text{subject to} \quad H(\pi) = m. \tag{5.2.7}$$

The optimal distribution π for both problems has the Gibbs form

$$\pi_\lambda(\mathbf{y}) = \frac{1}{Z(\lambda)} \exp(-\lambda D(\mathbf{y})), \tag{5.2.8}$$

where $Z(\lambda)$ is the normalizing factor

$$Z(\lambda) = \sum_{\mathbf{y}\in\mathcal{Y}} \exp(-\lambda D(\mathbf{y})). \tag{5.2.9}$$

The optimality of π_λ follows immediately from the fact that for any distribution μ with $E_\mu[D] = \sum_{\mathbf{y}\in\mathcal{Y}} \mu(\mathbf{y})D(\mathbf{y}) = D_\epsilon$, the difference between their entropies, $H(\pi_\lambda) - H(\mu) = D_{\mathrm{KL}}(\mu||\pi_\lambda) \geq 0$ [127]. The scalar parameter $\lambda > 0$ needs to be determined from the distortion constraint (5.2.5) or from the payload constraint (5.2.7), depending on the type of the sender. Provided m or D_ϵ are in the feasibility region of their corresponding constraints, the value of λ is unique. This follows from the fact that both the expected distortion and the entropy are monotone decreasing in λ. To see this, realize that by direct evaluation

$$\frac{\partial}{\partial\lambda} E_{\pi_\lambda}[D] = -Var_{\pi_\lambda}[D] \leq 0, \tag{5.2.10}$$

where $Var_{\pi_\lambda}[D] = E_{\pi_\lambda}[D^2] - (E_{\pi_\lambda}[D])^2$. Substituting (5.2.8) into (5.2.1), the entropy of the Gibbs distribution can be written as

$$H(\pi_\lambda) = \log_2 Z(\lambda) + \frac{1}{\ln 2} \lambda E_{\pi_\lambda}[D]. \tag{5.2.11}$$

Upon differentiating and using (5.2.10), we obtain

$$\frac{\partial}{\partial\lambda}H(\pi_\lambda) = \frac{1}{\ln 2}\left(\frac{Z'(\lambda)}{Z(\lambda)} + E_{\pi_\lambda}[D] - \lambda Var_{\pi_\lambda}[D]\right) \tag{5.2.12}$$

$$= -\frac{\lambda}{\ln 2}Var_{\pi_\lambda}[D] \leq 0. \tag{5.2.13}$$

The monotonicity also means that the equality distortion constraint in the optimization problem (5.2.5) can be replaced with inequality, which is perhaps more appropriate given the motivating discussion above.

By varying $\lambda \in [0, \infty)$, we obtain a relationship between the maximal expected payload (5.2.1) and the expected embedding distortion (5.2.3). For brevity, we will call this relationship the rate–distortion bound. What distinguishes this concept from a similar notion defined in information theory is that we consider the bound for a *given* cover $\mathbf{x}$ rather than for $\mathbf{X}$, which is a random variable. At this point, we feel that it is appropriate to note that while it is certainly possible to consider $\mathbf{x}$ to be generated by a cover source with a known distribution and approach the design of steganography from a different point of view, namely one in which π_λ is determined by minimizing the KL divergence between the distributions of cover and stego images while satisfying a payload constraint, we do not do so in this work.

Finally, we note that the assumption $|D(\mathbf{y})| < K$ implies that all stego objects appear with non-zero probability, $\pi_\lambda(\mathbf{y}) \geq \frac{1}{Z(\lambda)}\exp(-\lambda K)$, a fact that is crucial for the theory developed in the rest of this work.

Remark 5.1. In statistical physics, the term distortion is known as energy. The optimality of Gibbs distribution is formulated as the Gibbs variational principle: "Among all distributions with a given energy, the Gibbs distribution (5.2.8) has the highest entropy." The parameter λ is called the inverse temperature, $\lambda = 1/kT$, where T is the temperature and k the Boltzmann constant. The normalizing factor $Z(\lambda)$ is called the partition function.

5.3 The Separation Principle

The design of steganographic methods that attempt to minimize embedding distortion should be driven by their performance. The obvious choice here is to contrast the performance with the rate–distortion bound. This is a meaningful comparison for the distortion-limited sender who can assess the performance of a practical embedding scheme by its loss of payload w.r.t. the maximum payload embeddable using a fixed distortion. This so-called "coding loss" informs the sender of how much payload is lost for a fixed statistical detectability. On the other hand, it is much harder for the payload-limited sender to assess how the increased distortion of a suboptimal practical scheme impacts statistical detectability in practice. We could resolve this rather important practical issue if we were able to simulate the impact of a scheme that operates *on the bound.*[1] Because the problems of establishing the bounds, simulating optimal embedding, and creating a practical embedding algorithm are really three separate problems, we call this reasoning the *separation principle.* It involves addressing the following three tasks:

1. **Establishing the rate–distortion bounds.** This means solving the optimization problems (5.2.4) or (5.2.6) and expressing the largest payload embeddable using a bounded distortion (or minimal distortion needed to embed a given payload). These bounds inform the steganographer about the best performance that can be theoretically achieved. Depending on the form of the distortion function D, establishing the bounds is usually rather challenging and one may have to resort to numerical methods (Section 5.5.2). For an additive distortion (to be precisely defined shortly), an analytic form of the bounds may be obtained (Section 5.4).

2. **Simulating an optimal embedding method.** Often, it is very hard to construct a practical embedding method that performs close to the bound. However, we may be able to simulate the

[1] A scheme whose embedding distortion and payload lay on the rate–distortion bound derived for a given cover.

impact of such an optimal method and thus subject it to tests using steganalyzers even when we do not know how to construct a practical embedding algorithm or even compute the bound (see Section 5.5). This is important for developers as one can effectively "prune" the design process and focus on implementing the most promising candidates. The simulator will also inform the payload-limited sender about the potential improvement in statistical undetectability should the theoretical performance gap be closed. A simple example is provided by the case of the Hamming distortion function $D(\mathbf{y}) = \sum_i [y_i \neq x_i]$. Here, the maximal relative payload $\alpha = m/n$ (in bits per pixel or bpp) is bounded by $\alpha \leq h(\beta)$, where $\beta = \frac{1}{n} D_\epsilon$ is the relative embedding distortion known as the change rate. In this case, one can simulate the embedding impact of the optimal scheme by independently changing each pixel with probability $h^{-1}(\alpha)$.

3. **Constructing a practical near-optimal embedding method.** This point is of most interest to practitioners. The bounds and the simulator are necessary to evaluate the performance of any practical scheme. The designer tries to maximize the embedding throughput (the number of bits embedded per unit time) while embedding as close to the distortion bound as possible.

It should be stressed at this point that even though the optimal distribution of embedding modifications has a known analytic expression (5.2.8), it may be infeasible to compute the individual probabilities $\pi_\lambda(\mathbf{y})$ due to the complexity of evaluating the partition function $Z(\lambda)$, which is a sum over all $\mathbf{y}$, whose count can be a very large number even for small images. (For example, there are 2^n binary flipping patterns in LSB embedding.) This also implies that at present we do not know how to compute the expected distortion (5.2.3) or the entropy (5.2.1) (these tasks are postponed to Section 5.5). Fortunately, in many cases of practical interest we do not need to evaluate $\pi_\lambda(\mathbf{y})$ and will do just fine with being able to merely *sample from* π_λ. The ability to sample from π_λ is sufficient to simulate optimal embedding and realize practical embedding algorithms, and, in our case, even compute the rate–distortion bound.

In some special cases, however, such as when the embedding changes do not interact, the distortion D is additive and one can easily compute λ and the probabilities, evaluate the expected distortion and payload, and even construct near-optimal embedding schemes. As this special case will be used later in Section 5.7 to implement schemes with more general distortion functions D, we review it briefly in the next section.

5.4 Non-interacting Embedding Changes

When the distortion function D is additive over the pixels,

$$D(\mathbf{y}) = \sum_{i=1}^{n} \rho_i(y_i), \tag{5.4.1}$$

with bounded $\rho_i : \mathcal{I}_i \to \mathbb{R}$, we say that the embedding changes do not interact. In this case, the probability $\pi_\lambda(\mathbf{y})$ can be factorized into a product of marginal probabilities of changing the individual pixels (this follows directly from (5.2.8)):

$$\pi_\lambda(\mathbf{y}) = \prod_{i=1}^{n} \pi_\lambda(y_i) = \prod_{i=1}^{n} \frac{\exp(-\lambda \rho_i(y_i))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \rho_i(t_i))}. \tag{5.4.2}$$

The expected distortion and the maximal payload are:

$$E_{\pi_\lambda}[D] = \sum_{i=1}^{n} \sum_{t_i \in \mathcal{I}_i} \pi_\lambda(t_i) \rho_i(t_i), \tag{5.4.3}$$

$$H(\pi_\lambda) = -\sum_{i=1}^{n} \sum_{t_i \in \mathcal{I}_i} \pi_\lambda(t_i) \log_2 \pi_\lambda(t_i). \tag{5.4.4}$$

The impact of optimal embedding can be simulated by changing x_i to y_i with probabilities $\pi_\lambda(y_i)$ independently of the changes at other pixels. Since these probabilities can now be easily evaluated for a fixed λ, finding λ that satisfies the distortion ($E_{\pi_\lambda}[D] = D_\epsilon$) or the payload ($H(\pi_\lambda) = m$) constraint amounts to solving an algebraic equation for λ (see [38] or [37]). Because both the expected distortion and the entropy are monotone w.r.t. λ, the solution is unique.

The only practical near-optimal embedding algorithm for this case known to the author is based on syndrome-trellis codes and is described in Chapter 6. For the sake of this chapter, it is sufficient to assume the existence of a practical algorithm which is able to solve both payload- and distortion-limited versions of the embedding problem with additive distortion function (5.4.1) without any need for sharing the original cover $\mathbf{x}$ and the set of distortion functions $\{\rho_i | i \in \{1, \ldots, n\}\}$ with the receiver.

Finally, we note that the complete derivation of the rate–distortion bound for binary embedding appears, e.g., in Chapter 7 of [37].

5.5 Simulated Embedding and Rate–distortion Bound

In Section 5.2, we showed that minimal-embedding-distortion steganography should select the stego image $\mathbf{y}$ with probability $\pi_\lambda(\mathbf{y}) \propto \exp(-\lambda D(\mathbf{y}))$ expressed in the form of a Gibbs distribution. We now explain a general iterative procedure using which one can sample from any Gibbs distribution and thus simulate optimal embedding. The method is recognized as one of the Markov Chain Monte Carlo (MCMC) algorithms known as the Gibbs sampler.[2] This sampling algorithm will allow us to construct practical embedding schemes in Sections 5.6 and 5.7. We also explain how to compute the rate–distortion bound for a fixed image using the thermodynamic integration. The Gibbs sampler and the thermodynamic integration appear, for example, in [127] and [88], respectively.

5.5.1 The Gibbs Sampler

We start by defining the local characteristics of a Gibbs field as the conditional probabilities of the ith pixel attaining the value y'_i conditioned on the rest of the image:

$$\pi_\lambda(Y_i = y'_i | \mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) = \frac{\pi_\lambda(y'_i \mathbf{y}_{\sim i})}{\sum_{t_i \in \mathcal{I}_i} \pi_\lambda(t_i \mathbf{y}_{\sim i})}. \tag{5.5.1}$$

For all possible stego images $\mathbf{y}, \mathbf{y}' \in \mathcal{Y}$, the local characteristics (5.5.1) define the following matrices $\mathbb{P}(i) = (p_{\mathbf{y},\mathbf{y}'}(i))$, for each pixel $i \in \{1, \ldots, n\}$:

$$p_{\mathbf{y},\mathbf{y}'}(i) = \begin{cases} \pi_\lambda(Y_i = y'_i | \mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) & \text{when } \mathbf{y}'_{\sim i} = \mathbf{y}_{\sim i} \\ 0 & \text{otherwise.} \end{cases} \tag{5.5.2}$$

Every matrix $\mathbb{P}(i)$ has $|\mathcal{Y}|$ rows and the same number of columns (which means it is very large) and its elements are mostly zero except when $\mathbf{y}'$ was obtained from $\mathbf{y}$ by modifying y_i to y'_i and all other pixels stayed the same. Because $\mathbb{P}(i)$ is stochastic (the sum of its rows is one),

$$\sum_{\mathbf{y}' \in \mathcal{Y}} p_{\mathbf{y},\mathbf{y}'}(i) = 1, \text{ for all rows } \mathbf{y} \in \mathcal{Y}, \tag{5.5.3}$$

$\mathbb{P}(i)$ is a transition probability matrix of some Markov chain on $\mathcal{Y}$. All such matrices satisfy the so-called detailed balance equation

$$\pi_\lambda(\mathbf{y}) p_{\mathbf{y},\mathbf{y}'}(i) = \pi_\lambda(\mathbf{y}') p_{\mathbf{y}',\mathbf{y}}(i), \quad \text{for all } \mathbf{y}, \mathbf{y}' \in \mathcal{Y},\ i. \tag{5.5.4}$$

[2] More detailed discussion regarding our choice of the MCMC sampler appear later in this section.

Algorithm 5.1 One sweep of a Gibbs sampler.

```
Set pixel counter $i = 1$
while $i \le n$ do
    Compute the local characteristics:
        $p_{\mathbf{y},y'_{\sigma(i)}\mathbf{y}_{\sim\sigma(i)}}(\sigma(i)),\ y'_{\sigma(i)} \in \mathcal{I}_{\sigma(i)}$    (5.5.11)
    Select one $y'_{\sigma(i)} \in \mathcal{I}_{\sigma(i)}$ pseudorandomly according to the probabilities (5.5.11) and change $y_{\sigma(i)} \leftarrow y'_{\sigma(i)}$
    $i \leftarrow i + 1$
end while
return $\mathbf{y}$
```

To see this, realize that unless $\mathbf{y}_{\sim i} = \mathbf{y}'_{\sim i}$, we are looking at the trivial equality $0 = 0$. For $\mathbf{y}_{\sim i} = \mathbf{y}'_{\sim i}$, we have the following chain of equalities:

$$\pi_\lambda(\mathbf{y})p_{\mathbf{y},\mathbf{y}'}(i) \overset{(a)}{=} \pi_\lambda(\mathbf{y})\frac{\pi_\lambda(y'_i\mathbf{y}_{\sim i})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})} \tag{5.5.5}$$

$$\overset{(b)}{=} \frac{\pi_\lambda(\mathbf{y})\pi_\lambda(\mathbf{y}')}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}_{\sim i})} \tag{5.5.6}$$

$$= \pi_\lambda(\mathbf{y}')\frac{\pi_\lambda(\mathbf{y})}{\sum_{t_i\in\mathcal{I}_i}\pi_\lambda(t_i\mathbf{y}'_{\sim i})} \tag{5.5.7}$$

$$\overset{(c)}{=} \pi_\lambda(\mathbf{y}')p_{\mathbf{y}',\mathbf{y}}(i). \tag{5.5.8}$$

Equality (a) follows from the definition of $\mathbb{P}(i)$ (5.5.2), (b) from the fact that $\mathbf{y}_{\sim i} = \mathbf{y}'_{\sim i}$, and (c) from $\pi_\lambda(\mathbf{y}) = \pi_\lambda(y_i\mathbf{y}'_{\sim i})$ and again (5.5.2).

Next, we define the boldface symbol $\boldsymbol{\pi}_\lambda \in [0,\infty)^{|\mathcal{Y}|}$ as the vector of $|\mathcal{Y}|$ non-negative elements $\boldsymbol{\pi}_\lambda = \pi_\lambda(\mathbf{y})$, $\mathbf{y} \in \mathcal{Y}$. Using (5.5.4) and then (5.5.3), we can now easily show that the vector $\boldsymbol{\pi}_\lambda$ is the left eigenvector of $\mathbb{P}(i)$ corresponding to the unit eigenvalue:

$$(\boldsymbol{\pi}_\lambda\mathbb{P}(i))_{\mathbf{y}'} = \sum_{\mathbf{y}\in\mathcal{Y}}\pi_\lambda(\mathbf{y})p_{\mathbf{y},\mathbf{y}'}(i) \tag{5.5.9}$$

$$= \sum_{\mathbf{y}\in\mathcal{Y}}\pi_\lambda(\mathbf{y}')p_{\mathbf{y}',\mathbf{y}}(i) = \pi_\lambda(\mathbf{y}'). \tag{5.5.10}$$

In (5.5.9), $(\boldsymbol{\pi}_\lambda\mathbb{P}(i))_{\mathbf{y}'}$ is the $\mathbf{y}'$th element of the product of the vector $\boldsymbol{\pi}_\lambda$ and the matrix $\mathbb{P}(i)$.

We are now ready to describe the Gibbs sampler [51], which is a key element in our framework. Let σ be a permutation of the index set $\mathcal{S}$ called the visiting schedule ($\sigma(i)$, $i = 1,\ldots,n$ is the ith element of the permutation σ). One sample from π_λ is then obtained by repeating a series of "sweeps" defined below. As we explain the sweeps and the Gibbs sampler, the reader is advised to inspect Algorithm 5.1 to better understand the process.

The sampler is initialized by setting $\mathbf{y}$ to some initial value. For faster convergence, a good choice is to select y_i from $\mathcal{I}_i$ according to the local characteristics $\pi_\lambda(y_i\mathbf{x}_{\sim i})$. A sweep is a procedure applied to an image during which all pixels are updated sequentially in the order defined by the visiting schedule σ. The pixels are updated based on their local characteristics (5.5.1) computed from the current values of the stego image $\mathbf{y}$. The entire sweep can be described by a transition probability matrix $\mathbb{P}(\sigma) \triangleq (p_{\mathbf{y},\mathbf{y}'}(\sigma))$ obtained by matrix-multiplications of the individual transition probability matrices $\mathbb{P}(\sigma(i))$:

$$p_{\mathbf{y},\mathbf{y}'}(\sigma) \triangleq (\mathbb{P}(\sigma(1))\cdot\mathbb{P}(\sigma(2))\cdots\mathbb{P}(\sigma(n)))_{\mathbf{y},\mathbf{y}'}\,. \tag{5.5.12}$$

After each sweep, the next sweep continues with the current image $\mathbf{y}$ as its starting position. It should be clear from the algorithm that at the end of each sweep each pixel i has a non-zero probability to get into any of its states from $\mathcal{I}_i$ defined by the embedding operation (because D is bounded). This means that all elements of $\mathcal{Y}$ will be visited with positive probability and thus the transition probability matrix $\mathbb{P}(\sigma)$ corresponds to a homogeneous irreducible Markov process with a *unique* left eigenvector corresponding to a unit eigenvalue (unique stationary distribution). Because $\boldsymbol{\pi}_\lambda$ is a left eigenvector corresponding to a unit eigenvalue for each matrix $\mathbb{P}(i)$, it is also a left eigenvector for $\mathbb{P}(\sigma)$ and thus its stationary distribution due to its uniqueness. A standard result from the theory of Markov chains (see, e.g. Chapter 4 in [127]) states that, for an irreducible Markov chain, no matter what distribution of embedding changes $\boldsymbol{\nu} \in [0,\infty)^{|\mathcal{Y}|}$ we start with, and independently of the visiting schedule σ, with increased number of sweeps, k, the distribution of Gibbs samples converges in norm to the stationary distribution $\boldsymbol{\pi}_\lambda$:

$$||\boldsymbol{\nu}\left(\mathbb{P}(\sigma)\right)^k - \boldsymbol{\pi}_\lambda|| \to 0 \text{ with } k \to \infty \tag{5.5.13}$$

exponentially fast. This means that in practice we can obtain a sample from π_λ after running the Gibbs sampler for a sufficiently long time.[3] The visiting schedule can be randomized in each sweep as long as each pixel has a non-zero probability of being visited, which is a necessary condition for convergence.

5.5.2 Simulating Optimal Embedding

When applied to steganography, the Gibbs sampler allows the sender to simulate the effect of embedding using a scheme that operates on the bound. It is interesting that this can be done for any distortion function D and without knowing the rate–distortion bound. This is because the local characteristics (5.5.1)

$$\pi_\lambda(Y_i = y_i'|\mathbf{Y}_{\sim i} = \mathbf{y}_{\sim i}) = \frac{\exp(-\lambda D(y_i'\mathbf{y}_{\sim i}))}{\sum_{t_i\in\mathcal{I}_i}\exp(-\lambda D(t_i\mathbf{y}_{\sim i}))}, \tag{5.5.14}$$

do not require computing the partition function $Z(\lambda)$. We do need to know the parameter λ, though.

For the distortion-limited sender (5.2.5), the Gibbs sampler could be used directly to determine the proper value of λ in the following manner. For a given λ, it is known (Theorem 5.1.4 in [127]) that

$$\frac{1}{k}\sum_{j=1}^{k} D\big(\mathbf{y}^{(j)}\big) \to E_{\pi_\lambda}[D] \text{ as } k \to \infty \tag{5.5.15}$$

in L_2 and in probability, where $\mathbf{y}^{(j)}$ is the image obtained after the jth sweep of the Gibbs sampler. This requires running the Gibbs sampler and averaging the individual distortions for a sufficiently long time. When only a finite number of sweeps is allowed, the first few images $\mathbf{y}$ should be discarded to allow the Gibbs sampler to converge close enough to π_λ. The value of λ that satisfies $E_{\pi_\lambda}[D] = D_\epsilon$ can be determined, for example, using a binary search over λ.

To find λ for the payload-limited sender (5.2.4), we need to evaluate the entropy $H(\pi_\lambda)$, which can be obtained from $E_{\pi_\lambda}[D]$ using the method of thermodynamic integration [88]. From (5.2.10) and (5.2.13), we obtain

$$\frac{\partial}{\partial\lambda}H(\pi_\lambda) = \frac{\lambda}{\ln 2}\frac{\partial}{\partial\lambda}E_{\pi_\lambda}[D]. \tag{5.5.16}$$

Therefore, the entropy can be estimated from $E_{\pi_\lambda}[D]$ by integrating by parts:

$$H(\pi_\lambda) = H(\pi_{\lambda_0}) + \left[\frac{\lambda'}{\ln 2}E_{\pi_{\lambda'}}[D]\right]_{\lambda_0}^{\lambda} - \frac{1}{\ln 2}\int_{\lambda_0}^{\lambda} E_{\pi_{\lambda'}}[D]\mathrm{d}\lambda'. \tag{5.5.17}$$

The value of λ that satisfies the entropy (payload) constraint can be again obtained using a binary search. Having obtained the expected distortion and the entropy using the Gibbs sampler and the

[3] The convergence time may vary significantly depending on the Gibbs field at hand.

thermodynamic integration, the rate–distortion bound $[H(\pi_\lambda), E_{\pi_\lambda}[D]]$ can be plotted as a curve parametrized by λ.

In practice, one has to be careful when using (5.5.15), since no practical guidelines exist for determining a sufficient number of sweeps and heuristic criteria are often used [18, 127]. Although the convergence to π_λ is exponential in the number of sweeps, in general a large number of sweeps may be needed to converge close enough. Generally speaking, the stronger the dependencies between embedding changes the more sweeps are needed by the Gibbs sampler. In theory, the convergence of MCMC methods, such as the Gibbs sampler, may also slow down in the vicinity of "phase transitions," which we loosely define here as sudden changes in the spatial distribution of embedding changes when only slightly changing the payload (or distortion bound).

In our experiments reported later in this chapter, the Gibbs sampler always behaved well and converged fast. We attribute this to the fact that the dependencies among embedding modifications as measured using our distortion functions are rather weak and limited to short distances. The convergence, however, could become an issue for other types of cover sources with different distortion functions. While it is possible to compute the rate–distortion bounds and simulate optimal embedding using other MCMC algorithms, such as the Metropolis-Hastings sampler [127], that may converge faster than the Gibbs sampler and can exhibit a more robust behavior in practice, it is not clear how to adopt these algorithms for practical embedding. This is because all known coding methods in steganography essentially sample from a distribution of independent symbols. Thus, the Gibbs sampler comes out as a natural choice (Section 5.6) because it works by updating individual pixels, which is exactly the effect of algorithms working with non-interacting embedding changes we will describe in the next chapter.

A notable alternative to the Gibbs sampler and the thermodynamic integration for computing the rate–distortion bound is the Wang–Landau algorithm [123] that estimates the so-called density of stego images (density of states in statistical physics), $g(D)$, defined as the number of stego images $\mathbf{y}$ with distortion (energy) D. The partition function (and thus, via (5.2.11), the entropy) and the expected distortion can be computed from $g(D)$ by numerical integration:

$$Z(\lambda) \doteq \sum_{D\in\mathcal{D}} g(D)\exp(-\lambda D)\Delta, \tag{5.5.18}$$

$$E_{\pi_\lambda}[D] \doteq \frac{1}{Z(\lambda)} \sum_{D\in\mathcal{D}} Dg(D)\exp(-\lambda D)\Delta, \tag{5.5.19}$$

where $\mathcal{D} = \{d_1, \ldots, d_{n_D}\}$, $d_1 = -K$, $d_{n_D} = K$, $d_i - d_{i-1} = \Delta$ is a set of discrete values into which the dynamic range of D, $[-K, K]$ is quantized.

It should be noted that in general it is not possible to determine ahead of time which method will provide satisfactory performance. In our experiments described in Section 5.8, the thermodynamic integration worked very well and provided results identical to the much more complex Wang–Landau algorithm.

Note that computing the rate–distortion bound is not necessary for practical embedding. In Section 5.6, we introduce a special form of the distortion in terms of a sum over local potentials. In this case, both types of optimal senders can be simulated using algorithms that do not need to compute λ in the fashion described above. This is explained in Sections 5.6.1 and 5.6.2.

5.6 Local Distortion Function

Thanks to the Gibbs sampler, we can simulate the impact of embedding that is optimal in the sense of (5.2.4) and (5.2.6) without having to construct a specific steganographic scheme. This is important for steganography design as we can test the effect of various design choices and parameters and then implement only the most promising constructs. However, it is rather difficult to design near-optimal schemes for a general $D(\mathbf{y})$. Fortunately, it is possible to give the distortion function a specific form that will allow us to construct practical embedding algorithms. We will assume that D is a sum of local potentials defined on small groups of pixels called cliques. This local form of

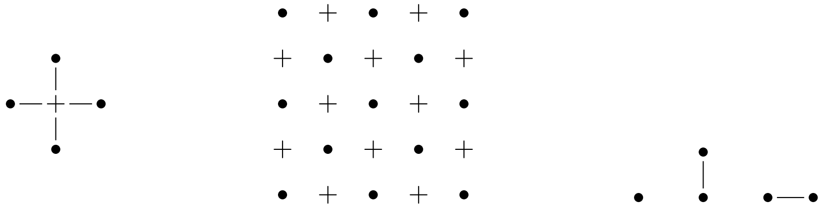

Figure 5.6.1: Left: The four-element cross-neighborhood. Center: Tessellation of the index set $\mathcal{S}$ into two disjoint sublattices $\mathcal{S}_e$ and $\mathcal{S}_o$. Right: All three possible cliques for the cross-neighborhood.

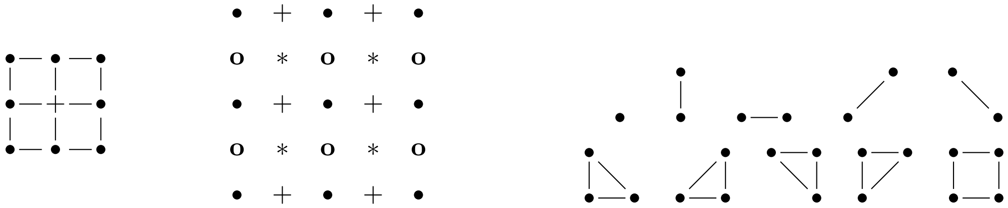

Figure 5.6.2: Left: The eight-element neighborhood. Center: Tessellation of the index set $\mathcal{S}$ into four disjoint sublattices marked with four different symbols. Right: All possible cliques for the eight-element neighborhood.

the distortion will be still quite general to capture dependencies among embedding changes and it allows us to construct a large spectrum of diverse embedding schemes – a topic left for Section 5.7.

First, we define a neighborhood system as a collection of subsets of the index set $\{\eta(i) \subset \mathcal{S} | i = 1, \ldots, n\}$ satisfying $i \notin \eta(i), \forall i$ and $i \in \eta(j)$ if and only if $j \in \eta(i)$. The elements of $\eta(i)$ are called neighbors of pixel i. A subset $c \subset \mathcal{S}$ is a clique if each pair of different elements from c are neighbors. The set of all cliques will be denoted $\mathcal{C}$. We do not use the calligraphic font for a clique even though it is a set (and thus deviate here from our convention) to comply with a well established notation used in previous art.

In this section and in Section 5.7, we will need to address pixels by their two-dimensional coordinates. We will thus be switching between using the index set $\mathcal{S} = \{1, \ldots, n\}$ and its two-dimensional equivalent $\mathcal{S} = \{(i,j) | 1 \leq i \leq n_1, 1 \leq j \leq n_2\}$ hoping that it will cause no confusion for the reader.

Example 5.1. The four-element cross neighborhood of pixel $x_{i,j}$ consisting of $\{x_{i-1,j}, x_{i+1,j}, x_{i,j-1}, x_{i,j+1}\}$ with a proper treatment at the boundary forms a neighborhood system (see Figure 5.6.1). The cliques contain either a single pixel (one-element) cliques $\{x_{i,j}\}$ or two horizontally or vertically neighboring pixels, $\{x_{i,j}, x_{i,j+1}\}$, $\{x_{i,j}, x_{i+1,j}\}$ (Figure 5.6.1). No other cliques exist.

Example 5.2. The eight-element 3×3 neighborhood also forms a neighborhood system (Figure 5.6.2). The cliques are as in Example 5.1 as well as all cliques containing pairs of diagonally neighboring pixels, $\{x_{i,j}, x_{i+1,j+1}\}$, $\{x_{i,j}, x_{i-1,j+1}\}$, three-pixel cliques forming a right-angle triangle (e.g., $\{x_{i,j}, x_{i,j+1}, x_{i+1,j}\}$), and four-pixel cliques forming a 2×2 square ($\{x_{i,j}, x_{i,j+1}, x_{i+1,j}, x_{i+1,j+1}\}$) (follow Figure 5.6.2). No other cliques exist for this neighborhood system.

Each neighborhood system allows tessellation of the index set $\mathcal{S}$ into disjoint subsets (sublattices) whose union is the entire set $\mathcal{S}$, so that any two pixels in each lattice are not neighbors. For example, for the cross-neighborhood $\mathcal{S} = \mathcal{S}_e \cup \mathcal{S}_o$, where

$$\mathcal{S}_e = \{(i,j) | i+j \text{ is even}\}, \qquad \mathcal{S}_o = \{(i,j) | i+j \text{ is odd}\}. \tag{5.6.1}$$

For the eight-element 3×3 neighborhood, there are four sublattices, $\mathcal{S} = \bigcup_{ab} \mathcal{S}_{ab}$, $1 \leq a, b \leq 2$, whose structure resembles the Bayer color filter array commonly used in digital cameras [37],

$$\mathcal{S}_{ab} = \{(a+2k, b+2l) | 1 \leq a+2k \leq n_1, 1 \leq b+2l \leq n_2\}. \tag{5.6.2}$$

For a clique $c \in \mathcal{C}$, we denote by $V_c(\mathbf{y})$ the local potential, which is an arbitrary bounded function that depends only on the values of $\mathbf{y}$ in the clique c, $V_c(\mathbf{y}) = V_c(\mathbf{y}_c)$. We remind that V_c may also depend on $\mathbf{x}$ in an arbitrary fashion. We are now ready to introduce a local form of the distortion function as

$$D(\mathbf{y}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{y}_c). \tag{5.6.3}$$

The important fact is that D is a sum of functions with a small support. Let us express the local characteristics (5.5.1) in terms of this newly-defined form (5.6.3):

$$\pi_\lambda(Y_i = y_i'|\mathbf{y}_{\sim i}) = \frac{\exp(-\lambda \sum_{c \in \mathcal{C}} V_c(y_i'\mathbf{y}_{\sim i}))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \sum_{c \in \mathcal{C}} V_c(t_i\mathbf{y}_{\sim i}))} \tag{5.6.4}$$

$$\overset{(a)}{=} \frac{\exp(-\lambda \sum_{c \in \mathcal{C}(i)} V_c(y_i'\mathbf{y}_{\sim i}))}{\sum_{t_i \in \mathcal{I}_i} \exp(-\lambda \sum_{c \in \mathcal{C}(i)} V_c(t_i\mathbf{y}_{\sim i}))}, \tag{5.6.5}$$

where $\mathcal{C}(i) = \{c \in \mathcal{C} | i \in c\}$, $i = 1, \ldots, n$. Equality (a) holds because $V_c(t_i\mathbf{y}_{\sim i})$ does not depend on t_i for cliques $c \notin \mathcal{C}(i)$ as they do not contain the ith element. Thus, the terms V_c for such cliques cancel from (5.6.5). This has a profound impact on the local characteristics, making the realization of Y_i conditionally *independent* of changes made outside of the union of cliques containing pixel i and thus outside of the neighborhood $\eta(i)$ given the neighborhood pixel values. For the cross-neighborhood system from Example 5.1, changes made to pixels belonging to the sublattice $\mathcal{S}_e$ do not interact and thus the Gibbs sampler can be parallelized by first updating *all* pixels from this sublattice in parallel and then updating in parallel *all* pixels from $\mathcal{S}_o$.[4]

The possibility to update all pixels in each sublattice all at once provides a recipe for constructing practical embedding schemes. Assume $\mathcal{S} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_s$ with mutually disjoint sublattices. We first describe the actions of a payload-limited sender (follow the pseudo-code in Algorithm 5.2).

5.6.1 Payload-limited Sender

The sender divides the payload of m bits into s equal parts of m/s bits, computes the local distortions

$$\rho_i(y_i'\mathbf{y}_{\sim i}) = \sum_{c \in \mathcal{C}(i)} V_c(y_i'\mathbf{y}_{\sim i}) \tag{5.6.6}$$

for pixels $i \in \mathcal{S}_1$, and embeds the first message part in $\mathcal{S}_1$. Then, it updates the local distortions of all pixels from $\mathcal{S}_2$ and embeds the second part in $\mathcal{S}_2$, updates the local distortions again, embeds the next part in $\mathcal{S}_3$, etc. Because the embedding changes in each sublattice do not interact, the embedding can be realized as discussed in Section 5.4. After all sublattices are processed, we say that one embedding sweep was completed. By repeating these embedding sweeps,[5] the resulting modified images will converge to a sample from π_λ.

The embedding in sublattice $\mathcal{S}_k$ will introduce embedding changes with probabilities (5.4.2), where the value of λ_k is determined by the individual distortions $\{\rho_i(y_i'\mathbf{y}_{\sim i}) | i \in \mathcal{S}_k\}$ (5.6.6) to satisfy the payload constraint of embedding m/s bits in the kth sublattice (again, e.g., using a binary search for λ_k). Because each sublattice extends over a different portion of the cover image while we split the payload evenly across the sublattices, λ_k may slightly vary with k because of variations in the individual distortions. This represents a deviation from the Gibbs sampler. Fortunately, the sublattices can often be chosen so that the image does not differ too much on every sublattice, which will guarantee that the sets of individual distortions $\{\rho_i(y_i'\mathbf{y}_{\sim i}) | i \in \mathcal{S}_k\}$ are also similar across the sublattices. Thus, with an increased number of sweeps, λ_k will converge to an approximately common value and the whole process represents a correct version of the Gibbs sampler.

[4] The Gibbs random field described by the joint distribution $\pi_\lambda(\mathbf{y})$ with distortion (5.6.3) becomes a Markov random field on the same neighborhood system. This follows from the Hammersley-Clifford theorem [127].

[5] After each embedding sweep, at each pixel the previous change is *erased* and the pixel is reconsidered again, just like in the Gibbs sampler.

Algorithm 5.2 One sweep of a Gibbs sampler for embedding m-bit message (payload-limited sender).

```
Require: $\mathcal{S} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_s$ {mutually disjoint sublattices}
for $k = 1$ to $s$ do
    for every $i \in \mathcal{S}_k$ do
        Use (5.6.6) to calculate cost of changing $y_i \to y'_i \in \mathcal{I}_i$
    end for
    Embed $m/s$ bits while minimizing $\sum_{i\in\mathcal{S}_k} \rho_i(y'_i\mathbf{y}_{\sim i})$.
    Update $\mathbf{y}_{\mathcal{S}_k}$ with new values and keep $\mathbf{y}_{\sim\mathcal{S}_k}$ unchanged.
end for
return $\mathbf{y}$
```

Algorithm 5.3 One sweep of a Gibbs sampler for a distortion-limit sender, $E_{\pi_\lambda}[D] = D_\epsilon$.

```
Require: $\mathcal{S} = \mathcal{S}_1 \cup \ldots \cup \mathcal{S}_s$ {mutually disjoint sublattices}
for $k = 1$ to $s$ do
    for every $i \in \mathcal{S}_k$ do
        Use (5.6.6) to calculate cost of changing $y_i \to y'_i \in \mathcal{I}_i$
    end for
    Embed $m_k$ bits while $\sum_i \rho_i(y'_i\mathbf{y}_{\sim i}) = D_\epsilon \times |\{c \in \mathcal{C}|c \cap \mathcal{S}_k \neq \emptyset\}|/|\mathcal{C}|$.
    Update $\mathbf{y}_{\mathcal{S}_k}$ with new values and keep $\mathbf{y}_{\sim\mathcal{S}_k}$ unchanged.
end for
return $\mathbf{y}$ and $\sum_k m_k$ {stego image and number of bits}
```

In binary embedding ($\mathcal{I}_i = \{x_i^{(0)}, x_i^{(1)}\}$), note that the two distortions $\rho_i^{(0)}(x_i^{(0)}\mathbf{y}_{\sim i}) = D(x_i^{(0)}\mathbf{y}_{\eta(i)})$, $\rho_i^{(1)}(x_i^{(1)}\mathbf{y}_{\sim i}) = D(x_i^{(1)}\mathbf{y}_{\eta(i)})$ at pixel i depend on the current pixel values in its neighborhood $\eta(i)$. Therefore, both $\rho_i^{(0)}$ and $\rho_i^{(1)}$ can be non-zero at the same time and we can even have $\rho_i^{(1)} < \rho_i^{(0)}$. It is the neighborhood of i that ultimately determines whether or not it is beneficial to preserve the value of the pixel!

5.6.2 Distortion-limited Sender

A similar approach can be used to implement the distortion-limited sender with a distortion limit D_ϵ. Consider a simulation of such embedding by a Gibbs sampler with the correct λ (obtained from a binary search as described in Section 5.5.2) on the sublattice $\mathcal{S}_k \subset \mathcal{S}$. Assuming again that all sublattices have the same distortion properties, the distortion obtained from cliques containing pixels from $\mathcal{S}_k$ should be proportional to the number of such cliques. Formally,

$$E_{\pi_\lambda(\mathbf{Y}_{\mathcal{S}_k}|\mathbf{Y}_{\sim\mathcal{S}_k})}[D] = D_\epsilon \frac{|\{c \in \mathcal{C}|c \cap \mathcal{S}_k \neq \emptyset\}|}{|\mathcal{C}|}. \tag{5.6.7}$$

As described in Algorithm 5.3, the sender can realize this by embedding as many bits to every sublattice as possible while achieving the distortion (5.6.7). Note that we do not need to compute the partition function for every image in order to realize the embedding. Moreover, in practice when the embedding is implemented using syndrome-trellis codes (see Chapter 6), the search for the correct parameter λ, as described in Section 5.5.2, is not needed either as long as the distortion properties of every sublattice are the same. This is because the codes need the local *distortion* $\rho_i(y'_i\mathbf{y}_{\sim i})$ (5.6.6) at each lattice pixel i and not the embedding probabilities. (This eliminates the need for λ.)

The issue of the minimal sufficient number of embedding sweeps for both algorithms needs to be studied specifically for each distortion measure (see the discussion in the experimental Section 5.8). By replacing a specific practical embedding method with a simulator of optimal embedding, we can simulate the impact of optimal algorithms (for both senders) without having to determine the value of the parameter λ as described in Section 5.5.2. We still need to compute λ_k for each sublattice

$\mathcal{S}_k$ to obtain the probabilities of modifying each pixel (5.4.2), but this can be done as described in Section 5.4 without having to use the Gibbs sampler or the thermodynamic integration.

Finally, we comment on how to handle wet pixels within this framework. Since we assume that the distortion is bounded ($|D(\mathbf{y})| < K$ for all $\mathbf{y} \in \mathcal{Y}$), wet pixels are handled by forcing $\mathcal{I}_i = \{x_i\}$. Because this knowledge may not be available to the decoder in practice, practical coding schemes should treat them either by setting $\rho_i(y_i) = \infty$ or to some large constant for $y_i \neq x_i$ (we provide all the details in Section 6.3.5).

5.6.3 Practical Limits of the Gibbs Sampler

Thanks to the bounds established in Section 5.2, we know that the maximal payload that can be embedded in this manner is the entropy of π_λ (5.2.11). Assuming the embedding proceeds on the bound for the individual sublattices, the question is how close the total payload embedded in the image is to $H(\pi_\lambda)$. Following the Gibbs sampler, the configuration of the stego image will converge to a sample $\mathbf{y}$ from π_λ. Let us now go through one more sweep. We denote by $\mathbf{y}^{[k]}$ the stego image before starting embedding in sublattice $\mathcal{S}_k$, $k = 1, \ldots, s$. In each sublattice, the following payload is embedded:

$$H\big(\mathbf{Y}_{\mathcal{S}_k}\big|\mathbf{Y}_{\sim\mathcal{S}_k} = \mathbf{y}^{[k]}_{\sim\mathcal{S}_k}\big). \tag{5.6.8}$$

We now use the following result from information theory. For any random variables $X_1, \ldots, X_s$,

$$\sum_{k=1}^{s} H(X_k|X_{\sim k}) \leq H(X_1, \ldots, X_s), \tag{5.6.9}$$

with equality only when all variables are independent.[6] Thus, we will have in general

$$H^{-}(\mathbf{Y}) \triangleq \sum_{k=1}^{s} H\big(\mathbf{Y}_{\mathcal{S}_k}\big|\mathbf{Y}_{\sim\mathcal{S}_k} = \mathbf{y}^{[k]}_{\sim\mathcal{S}_k}\big) < H(\mathbf{Y}) = H(\pi_\lambda). \tag{5.6.10}$$

The term $H^{-}(\mathbf{Y})$ is recognized as the erasure entropy [118, 119] and it is equal to the conditional entropy $H(\mathbf{Y}^{(l+1)}|\mathbf{Y}^{(l)})$ (entropy rate) of the Markov process defined by our Gibbs sampler (c.f., (5.5.12)), where $\mathbf{Y}^{(l)}$ is the random variable obtained after l sweeps of the Gibbs sampler.

The erasure-entropy inequality (5.6.10) means that the embedding scheme will be suboptimal, unable to embed the maximal payload $H(\pi_\lambda)$. The actual loss can be assessed by evaluating the entropy of $H(\pi_\lambda)$, e.g., using the algorithms described in Section 5.5. An example of such comparison is presented in Section 5.8.4.

The last remaining issue is the choice of the potentials V_c. In the next section, we show one example, where V_c are chosen to tie the principle of minimal embedding distortion to the preservation of the cover-source model. We also describe a specific embedding method and subject it to experiments using blind steganalyzers.

5.7 Practical Embedding Constructions

We are now in the position to describe a practical embedding method that uses the theory developed so far. First and foremost, the potentials V_c should measure the detectability of embedding changes. We have substantial freedom in choosing them and the design may utilize reasoning based on theoretical cover source models as well as heuristics stemming from experiments using blind steganalyzers. The proper design of potentials is a complicated subject in itself and opens space for further research. The purpose of this chapter is to introduce a general framework rather than optimizing the design, which is covered in Chapter 7. Here, we describe a specific example of a more general approach that builds upon the latest results in steganography and steganalysis and one that gave us an opportunity to validate the proposed framework by showing an improvement over the current state of the art in Section 5.8.

[6] For $k = 2$, this result follows immediately from $H(X_1|X_2) + H(X_2|X_1) = H(X_1, X_2) - I(X_1; X_2)$. The result for $s > 2$ can be obtained by induction over s.

5.7.1 Additive Approximation

As argued in the introduction, the steganography design principles based on model preservation and on minimizing distortion coincide when the distortion is defined as a norm of the difference of feature vectors used to model cover images:

$$D(\mathbf{y}) = ||f(\mathbf{x}) - f(\mathbf{y})|| \triangleq \sum_{k=1}^{d} w_k |f_k(\mathbf{x}) - f_k(\mathbf{y})|. \tag{5.7.1}$$

Here, $f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_d(\mathbf{x})) \in \mathbb{R}^d$ is a d-dimensional feature vector of image $\mathbf{x}$ and $\mathbf{w} = (w_1, \ldots, w_d)$ are weights. The properties of D defined in this manner depend on the properties of the functions f_k. In general, however, D is not additive. In the past, steganographers were forced to use some *additive approximation* of D to realize the embedding in practice. A general method for turning an arbitrary distortion measure into an additive proceeds is:

$$\hat{D}(\mathbf{y}) = \sum_{i=1}^{n} D(y_i \mathbf{x}_{\sim i}). \tag{5.7.2}$$

Embedding with the additive measure $\hat{D}$ can be simulated (and realized) as explained in Section 5.4. The approximation, of course, ensues a capacity loss due to a mismatch in the minimized distortion function. Thanks to the methods introduced in Section 5.5.2, this loss can now be contrasted against the rate–distortion bound for the original measure D. However, we cannot build a practical scheme unless D can be written as a sum of *local* potentials. Next, we explain how to turn D into this form using the idea of a bounding distortion.

5.7.2 Bounding Distortion

Most features used in steganalysis can be written as a sum of locally-supported functions across the image

$$f_k(\mathbf{x}) = \sum_{c \in \mathcal{C}} f_c^{(k)}(\mathbf{x}), \quad k = 1, \ldots, d. \tag{5.7.3}$$

For example, the kth histogram bin of image $\mathbf{x}$ can be written using the Iverson bracket as

$$h_k(\mathbf{x}) = \sum_{i \in \mathcal{S}} [x_i = k], \tag{5.7.4}$$

while the klth element of a horizontal co-occurrence matrix $\mathbb{G}(\mathbf{x}) = (g_{k,l}(\mathbf{x}))$ as

$$g_{k,l}(\mathbf{x}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2 - 1} [x_{i,j} = k][x_{i,j+1} = l] \tag{5.7.5}$$

is a sum over horizontally adjacent pixels (horizontal two-pixel cliques). For such locally-supported features, we can obtain an upper bound on $D(\mathbf{y}) = ||f(\mathbf{x}) - f(\mathbf{y})||$, $\mathbf{y} \in \mathcal{Y}$, that has the required form:

$$||f(\mathbf{x}) - f(\mathbf{y})|| = \sum_{k=1}^{d} w_k \left| \sum_{c \in \mathcal{C}} f_c^{(k)}(\mathbf{x}) - \sum_{c} f_c^{(k)}(\mathbf{y}) \right| \tag{5.7.6}$$

$$\leq \sum_{k=1}^{d} w_k \sum_{c \in \mathcal{C}} |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})| \tag{5.7.7}$$

$$= \sum_{c \in \mathcal{C}} \sum_{k=1}^{d} w_k |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})| \tag{5.7.8}$$

$$= \sum_{c \in \mathcal{C}} V_c(\mathbf{y}), \tag{5.7.9}$$

where

$$V_c(\mathbf{y}) = \sum_{k=1}^{d} w_k |f_c^{(k)}(\mathbf{x}) - f_c^{(k)}(\mathbf{y})|. \tag{5.7.10}$$

Following our convention explained in Section 5.2, we describe the methodology for a fixed cover image $\mathbf{x}$ and thus do not make the dependence of V_c on $\mathbf{x}$ explicit. The sum $\sum_{c \in \mathcal{C}} V_c(\mathbf{y})$ will be called the *bounding distortion.*

We now provide a specific example of this approach. The choice is motivated by our desire to work with a modern, well-established feature set so that later, in Section 5.8, we can validate the usefulness of the proposed framework by constructing a high-capacity steganographic method undetectable using current state-of-the-art steganalyzer. Similarly as in the design of steganalytic features, our goal is to capture the sensitivity of different cliques to embedding. Similarly as in the HUGO algorithm [94], we form the cliques from k-pixel lines for some small k in different orientations. The construction is a slight modification of the SPAM set [92] described in Section 2.3.1, which is the basis of the current most reliable blind steganalyzer in the spatial domain. The features are constructed by considering the differences between neighboring pixels (e.g., horizontally adjacent pixels) as a higher-order Markov chain and taking the sample joint probability matrix (co-occurrence matrix) as the feature. The advantage of using the joint matrix instead of the transition probability matrix is that the norm of the feature difference can be readily upper-bounded by the desired local form (5.7.10).

To formally define the feature for an $n_1 \times n_2$ image $\mathbf{x}$, let us consider the following co-occurrence matrix $\mathbb{G}^{\rightarrow}(\mathbf{x}) = (g^{\rightarrow}_{k,l}(\mathbf{x}))$ computed from horizontal pixel differences, $\mathbb{D}^{\rightarrow}(\mathbf{x}) = (d^{\rightarrow}_{i,j}(\mathbf{x}))$, $d^{\rightarrow}_{i,j}(\mathbf{x}) = x_{i,j+1} - x_{i,j}$, for $i = 1, \ldots, n_1$, and $j = 1, \ldots, n_2 - 1$:

$$g^{\rightarrow}_{k,l}(\mathbf{x}) = \frac{1}{n_1(n_2-2)} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2-2} [(d^{\rightarrow}_{i,j}, d^{\rightarrow}_{i,j+1})(\mathbf{x}) = (k,l)]. \tag{5.7.11}$$

Clearly, $g^{\rightarrow}_{i,j}(\mathbf{x})$ is the normalized count of neighboring triples of pixels $(x_{i,j}, x_{i,j+1}, x_{i,j+2})$ with differences $x_{i,j+1} - x_{i,j} = k$ and $x_{i,j+2} - x_{i,j+1} = l$ in the entire image. The superscript arrow "→" denotes the fact that the differences are computed by subtracting the left pixel from the right one. Similarly,

$$g^{\leftarrow}_{k,l}(\mathbf{x}) = \frac{1}{n_1(n_2-2)} \sum_{i=1}^{n_1} \sum_{j=3}^{n_2} [(d^{\leftarrow}_{i,j}, d^{\leftarrow}_{i,j-1})(\mathbf{x}) = (k,l)] \tag{5.7.12}$$

with $d^{\leftarrow}_{i,j}(\mathbf{x}) = x_{i,j-1} - x_{i,j}$. By analogy, we can define vertical, diagonal, and minor diagonal matrices $\mathbb{G}^{\downarrow}$, $\mathbb{G}^{\uparrow}$, $\mathbb{G}^{\nearrow}$, $\mathbb{G}^{\swarrow}$, $\mathbb{G}^{\searrow}$, $\mathbb{G}^{\nwarrow}$. All eight matrices are sample joint probabilities of observing the differences k and l between three consecutive pixels along a certain direction. Due to the antisymmetry $d^{\rightarrow}_{i,j}(\mathbf{x}) = -d^{\leftarrow}_{i,j+1}(\mathbf{x})$ only $\mathbb{G}^{\rightarrow}$, $\mathbb{G}^{\nearrow}$, $\mathbb{G}^{\uparrow}$, $\mathbb{G}^{\nwarrow}$ are needed since $g^{\rightarrow}_{k,l} = g^{\leftarrow}_{-l,-k}$, and similarly for other matrices.

Because neighboring pixels in natural images are strongly dependent, each matrix exhibits a sharp peak around $(k,l) = (0,0)$ and then quickly falls off with increasing k and l. When such matrices are used for steganalysis [92], they are truncated to a small range, such as $-T \leq k,l \leq T$, $T = 4$, to prevent the onset of the "curse of dimensionality." On the other hand, in steganography we can use large-dimensional models ($T = 255$) because it is easier to preserve a model than to learn it.[7] Another reason for using a high-dimensional feature space is to avoid "overtraining" the embedding algorithm to a low-dimensional model as such algorithms may become detectable by a slightly modified feature set, an effect already reported in the DCT domain [76].

By embedding a message, $g^{\rightarrow}_{k,l}(\mathbf{x})$ is modified to $g^{\rightarrow}_{k,l}(\mathbf{y})$. The differences between the features will thus serve as a measure of embedding impact closely tied to the model (the indices i and j run from

[7] Similar reasoning for constructing the distortion function was used in the HUGO algorithm [94].

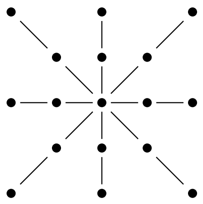

Figure 5.7.1: The union of all 12 cliques consisting of three pixels arranged in a straight line in the 5×5 square neighborhood.

1 to n_1 and $n_2 - 2$, respectively):

$$|g_{k,l}^{\rightarrow}(\mathbf{y}) - g_{k,l}^{\rightarrow}(\mathbf{x})| = \frac{1}{n_1(n_2-2)}\left|\sum_{i,j}[(d_{i,j}^{\rightarrow}, d_{i,j+1}^{\rightarrow})(\mathbf{y}) = (k,l)] - [(d_{i,j}^{\rightarrow}, d_{i,j+1}^{\rightarrow})(\mathbf{x}) = (k,l)]\right| \quad (5.7.13)$$

$$\leq \frac{1}{n_1(n_2-2)}\sum_{i,j}\left|[(d_{i,j}^{\rightarrow}, d_{i,j+1}^{\rightarrow})(\mathbf{y}) = (k,l)] - [(d_{i,j}^{\rightarrow}, d_{i,j+1}^{\rightarrow}(\mathbf{x}) = (k,l)]\right| \quad (5.7.14)$$

$$= \sum_{c \in \mathcal{C}^{\rightarrow}} H_c^{(k,l)\rightarrow}(\mathbf{y}), \quad (5.7.15)$$

where we defined the following locally-supported functions

$$H_c^{(k,l)\rightarrow}(\mathbf{y}) = \frac{1}{n_1(n_2-2)}\left|[(d_{i,j}^{\rightarrow}, d_{i,j+1}^{\rightarrow})(\mathbf{y}) = (k,l)] - [(d_{i,j}^{\rightarrow}, d_{i,j+1}^{\rightarrow})(\mathbf{x}) = (k,l)]\right| \quad (5.7.16)$$

on all horizontal cliques $\mathcal{C}^{\rightarrow} = \{c | c = \{(i,j),(i,j+1),(i,j+2)\}\}$. Notice that the absolute value had to be pulled into the sum to give the potentials a small support. Again, we drop the symbol for the cover image, $\mathbf{x}$, from the argument of $H_c^{(k,l)}$ for the same reason why we do not make the dependence on $\mathbf{x}$ explicit for all other variables, sets, and functions.

Since the other three matrices can be written in this manner as well, we can write the distortion function in the following final form

$$D(\mathbf{y}) = \sum_{c \in \mathcal{C}} V_c(\mathbf{y}), \quad (5.7.17)$$

now with $\mathcal{C} = \mathcal{C}^{\rightarrow} \cup \mathcal{C}^{\nearrow} \cup \mathcal{C}^{\uparrow} \cup \mathcal{C}^{\nwarrow}$, the set of three-pixel cliques along all four directions, and

$$V_c(\mathbf{y}) = \sum_{k,l} w_{k,l} H_c^{(k,l)\rightarrow}(\mathbf{y}), \text{ for each clique } c \in \mathcal{C}^{\rightarrow}, \quad (5.7.18)$$

and similarly for the other three clique types. Notice that we again introduced weights $w_{k,l} > 0$ into the definition of V_c so that we can adjust them according to how sensitive steganalysis is to the individual differences. For example, if we observe that a certain difference pair (k,l) varies significantly over cover images, by assigning it a smaller weight we allow it to be modified more often, while those differences that are stable across covers but sensitive to embedding should be intuitively assigned a larger value so that the embedding does not modify them too much.

To complete the picture, the neighborhood system here is formed by 5×5 neighborhoods and thus the index set can be decomposed into nine disjoint sublattices $\mathcal{S} = \bigcup_{ab} \mathcal{S}_{ab}$, $1 \leq a, b \leq 3$,

$$\mathcal{S}_{ab} = \{(a+3k, b+3l) | 1 \leq a+3k \leq n_1, 1 \leq b+3l \leq n_2\}. \quad (5.7.19)$$

To better explain the effect of embedding changes on the distortion, realize that each pixel belongs to three horizontal, three vertical, three diagonal, and three minor-diagonal cliques. When a single pixel $x_{i,j}$ is changed, it affects only the 12 potentials whose clique contains $x_{i,j}$. Let us say that the original pixel values $c_0 = \{x_{i,j}, x_{i,j+1}, x_{i,j+2}\}$ had differences k, l, and the pixel value changed from $x_{i,j}$ to $y_{i,j} = x_{i,j} + 1$. Then, the pixel differences will be modified to $k-1, l$. Considering just the contribution from $H_{c_0}^{(k,l)\rightarrow}$ to the potential V_{c_0} (5.7.18), it will increase by the sum of $w_{k,l}$ (the pair k, l is leaving cover) and $w_{k-1,l}$ (a new pair appears in the stego image).

5.7.3 Other Options

The framework presented in this chapter allows the sender to formulate the local potentials directly instead of obtaining them as the bounding distortion. For example, the cliques and their potentials may be determined by the local image content or by learning the cover source using the method of fields of experts [100]. The merit of these possibilities can be evaluated by steganalyzers trained on a large set of images. The important question of optimizing the local potential functions w.r.t. statistical detectability is an important direction we will study in Chapter 7.

5.8 Experiments

In this section, we validate the proposed framework experimentally and include a comparison between simple steganographic algorithms, such as binary and ternary ±1 embedding and steganography implemented via the bounding distortion and the additive approximation (5.7.2). We do so for the payload-limited sender in Section 5.8.2 as well as the distortion-limited sender (Section 5.8.3). Following the separation principle, we study the security of all embedding algorithms by comparing their performance when simulated at their corresponding rate–distortion bounds. Methods allowing to implement the proposed algorithms in practice are described in Chapter 6. For the case of the bounding distortion, the capacity loss w.r.t. the optimal payload given by $H(\pi_\lambda)$ is evaluated by means of the thermodynamic integration algorithm from Section 5.5.2.

5.8.1 Tested Embedding Methods

For the methods based on additive approximation and the bounding distortion, we used as a feature vector the joint probability matrix $\mathbb{G}^{\rightarrow}(\mathbf{x}) = (g^{\rightarrow}_{k,l,m}(\mathbf{x}))$ defined similarly as in (5.7.11) with the difference vector computed from *four* consecutive pixels $(d^{\rightarrow}_{i,j}, d^{\rightarrow}_{i,j+1}, d^{\rightarrow}_{i,j+2}) = (k,l,m)$. As above, four such matrices corresponding to four spatial directions were computed. The matrices were used at their full size $T = 255$ leading to model dimensionality $d = 4 \times 511^3 \approx 5 \cdot 10^8$.

The weights were chosen to be small for those triples $(d^{\rightarrow}_{i,j}, d^{\rightarrow}_{i,j+1}, d^{\rightarrow}_{i,j+2}) = (k,l,m)$ that occur infrequently in images and large for frequented triples. Following the recommendation described in [94], since the frequency of occurrence of the triples falls off quickly with their norm, we choose the weights as

$$w_{k,l,m} = \left(\sigma + \sqrt{k^2 + l^2 + m^2}\right)^{-\theta}, \tag{5.8.1}$$

with $\theta = 1$ and $\sigma = 1$. The purpose of the weights is to force the embedding algorithm to modify those parts of the model that are difficult to model accurately, forcing thus the steganalyst to use a more accurate model. Here, the advantage goes to the steganographer, because preserving a high-dimensional feature vector is more feasible than accurately modeling it.

Because the neighborhood $\eta(i)$ in this case contains 7×7 pixels, the image was divided into 16 square sublattices on which embedding was carried out independently. We tested binary embedding, $\mathcal{I}_i = \{x_i, x'_i\}$, where x'_i was selected randomly and uniformly from $\{x_i - 1, x_i + 1\}$ and then fixed for all experiments with cover $\mathbf{x}$. The payload-limited sender was simulated using the Gibbs sampler constrained to only two sweeps. Increasing the number of sweeps did not lead to further improvement. The curiously low number of sweeps sufficient to properly implement the Gibbs sampler is most likely due to the fact that the dependencies dictated by the bounding distortion are rather weak.

We implemented this framework with three different ranges of stego pixels: binary flipping patterns, $\mathcal{I}_i = \{x_i, y_i\}$, where y_i was selected randomly and uniformly from $\{x_i - 1, x_i + 1\}$ and then fixed for all experiments with cover $\mathbf{x}$, ternary patterns, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$, and pentary patterns, $\mathcal{I}_i = \{x_i - 2, \ldots, x_i + 2\}$. For all three cases, we simulated the method based on the bounding distortion (5.7.10) and the additive approximation (5.7.2) on the $d = 4 \times 511^2$-dimensional feature space of joint probability matrices $\mathbb{G}^{\rightarrow}$, $\mathbb{G}^{\nearrow}$, $\mathbb{G}^{\uparrow}$, and $\mathbb{G}^{\nwarrow}$.

For comparison, we contrasted the performance against two standard embedding methods: binary ±1 embedding constrained to the same sets $\mathcal{I}_i$ as the Gibbs sampler and ternary ±1 embedding with

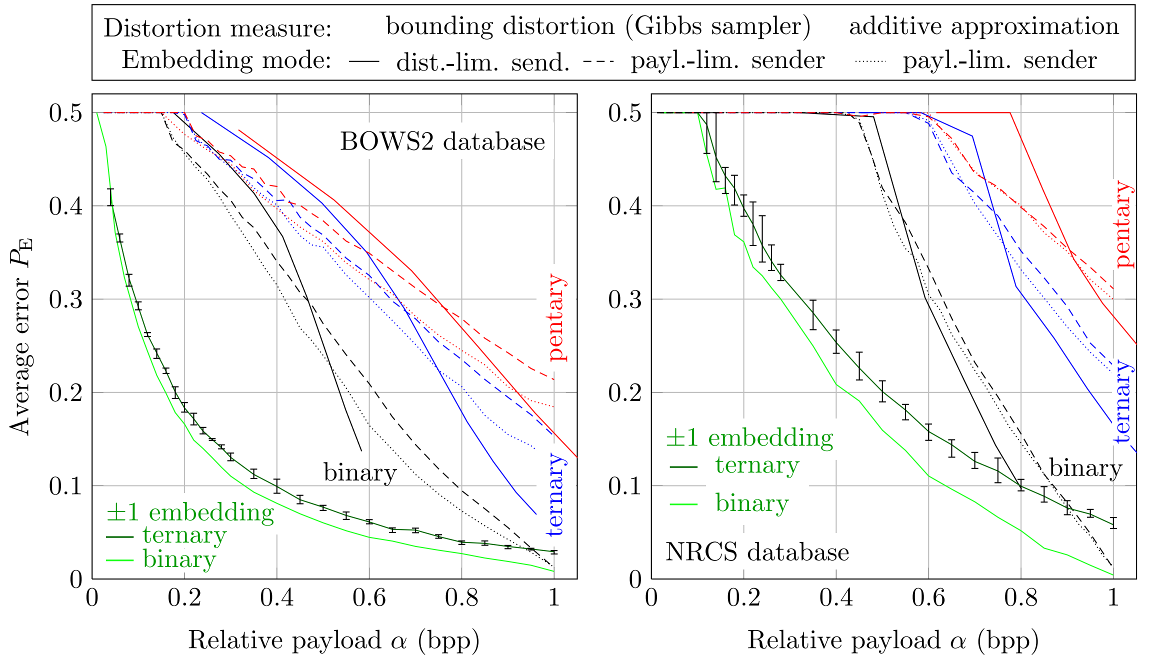

Figure 5.8.1: Comparison of ±1 embedding with optimal binary and ternary coding with binary embedding algorithms based on the Gibbs construction with a bounding distortion and the additive approximation as described in Section 5.8.1. The error bars depict the minimum and maximum steganalyzer error P_E (**??**) over five runs of SVM classifiers with different division of images into training and testing set.

$\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\}$. Both schemes are special cases of our framework with $D(\mathbf{y}) = \sum_i [x_i \neq y_i]$. We repeat that all schemes were simulated on their corresponding bounds.

All algorithms were tested on two image sources with different noise characteristics: the BOWS2 database [5] containing approximately 10800 grayscale images with a fixed size of 512 × 512 pixels coming from rescaled and cropped natural images of various sizes, and the NRCS database[8] with 3322 color scans of analogue photographs mostly of size 2100 × 1500 pixels converted to grayscale. For algorithms based on the Gibbs construction, simulating the optimal noise in C++ took less than 5 seconds for BOWS2 images and 60 seconds for the larger images from the NRCS database (for both the payload and distortion-limited sender).

Steganalysis was carried out using the second-order SPAM feature set with $T = 3$ using the methodology described in Section 2.3. We compare the schemes using the minimum average classification error P_E.

5.8.2 Payload-limited sender

Figure 5.8.1 displays the comparison of all tested embedding methods. For the BOWS2 database, the methods based on the additive approximation and the bounding distortion are completely undetectable for payloads smaller than 0.15 bpp (bits per pixel), which suggests that the embedding changes are made in pixels not covered by the SPAM features. This number increases to at least 0.45 bpp for the NRCS database which is expected because its images are more noisy. For such payloads, the detector makes random guesses and, thus, due to the large number of testing samples, its error becomes exactly $P_E = 0.5$. With the relative payload α approaching 1, binary embedding schemes degenerate to binary ±1 embedding and thus become equally detectable. The same holds for ternary schemes. Both schemes allow communicating more than ten times larger payloads with

[8] http://photogallery.nrcs.usda.gov/

$P_E = 40\%$, when compared to ternary ±1 embedding (on the BOWS2 database), and roughly four times larger payloads for the NRCS database. The results also suggest that secure payload can be further increased by allowing embedding changes of larger amplitude (up to ±2). Of course, this benefit is closely tied to the design of D because larger changes are easily detectable when not made adaptively [115].

The advantage of using the Gibbs sampler for embedding is more apparent for larger payloads, when the embedding changes start to interact (the BOWS2 database only). We believe this is due to strong inter-pixel dependencies caused by resizing the original image.

5.8.3 Distortion-limited sender

In this chapter, we worked out the proposed methodology for both the payload-limited sender and the distortion-limited sender. The former embeds a fixed payload in every image with minimal distortion, while the latter embeds the maximal payload for a given distortion in every image.[9] The distortion-limited sender better corresponds to our intuition that, for a fixed statistical detectability, more textured or noisy images can carry a larger secure payload than smoother or simpler images. The fact that the size of the hidden message is driven by the cover image essentially represents a more realistic case of the batch steganography paradigm [65].

Since the payload is now determined by image content, it varies over the database. In this setup, we trained the steganalyzer on stego images embedded with a fixed distortion constraint D_ϵ. To be able to display the results in Figure 5.8.1, we reparametrized P_E to be a function of the relative payload α, which we obtain for each D_ϵ by averaging α over all images from the database. The solid lines represent the results obtained from the Gibbs sampler (Algorithm 5.3 with three sweeps) with $D(\mathbf{y})$ defined as the bounding distortion. As long as the distortion adequately measures statistical detectability, the distortion-limited sender should be more secure than the payload-limited sender. Figure 5.8.1 confirms this up to a certain payload where the performance is swapped. This means that either our distortion function is suboptimal or the steganalyzer does not properly measure statistical detectability.

Because the images in both databases are all of the same size, a fixed value of D_ϵ was used for all images. When dealing with images of varying size, we should set $D_\epsilon = d_\epsilon\sqrt{n}$, at least for stegosystems falling under the Square-root law (see Chapter 3).

As a final remark, we would like to point out that even though the improvement brought by the Gibbs construction over the additive approximation is not very large (and negligible for the NRCS database) it will likely increase in the future as practical steganalysis manages to better exploit inter-pixel dependencies. This is because mutually independent embedding cannot properly preserve dependencies or model interactions among embedding changes. For example, steganography in digital-camera color images will likely benefit from the Gibbs construction due to strong dependencies among color planes caused by color interpolation and other in-camera processing.

5.8.4 Analysis of Upper Bounds

As described in Section 5.6.3, Algorithm 5.2 for the payload-limited sender is unable to embed the optimal payload of $H(\pi_\lambda)$ for three reasons. The performance may be affected by the small number of sweeps of the Gibbs sampler, the parameter λ may vary slightly among the sublattices, and the algorithm embeds the erasure entropy $H^-(\pi_\lambda) \leq H(\pi_\lambda)$. The combined effect of these factors is of great importance for practitioners and is evaluated below for two images using the Gibbs sampler and the thermodynamic integration as explained in Section 5.5.2.

Since the Gibbs construction depends on the cover image $\mathbf{x}$, we present the results for two grayscale images of size 512×512 pixels coming from two different sources. The test image "Lenna" was obtained from http://en.wikipedia.org/wiki/File:Lenna.png and converted to grayscale using GNU Image Manipulation Program (GIMP) and "0.png" is from the BOWS2 database. In both cases, we used the same sets $\mathcal{I}_i$ and the same feature set as in the previous section with the bounding distortion with weight parameters $\sigma = 1$ and $\theta = 1$.

[9] For schemes with uniform embedding cost, these two cases coincide.

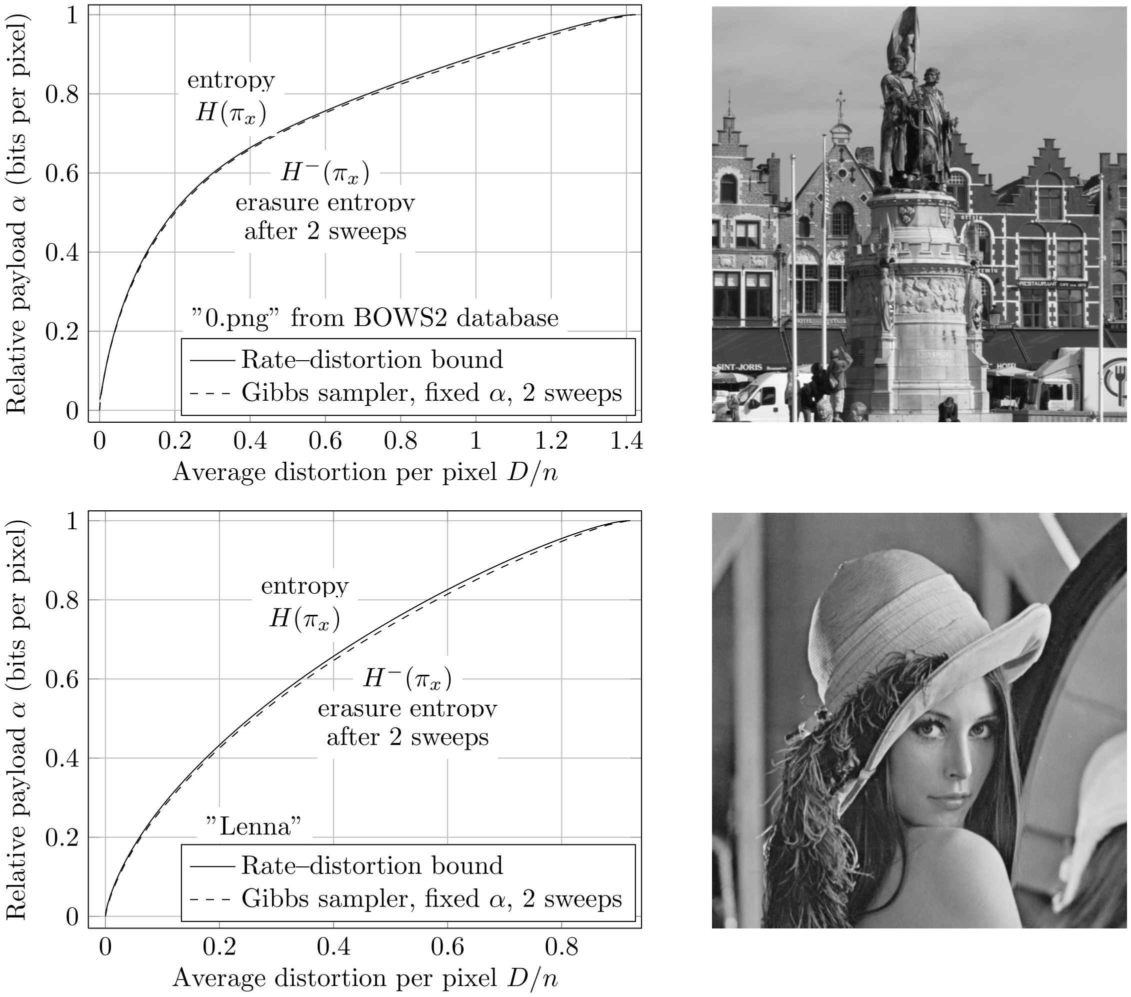

Figure 5.8.2: Comparison of the payload loss of Algorithm 5.2 for cover images “0.png” and “Lenna” shown on the right. The rate–distortion bounds were obtained using the Gibbs sampler (5.5.15) and the thermodynamic integration (5.5.17).

The image “0.png” contains more areas with edges and textures than “Lenna” and thus for small distortions, it offers a larger capacity than “Lenna” because the weights (5.8.1) around edges and complex texture are small. This is apparent from the slopes of the rate–distortion bounds in Figure 5.8.2.

The same figure compares the rate–distortion performance of the payload-limited sender simulated by the Gibbs sampler with only two sweeps as described in Algorithm 5.2. For a given payload, the distortion was obtained as an average over 100 random messages. The comparison shows that the payload loss of Algorithm 5.2 to the optimal $H(\pi_\lambda)$ is quite small. Note that the erasure entropy, $H^{-}(\pi_\lambda)$, plotted in the figure has been computed over the sublattices after two sweeps and thus already contains the impact of all three factors discussed at the beginning of this section.

5.9 Summary and Conclusion

Currently, the most successful principle for designing practical steganographic systems that embed in empirical covers is based on minimizing a suitably defined distortion measure. Implementation difficulties and a lack of practical embedding methods have so far limited the application of this principle to a rather special class of distortion measures that are additive over pixels. With the development of near-optimal low-complexity coding schemes, such as the syndrome-trellis codes (Chapter 6), this direction has essentially reached its limits. It is our firm belief that further substantial increase in secure payload is possible only when the sender uses adaptive schemes that place embedding changes based on the local content, that dare to modify pixels in some regions by more than 1, and that consider interactions among embedding changes while preserving higher-order statistics among pixels. This chapter is an important step in this direction.

We offer the steganographer a complete methodology for embedding while minimizing an arbitrarily defined distortion measure D. The absence of any restrictions on D means that the remaining task left to the sender is to find a distortion measure that correlates with statistical detectability. An appealing possibility is to define D as a weighted norm of the difference between cover and stego feature vectors used in steganalysis. This immediately connects the principle of minimum-distortion steganography with the concept of model preservation which has so far been limited to low-dimensional models. Being able to preserve a large-dimensional model gives the steganographer a great advantage over the steganalyst because of the difficulties associated with learning a high-dimensional cover source model using statistical learning tools.

The proposed framework is called the Gibbs construction and it connects steganography with statistical physics, which contributed with many practical algorithms. In particular, the Gibbs sampler combined with the thermodynamic integration can be used to derive the rate–distortion bound, simulate the impact of optimal embedding, and realize near-optimal embedding algorithms. These three tasks can be addressed separately (the so-called "separation principle") giving the sender a great amount of design flexibility as well as control over losses of practical schemes.

An important case elaborated in this chapter corresponds to D defined as a sum of local potentials over small pixel neighborhoods. Here, the optimal distribution of embedding modifications reduces to a Markov random field and the Gibbs sampler can be turned into a practical embedding algorithm able to consider dependencies among embedding changes. When D cannot be written as a sum of local potentials, practical (suboptimal) methods can be realized by approximating D either with an additive distortion measure or with local potentials. The problem of finding the best approximation for a given non-local D is of its own interest. We did not cover the task of minimizing the statistical detectability with respect to the distortion function at this point. This problem is postponed to Chapter 7.

We described the proposed methodology both for a payload-limited sender and the distortion-limited sender. The former embeds a fixed payload in every image with minimal distortion, while the latter embeds the maximal payload for a given distortion in every image. The distortion-limited sender better corresponds to our intuition that, for a fixed statistical detectability, more textured or noisy images can carry a larger secure payload than smoother or simpler images. The fact that the size of the hidden message is driven by the cover image essentially represents a more realistic case of the batch steganography paradigm [65].

Finally, the proposed methodology can be applied to other data hiding problems where the statistical detectability constraint could be replaced by a perceptual distortion constraint.

The source code used for all experiments in this chapter can be found at
http://dde.binghamton.edu/download/gibbs.

Chapter 6

Minimizing Additive Distortion Function in Steganography

The vast majority of steganographic schemes known today can be interpreted as if they minimize suitably defined distortion function between cover and stego images. The Gibbs construction described in Chapter 5 provides a solid theoretical background for such stegosystems. When the distortion function is additive over pixel cliques, the Gibbs sampler allows us to simulate the statistical impact of the optimal embedding algorithm under the payload or the distortion constraint. In fact, the Gibbs sampler decomposes the original embedding problem to a series of subproblems (called sweeps) where only pixel-additive (not clique-additive) distortion functions need to be minimized. In this chapter, we describe practical methods for minimizing pixel-additive distortion functions, which allows us to implement the Gibbs sampler (and many other algorithms) in practice. In this chapter, by "additive distortion" we mean the pixel-additive distortion in the sense of Section 5.4.

This chapter is organized as follows. In the first section, we formulate a simpler version of the embedding problem and quantities for evaluating the performance of practical algorithms with respect to each other and the known performance bounds. The syndrome coding method for steganographic communication is reviewed in Section 6.2. By pointing out the limitations of previous approaches, we motivate our contribution, which starts in Section 6.3, where we introduce a class of syndrome-trellis codes for binary embedding operations. We describe the construction and optimization of the codes and provide extensive experimental results on different distortion profiles including the wet paper channel. In Section 6.4, we show how to decompose the problem of embedding using non-binary embedding operations to a series of binary problems using a multi-layered approach so that practical algorithms can be realized using binary STCs. The application and merit of the proposed coding construction is demonstrated experimentally in Section 6.5 on covers formed by digital images in raster and transform (JPEG) domains. Both the binary and non-binary versions of payload- and distortion-limited senders are tested by blind steganalysis. Finally, the chapter is concluded in Section 6.6.

6.1 Problem Formulation

In this chapter, we follow the same notation and terminology as in Chapter 5 and focus on a special case when the distortion function D is additive over individual cover pixels

$$D(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{n} \rho_i(\mathbf{x}, y_i), \tag{6.1.1}$$

where $\rho_i : \mathcal{X} \times \mathcal{I}_i \to [-K, K]$, $0 < K < \infty$, are bounded functions expressing the cost of replacing the cover pixel x_i with y_i. Note that ρ_i may arbitrarily depend on the entire cover image $\mathbf{x}$, allowing thus the sender to place the embedding changes adaptively w.r.t. the image content. The fact that the value of $\rho_i(\mathbf{x}, y_i)$ is independent of changes made at other pixels implies that the embedding changes do not interact. The boundedness of $D(\mathbf{x}, \mathbf{y})$ is not limiting the sender in practice since the case when a particular value y_i is forbidden (a requirement often found in practical steganographic schemes [47]) can be resolved by excluding y_i from $\mathcal{I}_i$. In practice, the sets $\mathcal{I}_i$, $i \in \{1, \dots, n\}$, may depend on cover pixels and thus may not be available to the receiver. To handle this case, we expand the domain of ρ_i to $\mathcal{X} \times \mathcal{I}$ and define $\rho_i(\mathbf{x}, y_i) = \infty$ whenever $y_i \notin \mathcal{I}_i$.

We intentionally keep the definition of the distortion function rather general. In particular, we do *not* require $\rho_i(\mathbf{x}, x_i) \leq \rho_i(\mathbf{x}, y_i)$ for all $y_i \in \mathcal{I}_i$ to allow for the case when it is actually beneficial to make an embedding change instead of leaving the pixel unchanged – case which happens in the Gibbs construction.

We assume the sender obtains her payload in the form of a pseudo-random bit stream, such as by compressing or encrypting the original message. We further assume that the embedding algorithm associates every cover image $\mathbf{x}$ with a pair $\{\mathcal{Y}, \pi\}$, where $\mathcal{Y}$ is the set of all stego images into which $\mathbf{x}$ can be modified and π is their probability distribution characterizing the sender's actions, $\pi(\mathbf{y}) \triangleq P(\mathbf{Y} = \mathbf{y}|\mathbf{x})$. As in Chapter 5, we think of $\mathbf{x}$ as a constant parameter that is *fixed in the very beginning* and thus we do not further denote the dependency on it explicitly. For this reason, we simply write $D(\mathbf{y}) \triangleq D(\mathbf{x}, \mathbf{y})$.

If the receiver knew $\mathbf{x}$, the sender could send up to $H(\pi)$ bits on average while introducing the average distortion $E_\pi[D]$ by choosing the stego image according to π. By the Gel'fand–Pinsker theorem [50], the knowledge of $\mathbf{x}$ does not give any fundamental advantage to the receiver and the same performance can be achieved as long as $\mathbf{x}$ is known to the sender. Indeed, none of the practical embedding algorithms introduced in this chapter requires the knowledge of $\mathbf{x}$ or D for reading the message. We assume the same distortion- and payload-limited versions of the embedding problem as defined in Section 5.2 which we further denote as DLS and PLS problems, respectively.

6.1.1 Performance Bounds and Comparison Metrics

Both embedding problems described in Section 5.2 bear relationship to the problem of source coding with a fidelity criterion as described by Shannon [108] and the problem of source coding with side information available at the transmitter, the so-called Gel'fand-Pinsker problem [50]. PLS and DLS optimization problems are dual to each other, meaning that the optimal distribution for (5.2.4) and (5.2.5) is, for some value of D_ϵ, also optimal for (5.2.6) and (5.2.7). Following the results derived in Chapter 5, the optimal solution has the form of a Gibbs distribution:

$$\pi(\mathbf{y}) = \frac{\exp(-\lambda D(\mathbf{y}))}{Z(\lambda)} \overset{(a)}{=} \prod_{i=1}^{n} \frac{\exp(-\lambda \rho_i(y_i))}{Z_i(\lambda)} \triangleq \prod_{i=1}^{n} \pi_i(y_i), \tag{6.1.2}$$

where the parameter $\lambda \in [0, \infty)$ is obtained from the corresponding constraints (5.2.5) or (5.2.7) by solving an algebraic equation;[1] $Z(\lambda) = \sum_{\mathbf{y} \in \mathcal{Y}} \exp(-\lambda D(\mathbf{y}))$, $Z_i(\lambda) = \sum_{y_i \in \mathcal{I}_i} \exp(-\lambda \rho_i(y_i))$ are the corresponding partition functions. Step (a) follows from the additivity of D, which also leads to mutual independence of individual stego pixels y_i given $\mathbf{x}$.

By changing each pixel i with probability π_i (6.1.2) one can *simulate* embedding with optimal π. In Section 6.5, we use the simulators to benchmark different coding algorithms we develop in this chapter by comparing the security of practical schemes using blind steganalysis.

An established way of evaluating coding algorithms in steganography is to compare the *embedding efficiency* $e(\alpha) = \alpha n / E_\pi[D]$ (in bits per unit distortion) for a fixed expected relative payload $\alpha = m/n$ with the upper bound derived from (6.1.2). When the number of changes is minimized, e is the average number of bits hidden per embedding change. For general functions ρ_i, the interpretation of this metric becomes less clear. A different and more easily interpretable metric is to compare the payload, m, of an embedding algorithm w.r.t. the payload, m_{MAX}, of the optimal DLS for a fixed D_ϵ,

$$l(D_\epsilon) = \frac{m_{\mathrm{MAX}} - m}{m_{\mathrm{MAX}}}, \tag{6.1.3}$$

which we call the *coding loss*.

[1] A simple binary search will do the job because both $H(\pi)$ and $E_\pi[D]$ are monotone w.r.t. λ.

6.1.2 Binary Embedding Operation

In this section, we show that for binary embedding operations, it is enough to consider a slightly narrower class of distortion functions without experiencing any loss of generality. The binary case is very important as the embedding method introduced in this chapter is first developed for this special case and then extended to non-binary operations.

For binary embedding with $\mathcal{I}_i = \{x_i, \overline{x}_i\}$, $x_i \neq \overline{x}_i$, we define $\rho_i^{\min} = \min\{\rho_i(\mathbf{x}, x_i), \rho_i(\mathbf{x}, \overline{x}_i)\}$, $\varrho_i = |\rho_i(\mathbf{x}, x_i) - \rho_i(\mathbf{x}, \overline{x}_i)| \geq 0$, and rewrite (6.1.1) as:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \rho_i^{\min} + \sum_{i=1}^{n} \varrho_i \cdot [\rho_i^{\min} < \rho_i(\mathbf{x}, y_i)]. \tag{6.1.4}$$

Because the first sum does not depend on $\mathbf{y}$, when minimizing D over $\mathbf{y}$ it is enough to consider only the second term. It now becomes clear that embedding in cover $\mathbf{x}$ while minimizing (6.1.4) is equivalent to embedding in cover $\mathbf{z}$

$$z_i = \begin{cases} x_i & \text{when } \rho_i^{\min} = \rho_i(\mathbf{x}, x_i) \\ \overline{x}_i & \text{when } \rho_i^{\min} = \rho_i(\mathbf{x}, \overline{x}_i). \end{cases} \tag{6.1.5}$$

while minimizing

$$\tilde{D}(\mathbf{z}, \mathbf{y}) = \sum_{i=1}^{n} \tilde{\rho}_i(\mathbf{z}, y_i) \triangleq \sum_{i=1}^{n} \varrho_i \cdot [y_i \neq z_i], \tag{6.1.6}$$

with non-negative costs $\tilde{\rho}_i(\mathbf{z}, z_i) = 0 \leq \tilde{\rho}_i(\mathbf{z}, \overline{z}_i) = \varrho_i$ for all i (when the cover pixel z_i is changed to $\bar{z}_i$, the distortion $\tilde{D}$ always increases). Thus, from now on for binary embedding operations, we will always consider distortion functions of the form:

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \varrho_i \cdot [y_i \neq x_i], \tag{6.1.7}$$

with $\varrho_i \geq 0$.

For example, F5 [125] uses the distortion function (6.1.7) with $\varrho_i = 1$ (the number of embedding changes), while nsF5 [47] employs wet paper codes, where $\varrho_i \in \{1, \infty\}$. In some embedding algorithms [43, 74, 102], where the cover is preprocessed and quantized before embedding, ϱ_i is proportional to the quantization error at pixel x_i.

Additionally, for binary embedding operations we speak of a *distortion profile* ϱ if $\varrho_i = \varrho(i/n)$ for all i, where ϱ is a non-decreasing[2] function $\varrho : [0, 1] \to [0, K]$. The following distortion profiles are of interest in steganography (this is not an exhaustive list): the *constant profile*, $\varrho(x) = 1$, when all pixels have the same impact on detectability when changed; the *linear profile*, $\varrho(x) = 2x$, when the distortion is related to a quantization error uniformly distributed on $[-Q/2, Q/2]$ for some quantization step $Q > 0$; and the *square profile*, $\varrho(x) = 3x^2$, which can be encountered when the distortion is related to a quantization error that is not uniformly distributed.

In this chapter, we normalize the profile ϱ so that $E_\pi[D]/n = \sum_{i=1}^{n} \pi_i \varrho_i / n = 0.5$ when embedding a full payload $m = n$. With this convention, Figure 6.1.1 displays the lower bounds on the average per-pixel distortion for three distortion profiles.

In practice, some cover pixels may require $\mathcal{I}_i = \{x_i\}$ and thus $\varrho_i = \infty$ (the so-called *wet pixels* [43, 45, 47]) to prevent the embedding algorithm from modifying them. Since such pixels are essentially constant, in this case we measure the relative payload α with respect to the set of *dry pixels* $\{x_i | \varrho_i < \infty\}$, i.e., $\alpha = m / |\{x_i | \varrho_i < \infty\}|$. The overall channel is called the wet paper channel and it is characterized by the profile ϱ of dry pixels and *relative wetness* $\tau = |\{x_i | \varrho_i = \infty\}|/n$. The wet paper channel is often required when working with images in the JPEG domain [47].

[2] By reindexing the pixels, we can indeed assume that $\varrho_1 \leq \varrho_2 \leq \cdots \leq \varrho_n \leq K$.

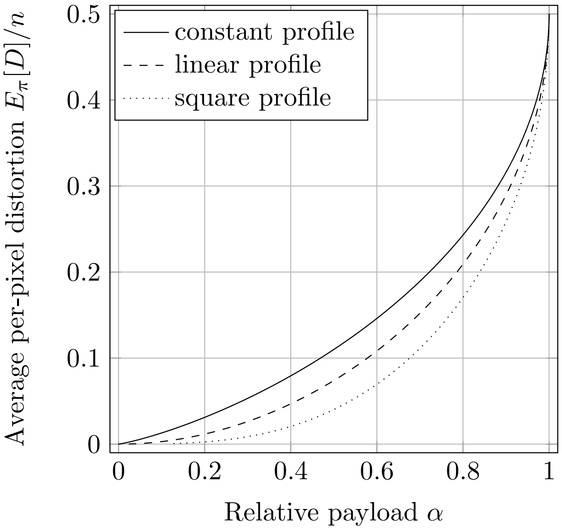

Figure 6.1.1: Lower bound on the average per-pixel distortion, $E_\pi[D]/n$, as a function of relative payload α for different distortion profiles.

6.2 Syndrome Coding

The PLS and the DLS can be realized in practice using a general methodology called *syndrome coding*. In this section, we briefly review this approach and its history paving our way to Section 6.3 and 6.4, where we explain the main contribution of this chapter – the syndrome-trellis codes.

Let us first assume a binary version of both embedding problems. Let $\mathcal{P} : \mathcal{I}_i \rightarrow \{0, 1\}$ be a parity function shared between the sender and the receiver satisfying $\mathcal{P}(x_i) \neq \mathcal{P}(y_i)$ such as $\mathcal{P}(x) = x \bmod 2$. The sender and the receiver need to implement the embedding and extraction mappings defined as $\mathrm{Emb}: \mathcal{X} \times \{0, 1\}^m \rightarrow \mathcal{Y}$ and $\mathrm{Ext}: \mathcal{Y} \rightarrow \{0, 1\}^m$ satisfying

$$\mathrm{Ext}(\mathrm{Emb}(\mathbf{x}, \mathbf{m})) = \mathbf{m} \quad \forall \mathbf{x} \in \mathcal{X}, \forall \mathbf{m} \in \{0, 1\}^m,$$

respectively. In particular, we do not assume the knowledge of the distortion function D at the receiver and thus the embedding scheme can be seen as being universal in this sense. A common information-theoretic strategy for solving the PLS problem is known as binning [86], which we implement using cosets of a linear code. Such a construction, better known as syndrome coding, is capacity achieving for the PLS problem if random linear codes are used.

In syndrome coding, the embedding and extraction mappings are realized using a binary linear code $\mathcal{C}$ of length n and dimension $n - m$:

$$\mathrm{Emb}(\mathbf{x}, \mathbf{m}) = \arg \min_{\mathcal{P}(\mathbf{y}) \in \mathcal{C}(\mathbf{m})} D(\mathbf{x}, \mathbf{y}), \tag{6.2.1}$$

$$\mathrm{Ext}(\mathbf{y}) = \mathbb{H}\mathcal{P}(\mathbf{y})^T, \tag{6.2.2}$$

where $\mathcal{P}(\mathbf{y}) = (\mathcal{P}(y_1), \ldots, \mathcal{P}(y_n))$, $\mathbb{H} \in \{0, 1\}^{m \times n}$ is a parity-check matrix of the code $\mathcal{C}$, $\mathcal{C}(\mathbf{m}) = \{\mathbf{z} \in \{0, 1\}^n | \mathbb{H}\mathbf{z}^T = \mathbf{m}\}$ is the coset corresponding to syndrome $\mathbf{m}$, and all operations are in binary arithmetic.

Unfortunately, random linear codes are not practical due to the exponential complexity of the optimal binary coset quantizer (6.2.1), which is the most challenging part of the problem. In this work, we describe a rich class of codes for which the quantizer can be solved optimally with linear time and space complexity w.r.t. n.

Since the DLS is a dual problem to the PLS, it can be solved by (6.2.1) and (6.2.2) once an appropriate message size m is known. This can be obtained in practice by $m = m_{\mathrm{MAX}}(1 - l')$, where $m_{\mathrm{MAX}} = H(\pi_\lambda)$ is the maximal average payload obtained from the optimal distribution

(6.1.2) achieving average distortion D_ϵ and l' is an experimentally-obtained coding loss we expect the algorithm will achieve.

One possible approach for solving a non-binary version of both embedding problems is to increase the size of the alphabet and use (6.2.1) and (6.2.2) with a non-binary code $\mathcal{C}$, such as the ternary Hamming code. A more practical alternative with lower complexity is the multi-layered construction proposed in Section 6.4, which decomposes (6.2.1) and (6.2.2) into a series of binary embedding subproblems. Such decomposition leads to the optimal solution of PLS and DLS as long as each binary subproblem is solved optimally. For this reason, in Section 6.3 we focus on the binary PLS problem for a large variety of relative payloads and different distortion profiles including the wet paper channel.

6.2.1 Prior Art

The problem of minimizing the embedding impact in steganography, introduced above as the PLS problem, has been already conceptually described by Crandall [19] in his essay posted on the steganography mailing list in 1998. He suggested that whenever the encoder embeds at most one bit per pixel, it should make use of the embedding impact defined for every pixel and minimize its total sum:

> "Conceptually, the encoder examines an area of the image and weights each of the options that allow it to embed the desired bits in that area. It scores each option for how conspicuous it is and chooses the option with the best score."

Later, Bierbrauer [6, 7] studied a special case of this problem and described a connection between codes (not necessarily linear) and the problem of minimizing the number of changed pixels (the constant profile). This connection, which has become known as matrix embedding (encoding), was made famous among steganographers by Westfeld [125] who incorporated it in his F5 algorithm. A binary Hamming code was used to implement the syndrome-coding scheme for the constant profile. Later on, different authors suggested other linear codes, such as Golay [117], BCH [106], random codes of small dimension [48], and non-linear codes based on the idea of a blockwise direct sum [7]. Current state-of-the-art methods use codes based on Low Density Generator Matrices (LDGMs) [38] in combination with the ZZW construction [129]. The embedding efficiency of these codes stays rather close to the bound for arbitrarily small relative payloads [36].

The versatile syndrome-coding approach can also be used to communicate via the wet paper channel using the so-called wet paper codes [43]. Wet paper codes minimizing the number of changed dry pixels were described in [44, 106, 131, 26].

Even though other distortion profiles, such as the linear profile, are of great interest to steganography, no general solution with performance close to the bound is currently known. The authors of [74] approached the PLS problem by minimizing the distortion on a block-by-block basis utilizing a Hamming code and a suboptimal quantizer implemented using a brute-force search that allows up to three embedding changes. Such an approach, however, provides highly suboptimal performance far from the theoretical bound (see Figure 6.3.7). A similar approach based on BCH codes and a brute-force quantizer was described in [102] achieving a slightly better performance than Hamming codes. Neither Hamming or BCH codes can be used to deal with the wet paper channel without significant performance loss. To the best of our knowledge, no solution is known that could be used to solve the PLS problem with arbitrary distortion profile containing wet pixels.

One promising direction towards replacing the random linear codes while keeping the optimality of the construction has recently been proposed by Arikan [2], who introduced the so-called polar codes for the channel coding problem. One advantage is that the complexity of encoding and decoding algorithms for polar codes is $n \log n$. Moreover, most of the capacity-achieving properties of random linear codes are retained even for other information-theoretic problems and thus polar codes are known to be optimal for the PLS problem [80] (at least for the uniform profile). Unfortunately, to apply such codes, the number of pixels, n, must be very high, which may not be always satisfied in practice. We believe that the proposed syndrome-trellis codes offer better trade-offs when used in practical embedding schemes.

6.3 Syndrome-Trellis Codes

In this section, we focus on solving the binary PLS problem with distortion function (6.1.6) and propose a large class of linear codes which we call the syndrome-trellis codes. These codes will serve as a building block for non-binary PLS and DLS problems in Section 6.4.

The construction behind STCs is not new from an information-theoretic perspective, since the STCs are trellis codes[3] represented in a dual domain. However, STCs are very interesting for practical steganography since they allow solving both embedding problems with a very small coding loss over a wide range of distortion profiles even with wet pixels. The same code can be used with all profiles making the embedding algorithm practically universal. STCs offer general and state-of-the-art solution for both embedding problems in steganography. Here, we give the description of the codes along with their graphical representation, the syndrome trellis. Such construction is prepared for the Viterbi algorithm, which is optimal for solving (6.2.1). Important practical guidelines for optimizing the codes and using them for the wet paper channel are also covered. Finally, we study the performance of these codes by extensive numerical simulations using different distortion profiles including the wet paper channel.

Syndrome-trellis codes targeted to applications in steganography were described in [30], which was written for practitioners. In this chapter, we expect the reader to have a working knowledge of convolutional codes which are often used in data-hiding applications such as digital watermarking. Convolutional codes are otherwise described in Chapters 25 and 48 in [83]. For a complete example of the Viterbi algorithm used in the context of STCs, we refer the reader to [30].

Our main goal is to develop efficient syndrome-coding schemes for an *arbitrary* relative payload α with the main focus on small relative payloads (think of $\alpha \leq 1/2$ for example). In steganography, the relative payload must decrease with increasing size of the cover object in order to maintain the same level of security, which is a consequence of the square root law [32]. Moreover, recent results from steganalysis in both spatial [92] and DCT domains [77] suggest that the secure payload for digital image steganography is always far below 1/2. Another reason for targeting smaller payloads is the fact that as $\alpha \to 1$, all binary embedding algorithms tend to introduce changes with probability 1/2, no matter how optimal they are. Denoting with $R = (n - m)/n$ the rate of the linear code $\mathcal{C}$, then $\alpha \to 0$ translates to $R = 1 - \alpha \to 1$, which is characteristic for applications of syndrome coding in steganography.

6.3.1 From Convolutional Codes to Syndrome-Trellis Codes

Since Shannon [108] introduced the problem of source coding with a fidelity criterion in 1959, convolutional codes were probably the first "practical" codes used for this problem [120]. This is because the gap between the bound on the expected per-pixel distortion and the distortion obtained using the optimal encoding algorithm (the Viterbi algorithm) decreases exponentially with the constraint length of the code [120, 58]. The complexity of the Viterbi algorithm is linear in the block length of the code, but exponential in its constraint length (the number of trellis states grows exponentially in the constraint length).

When adapted to the PLS problem, convolutional codes can be used for syndrome coding since the best stego image in (6.2.1) can be found using the Viterbi algorithm. This makes convolutional codes (of small constraint length) suitable for our application because the entire cover object can be used and the speed can be traded for performance by adjusting the constraint length. Note that the receiver does not need to know D since only the Viterbi algorithm requires this knowledge. By increasing the constraint length, we can achieve the average per-pixel distortion that is arbitrarily close to the bounds and thus make the coding loss (6.1.3) approach zero. Convolutional codes are often represented with shift-registers (see Chapter 48 in [83]) that generate the codeword from a set of information bits. In channel coding, codes of rates $R = 1/k$ for $k = 2, 3, \ldots$ are usually considered for their simple implementation.

The main drawback of convolutional codes, when implemented using shift-registers, comes from our requirement of small relative payloads (code rates close to one). A convolutional code of rate

[3] In this terminology, convolutional codes are time-invariant trellis codes.

Parity-check matrix

$$\hat{\mathbb{H}} = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad \mathbb{H} = \begin{pmatrix} 1 & 0 & & & & & & \\ 1 & 1 & 1 & 0 & & & & \\ & & 1 & 1 & 1 & 0 & & \\ & & & & 1 & 1 & & \\ & & & & & \ddots & 1 & 0 \\ & & & & & & 1 & 1 & 1 & 0 \end{pmatrix}$$

Syndrome trellis

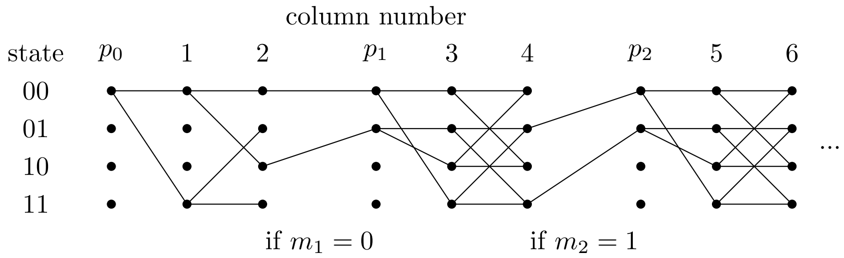

Figure 6.3.1: Example of a parity-check matrix $\mathbb{H}$ formed from the submatrix $\hat{\mathbb{H}}$ ($h = 2, w = 2$) and its corresponding syndrome trellis. The last $h - 1$ submatrices in $\mathbb{H}$ are cropped to achieve the desired relative payload α. The syndrome trellis consists of repeating blocks of $w+1$ columns, where "p_0" and "p_i", $i > 0$, denote the starting and pruning columns, respectively. The column labeled $l \in \{1, 2, \ldots\}$ corresponds to the lth column in the parity-check matrix $\mathbb{H}$.

$R = (k-1)/k$ requires $k-1$ shift registers in order to implement a scheme for $\alpha = 1/k$. Here, unfortunately, the complexity of the Viterbi algorithm in this construction grows exponentially with k. Instead of using puncturing (see Chapter 48 in [83]), which is often used to construct high-rate convolutional codes, we prefer to represent the convolutional code in the dual domain using its parity-check matrix. In fact, Sidorenko and Zyablov [110] showed that optimal decoding of convolutional codes (our binary quantizer) with rates $R = (k-1)/k$ can be carried out in the dual domain on the syndrome trellis with a much lower complexity. This approach is more efficient as $\alpha \to 0$ and thus we choose it for the construction of the codes presented in this chapter.

In the dual domain, a code of length n is represented by a parity-check matrix instead of a generator matrix as is more common for convolutional codes. Working directly in the dual domain allows the Viterbi algorithm to exactly implement the coset quantizer required for the embedding function (6.2.1). The message can be extracted in a straightforward manner by the recipient using the shared parity-check matrix.

6.3.2 Description of Syndrome-Trellis Codes

Although syndrome-trellis codes form a class of convolutional codes and thus can be described using a classical approach with shift-registers, it is advantageous to stay in the dual domain and describe the code directly by its parity-check matrix. The parity-check matrix $\mathbb{H} \in \{0,1\}^{m \times n}$ of a binary syndrome-trellis code of length n and codimension m is obtained by placing a small submatrix $\hat{\mathbb{H}}$ of size $h \times w$ along the main diagonal as in Figure 6.3.1. The submatrices $\hat{\mathbb{H}}$ are placed next to each other and shifted down by one row leading to a sparse and banded $\mathbb{H}$. The height h of the submatrix (called the *constraint height*) is a design parameter that affects the algorithm speed and efficiency (typically, $6 \leq h \leq 15$). The width of $\hat{\mathbb{H}}$ is dictated by the desired ratio of m/n, which coincides with the relative payload $\alpha = m/n$ when no wet pixels are present. If m/n equals to $1/k$ for some $k \in \mathbb{N}$, select $w = k$. For general ratios, find k such that $1/(k+1) < m/n < 1/k$. The matrix $\mathbb{H}$ will contain a mix of submatrices of width k and $k+1$ so that the final matrix $\mathbb{H}$ is of size $m \times n$. In this way, we can create a parity-check matrix for an arbitrary message and code size. The submatrix $\hat{\mathbb{H}}$ acts as an input parameter shared between the sender and the receiver and its choice is discussed in more detail in Section 6.3.4. For the sake of simplicity, in the following description we assume

Forward part of the Viterbi algorithm

```
wght[0] = 0
wght[1,...,2^h-1] = infinity
ix = im = 1
for i = 1,...,num of blocks (submatrices in H) {
 for j = 1,...,w {              // for each column
  for k = 0,...,2^h-1 {          // for each state
   w0 = wght[k] + x[ix]*rho[ix]
   w1 = wght[k XOR H_hat[j]] + (1-x[ix])*rho[ix]
   path[ix][k] = w1 < w0 ? 1 : 0  // C notation
   newwght[k] = min(w0, w1)
  }
  indx++
  wght = newwght
 }
  // prune states
 for j = 0,...,2^(h-1)-1
  wght[j] = wght[2*j + message[im]]
 wght[2^(h-1),...,2^h-1] = infinity
 im++
}
```

Backward part of the Viterbi alg.

```
embedding_cost = wght[0]
state = 0, ix--, im--
for i = num of blcks,...,1 (step -1) {
  for j = w,...,1 (step -1) {
    y[ix] = path[ix][state]
    state = state XOR (y[ix]*H_hat[j])
    ix--
  }
  state = 2*state + message[im]
  im--
}
```

Legend

```
INPUT: x, message, H_hat
  x = (x[1],...,x[n]) cover object
  message = (message[1],...,message[m])
  H_hat[j] = j th column in int notation

OUTPUT: y, embedding_cost
  y = (y[1],...,y[n]) stego object
```

Figure 6.3.2: Pseudocode of the Viterbi algorithm modified for the syndrome trellis.

$m/n = 1/w$ and thus the matrix $\mathbb{H}$ is of the size $b \times (b \cdot w)$, where b is the number of copies of $\hat{\mathbb{H}}$ in $\mathbb{H}$.

Similar to convolutional codes and their trellis representation, every codeword of an STC $\mathcal{C} = \{\mathbf{z} \in \{0,1\}^n | \mathbb{H}\mathbf{z}^T = 0\}$ can be represented as a unique path through a graph called the *syndrome trellis*. Moreover, the syndrome trellis is parametrized by $\mathbf{m}$ and thus can represent members of arbitrary coset $\mathcal{C}(\mathbf{m}) = \{\mathbf{z} \in \{0,1\}^n | \mathbb{H}\mathbf{z}^T = \mathbf{m}\}$. An example of the syndrome trellis is shown in Figure 6.3.1. More formally, the syndrome trellis is a graph consisting of b blocks, each containing $2^h(w+1)$ nodes organized in a grid of $w+1$ columns and 2^h rows. The nodes between two adjacent columns form a bipartite graph, i.e., all edges only connect nodes from two adjacent columns. Each block of the trellis represents one submatrix $\hat{\mathbb{H}}$ used to obtain the parity-check matrix $\mathbb{H}$. The nodes in every column are called *states*.

Each $\mathbf{z} \in \{0,1\}^n$ satisfying $\mathbb{H}\mathbf{z}^T = \mathbf{m}$ is represented as a path through the syndrome trellis which represents the process of calculating the syndrome as a linear combination of the columns of $\mathbb{H}$ with weights given by $\mathbf{z}$. Each path starts in the leftmost all-zero state in the trellis and extends to the right. The path shows the step-by-step calculation of the (partial) syndrome using more and more bits of $\mathbf{z}$. For example, the first two edges in Figure 6.3.1, that connect the state 00 from column p_0 with states 11 and 00 in the next column, correspond to adding ($\mathcal{P}(y_1) = 1$) or not adding ($\mathcal{P}(y_1) = 0$) the first column of $\mathbb{H}$ to the syndrome, respectively.[4] At the end of the first block, we terminate all paths for which the first bit of the partial syndrome does not match m_1. This way, we obtain a new column of the trellis, which will serve as the starting column of the next block. This column merely illustrates the transition of the trellis from representing the partial syndrome $(s_1, \ldots, s_h)$ to $(s_2, \ldots, s_{h+1})$. This operation is repeated at each block transition in the matrix $\mathbb{H}$ and guarantees that 2^h states are sufficient to represent the calculation of the partial syndrome throughout the whole syndrome trellis.

To find the closest stego object, we assign weights to all trellis edges. The weights of the edges entering the column with label l, $l \in \{1, \ldots, n\}$, in the syndrome trellis depend on the lth bit representation of the original cover object $\mathbf{x}$, $\mathcal{P}(x_l)$. If $\mathcal{P}(x_l) = 0$, then the horizontal edges (corresponding to not adding the lth column of $\mathbb{H}$) have a weight of 0 and the edges corresponding to adding the lth column of $\mathbb{H}$ have a weight of ϱ_l. If $\mathcal{P}(x_l) = 1$, the roles of the edges are reversed. Finally, all

[4] The state corresponds to the partial syndrome.

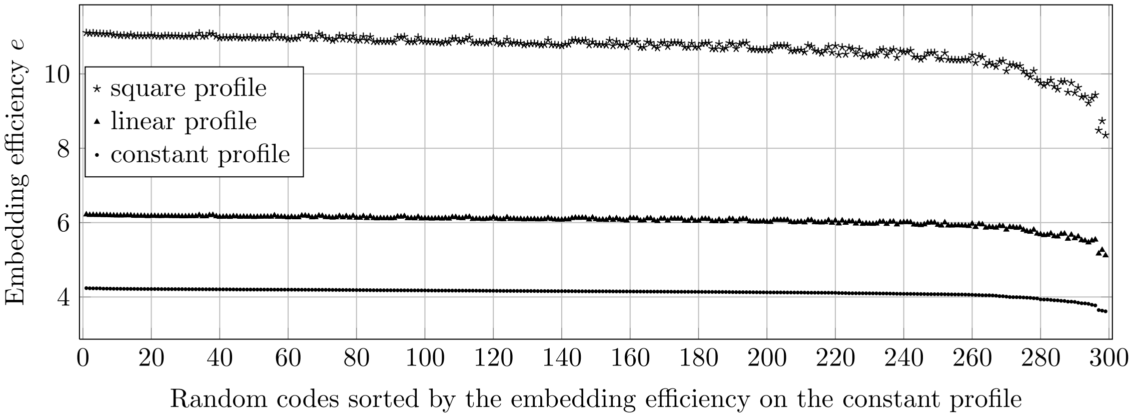

Figure 6.3.3: Embedding efficiency of 300 random syndrome-trellis codes satisfying the design rules for relative payload $\alpha = 1/2$ and constraint height $h = 10$. All codes were evaluated by the Viterbi algorithm with a random cover object of $n = 10^6$ pixels and a random message on the constant, linear, and square profiles. Codes are shown in the order determined by their embedding efficiency evaluated on the constant profile. This experiment suggests that codes good for the constant profile are good for other profiles. Codes designed for different relative payloads have a similar behavior.

edges connecting the individual blocks of the trellis have zero weight.

The embedding problem (6.2.1) for binary embedding can now be optimally solved by the *Viterbi algorithm* with time and space complexity $\mathcal{O}(2^h n)$. This algorithm consists of two parts, the *forward* and the *backward* part. The forward part of the algorithm consists of $n + b$ steps. Upon finishing the ith step, we know the shortest path between the leftmost all-zero state and every state in the ith column of the trellis. Thus in the final, $n + b$th step, we discover the shortest path through the entire trellis. During the backward part, the shortest path is traced back and the parities of the closest stego object $\mathcal{P}(\mathbf{y})$ are recovered from the edge labels. The Viterbi algorithm modified for the syndrome trellis is described in Figure 6.3.2 using a pseudocode.

6.3.3 Implementation Details

The construction of STCs is not constrained to having to repeat the same submatrix $\hat{\mathbb{H}}$ along the diagonal. Any parity-check matrix $\mathbb{H}$ containing at most h nonzero entries along the main diagonal will have an efficient representation by its syndrome trellis and the Viterbi algorithm will have the same complexity $\mathcal{O}(2^h n)$. In practice, the trellis is built on the fly because only the structure of the submatrix $\hat{\mathbb{H}}$ is needed (see the pseudocode in Figure 6.3.2). As can be seen from the last two columns of the trellis in Figure 6.3.1, the connectivity between trellis columns is highly regular which can be used to speed up the implementation by "vectorizing" the calculations.

In the forward part of the algorithm, we need to store one bit (the label of the incoming edge) to be able to reconstruct the path in the backward run. This space complexity is linear and should not cause any difficulty, since for $h = 10$, $n = 10^6$, the total of $2^{10} \cdot 10^6/8$ bytes ($\approx$ 122MB) of space is required. If less space is available, we can always run the algorithm on smaller blocks, say $n = 10^4$, without any noticeable performance drop. If we are only interested in the total distortion $D(\mathbf{y})$ and not the stego object itself, this information does not need to be stored at all and only the forward run of the Viterbi algorithm is required.

6.3.4 Design of Good Syndrome-Trellis Codes

A natural question regarding practical applications of syndrome-trellis codes is how to optimize the structure of $\hat{\mathbb{H}}$ for fixed parameters h and w and a given profile. If $\hat{\mathbb{H}}$ depended on the distortion profile, the profile would have to be somehow communicated to the receiver. Fortunately, this is not the case and a submatrix $\hat{\mathbb{H}}$ optimized for one profile seems to be good for other profiles as well. In this section, we study these issues experimentally and describe a practical algorithm for obtaining good submatrices.

Let us suppose that we wish to design a submatrix $\hat{\mathbb{H}}$ of size $h \times w$ for a given constraint height h and relative payload $\alpha = 1/w$. In [12], authors describe several methods for calculating the expected distortion of a given convolutional code when used in the source-coding problem with Hamming measure (uniform distortion profile). Unfortunately, the computational complexity of these algorithms do not permit us to use them for the code design. Instead, we rely on estimates obtained from embedding a pseudo-random message into a random cover object. The author was unable to find a better algorithm than an exhaustive search guided by some simple design rules.

First, $\hat{\mathbb{H}}$ should not have identical columns because the syndrome trellis would contain two or more different paths with exactly the same weight, which would lead to an overall decrease in performance. By running an exhaustive search over small matrices, we have observed that the best submatrices $\hat{\mathbb{H}}$ had ones in the first and last rows. For example, when $h = 7$ and $w = 4$, more than 97% of the best 1000 codes obtained from the exhaustive search satisfied this rule. Thus, we searched for good matrices among those that did not contain identical columns and with all bits in the first and last rows set to 1 (the remaining bits were assigned at random). In practice, we randomly generated $10 - 1000$ submatrices satisfying these rules and estimated their performance (embedding efficiency) experimentally by running the Viterbi algorithm with random covers and messages. For a reliable estimate, cover objects of size at least $n = 10^6$ are required.

To investigate the stability of the design w.r.t. to the profile, the following experiment was conducted. We fixed $h = 10$ and $w = 2$, which correspond to a code with $\alpha = 1/2$. The code design procedure was simulated by randomly generating 300 submatrices $\hat{\mathbb{H}}_1, \ldots, \hat{\mathbb{H}}_{300}$ satisfying the above design rules. The goodness of the code was evaluated using the embedding efficiency ($e = m/D(\mathbf{x}, \mathbf{y})$) by running the Viterbi algorithm on a random cover object (of size $n = 10^6$) and with a random message. This was repeated independently for all three profiles from Section 6.1.2. Figure 6.3.3 shows the embedding efficiency after ordering all 300 codes by their performance on the constant profile. Because the codes with a high embedding efficiency on the constant profile exhibit high efficiency for the other profiles, we consider the code design to be stable w.r.t. the profile and use these matrices with other profiles in practice. All further results are generated by using these matrices.

6.3.5 Wet Paper Channel

In this section, we investigate how STCs can be used for the wet paper channel described by relative wetness $\tau = |\{i|\varrho_i = \infty\}|/n$ with a given distortion profile of dry pixels. Although the STCs can be directly applied to this problem, the probability of not being able to embed a message without changing any wet pixel may be positive and depends on the number of wet pixels, the payload, and the code. The goal is to make this probability very small or to make sure that the number of wet pixels that must be changed is small (e.g., one or two). We now describe two different approaches to address this problem.

Let us assume that the wet channel is iid with probability of a pixel being wet $0 \leq \tau < 1$. This assumption is plausible because the cover pixels can be permuted using a stego key before embedding. For the wet paper channel, the relative payload is defined w.r.t. the dry pixels as $\alpha = m/|\{i|\rho_i < \infty\}|$. When designing the code for the wet paper channel with n-pixel covers, relative wetness τ, and desired relative payload α, the parity-check matrix $\mathbb{H}$ has to be of the size $[(1-\tau)\alpha n] \times n$.

The random permutation makes the Viterbi algorithm less likely to fail to embed a message without having to change some wet pixels. The probability of failure, p_{w}, decreases with decreasing α and τ and it also depends on the constraint height h. From practical experiments with $n = 10^6$ cover pixels, $\tau = 0.8$, and $h = 10$, we estimated from 1000 independent runs $p_{\mathrm{w}} \doteq 0.24$ for $\alpha = 1/2$, $p_{\mathrm{w}} \doteq 0.009$ for $\alpha = 1/4$, and $p_{\mathrm{w}} \doteq 0$ for $\alpha = 1/10$. In practice, the message size m can be used as a seed for the pseudo-random number generator. If the embedding process fails, embedding $m - 1$ bits leads to a different permutation while embedding roughly the same amount of message. In k trials, the probability of having to modify a wet pixel is at most p_{w}^k, which can be made arbitrarily

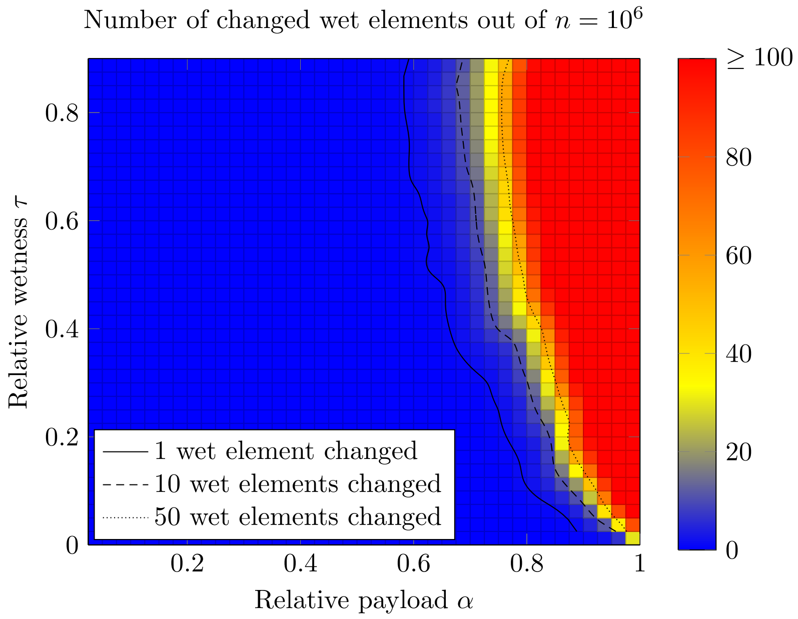

Figure 6.3.4: Average number of wet pixels out of $n = 10^6$ that need to be changed to find a solution to (6.2.1) using STCs with $h = 11$.

small.

Alternatively, the sender may allow a small number of wet pixels to be modified, say one or two, without affecting the statistical detectability in any significant manner. Making use of this fact, one can set the distortion of all wet cover pixels to $\hat{\varrho}_i = C$, $C > \sum_{\varrho_i<\infty} \varrho_i$ and $\hat{\varrho}_i = \varrho_i$ for i dry. The weight c of the best path through the syndrome trellis obtained by the Viterbi algorithm with distortion $\hat{\varrho}_i$ can be written in the form $c = n_c C + c'$, where n_c is the smallest number of wet cover pixels that had to be changed and c' is the smallest weight of the path over the pixels that are allowed to be changed.

Figure 6.3.4 shows the average number of wet pixels out of $n = 10^6$ required to be changed in order to solve (6.2.1) for STCs with $h = 11$. The exact value of ϱ_i is irrelevant in this experiment as long as it is finite. This experiment suggests that STCs can be used with arbitrary τ as long as $\alpha \leq 0.7$. As can be seen from Figure 6.3.5, increasing the amount of wet pixels does not lead to any noticeable difference in embedding efficiency for constant profile. Similar behavior has been observed for other profiles and holds as long as the number of changed wet pixels is small.

6.3.6 Experimental Results

We have implemented the Viterbi algorithm in C++ and optimized its performance by using Streaming SIMD Extensions instructions. Based on the distortion profile, the algorithm chooses between the float and 1 byte unsigned integer data type to represent the weight of the paths in the trellis. The following results were obtained using an Intel Core2 X6800 2.93GHz CPU machine utilizing a single CPU core.

Using the search described in Section 6.3.4, we found good syndrome-trellis codes of constraint height $h \in \{6, \dots, 12\}$ for relative payloads $\alpha = 1/w$, $w \in \{1, \dots, 20\}$. Some of these codes can be found in [30, Table 1]. In practice, almost every code satisfying the design rules is equally good. This fact can also be seen from Figure 6.3.3, where 300 random codes are evaluated over different profiles.

The effect of the profile shape on the coding loss for $\varrho(x) \approx x^d$ as a function of d is shown in Figure 6.3.6. The coding loss increases with decreasing relative payload α. This effect can be compensated by using a larger constraint height h.

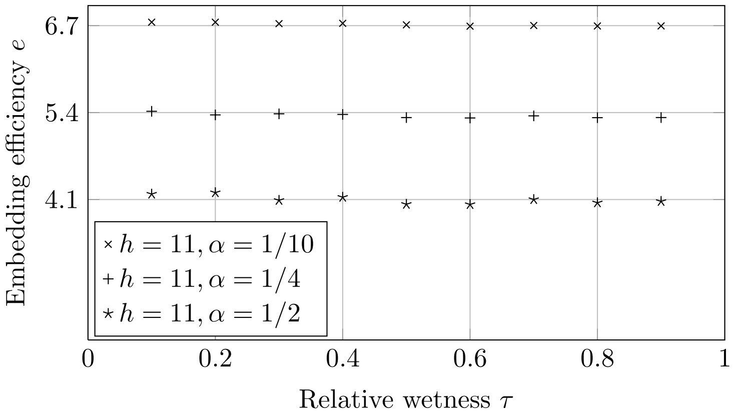

Figure 6.3.5: Effect of relative wetness τ of the wet paper channel with a constant profile on the embedding efficiency of STCs. The distortion was calculated w.r.t. the changed dry pixels only and $\alpha = m/(n - \tau n)$. Each point was obtained by quantizing a random vector of $n = 10^6$ pixels.

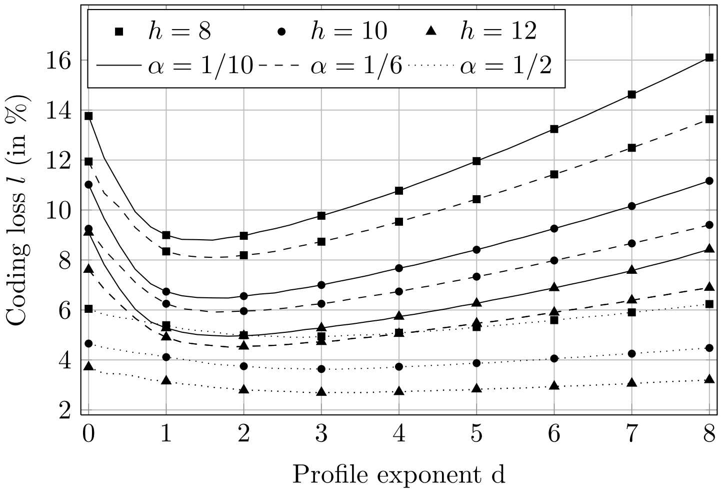

Figure 6.3.6: Comparison of the coding loss of STCs as a function of the profile exponent d for different payloads and constraint heights of STCs. Each point was obtained by quantizing a random vector of $n = 10^6$ pixels.

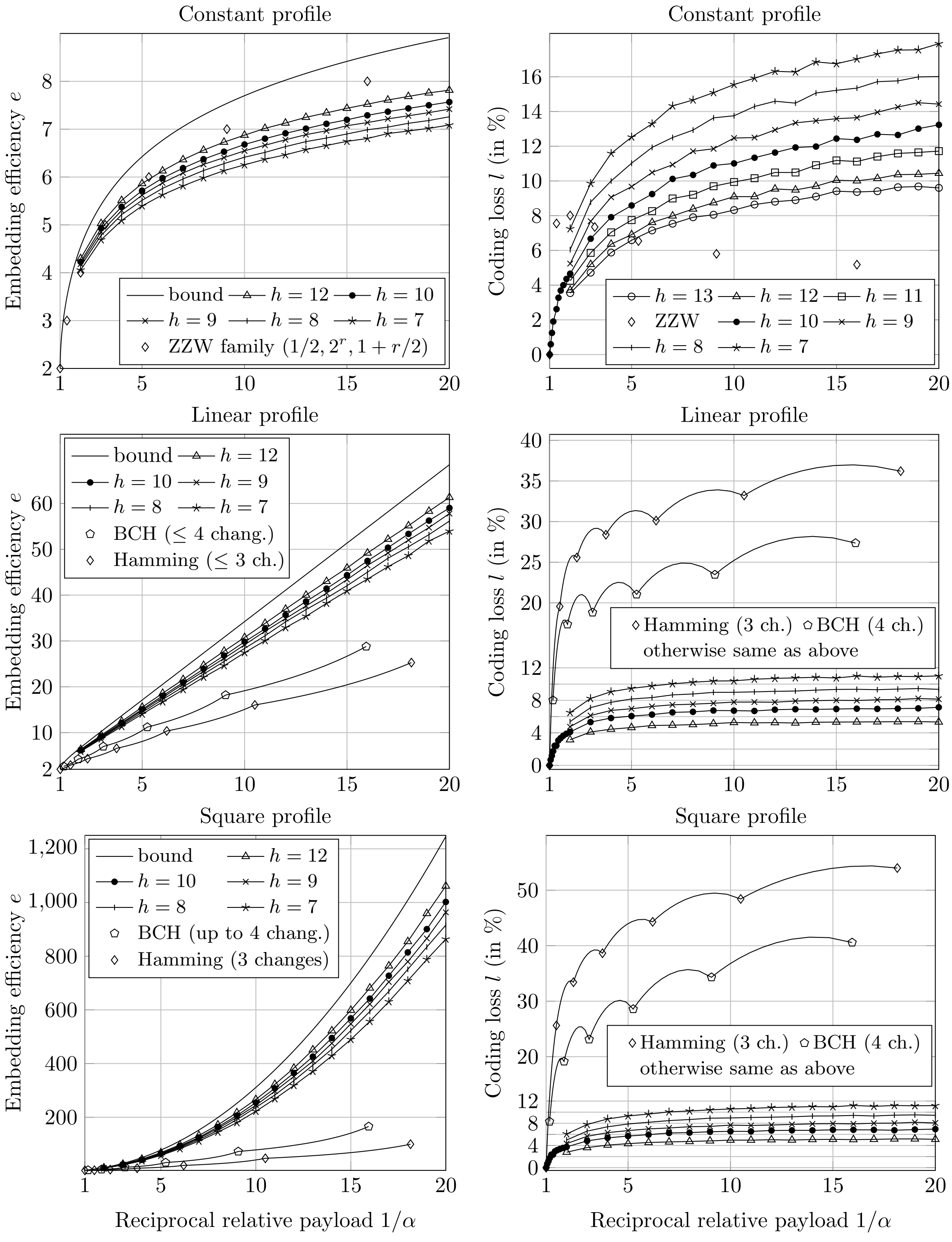

Figure 6.3.7: Embedding efficiency and coding loss of syndrome-trellis codes for three distortion profiles. Each point was obtained by running the Viterbi algorithm with $n = 10^6$ cover pixels. Hamming [74] and BCH [128] codes were applied on a block-by-block basis on cover objects with $n = 10^5$ pixels with a brute-force search making up to three and four changes, respectively. The line connecting a pair of Hamming or BCH codes represents the codes obtained by their block direct sum. For clarity, we present the coding loss results in range $\alpha \in [0.5, 1]$ only for constraint height $h = 10$ of the syndrome-trellis codes.

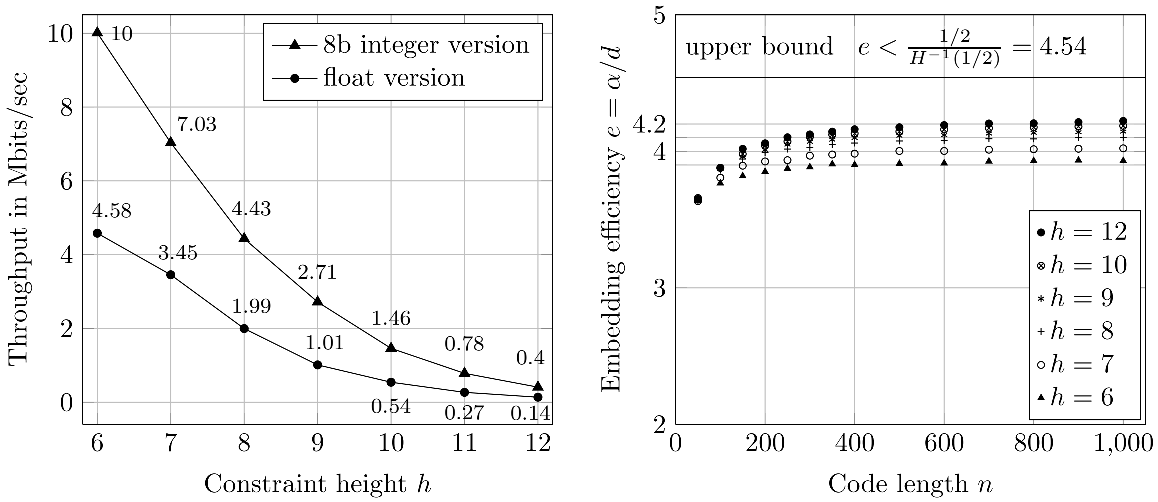

Figure 6.3.8: Results for the syndrome-trellis codes designed for relative payload $\alpha = 1/2$. Left: Average number of cover pixels ($\times 10^6$) quantized per second (throughput) shown for different constraint heights and two different implementations. Right: Average embedding efficiency for different code lengths n (the number of cover pixels), constraint heights h, and a constant distortion profile. Codes of length $n > 1000$ have similar performance as for $n = 1000$. Each point was obtained as an average over 1000 samples.

Figure 6.3.7 shows the comparison of syndrome-trellis codes for three profiles with other codes which are known for a given profile. The ZZW family [130] applies only to the constant profile. For a given relative payload α and constraint height h, the same submatrix $\hat{\mathbb{H}}$ was used for all profiles. This demonstrates the versatility of the proposed construction, since the information about the profile does not need to be shared, or, perhaps more importantly, the profile does not need to be known a priori for a good performance.

Figure 6.3.8 shows the average throughput (the number of cover pixels n quantized per second) based on the used data type. In practice, 1–5 seconds were enough to process a cover object with $n = 10^6$ pixels. In the same figure, we show the embedding efficiency obtained from very short codes for the constant profile. This result shows that the average performance of syndrome-trellis codes quickly approaches its maximum w.r.t. n. This is again an advantage, since some applications may require short blocks.

6.3.7 STCs in Context of Other Works

The concept of dividing a set of samples into different bins (the so-called binning) is a common tool used for solving many information-theoretic and also data-hiding problems [86]. From this point of view, the steganographic embedding problem is a pure source-coding problem, i.e., given cover $\mathbf{x}$, what is the "closest" stego object $\mathbf{y}$ in the bin indexed by the message. In digital watermarking, the same problem is extended by an attack channel between the sender and the receiver, which calls for a combination of good source and channel codes. This combination can be implemented using nested convolutional (trellis) codes and is better known as Dirty-paper codes [122]. Different practical application of the binning concept is in the distributed source coding problem [98]. Convolutional codes are attractive for solving these problems mainly because of the existence of the optimal quantizer – the Viterbi algorithm.

6.4 Multi-Layered Construction

Although it is straightforward to extend STCs to non-binary alphabets and thus apply them to q-ary embedding operations, their complexity rapidly increases (the number of states in the trellis increases from 2^h to q^h for constraint height h), limiting thus their performance in practice. In this section, we introduce a simple layered construction which has been largely motivated by [132] and can be considered as a generalization of this work. The main idea is to decompose the PLS and the DLS problems with a non-binary embedding operation into a sequence of similar problems with a binary embedding operation. Any solution to the binary PLS embedding problem, such as STCs, can then be used. This decomposition turns out to be optimal if each binary embedding problem is solved optimally. The multi-layered construction was described in [28].

According to (6.1.7), the binary coding algorithm for the PLS (5.2.6) or the DLS (5.2.4) is optimal if and only if it modifies each cover pixel with probability

$$\pi_i = \frac{\exp(-\lambda\varrho_i)}{1+\exp(-\lambda\varrho_i)}. \tag{6.4.1}$$

For a fixed value of λ, the values ϱ_i, $i = 1, \dots, n$, form sufficient statistic for π.

A solution to the PLS with a binary embedding operation can be used to derive the following "Flipping lemma" that we will heavily use later in this section.

Lemma 6.1 (Flipping lemma). *Given a set of probabilities $\{p_i\}_{i=1}^n$, the sender wants to communicate $m = \sum_{i=1}^n h(p_i)$ bits by sending bit strings $\mathbf{y} = \{y_i\}_{i=1}^n$ such that $P(y_i = 0) = p_i$. This can be achieved by a PLS with a binary embedding operation on $\mathcal{I} = \mathcal{I}_i = \{0,1\}$ for all i by embedding the payload in cover $x_i = [p_i < 1/2]$ with non-negative per-pixel costs $\varrho_i = \ln(\tilde{p}_i/(1-\tilde{p}_i))$, $\tilde{p}_i = \max\{p_i, 1-p_i\}$.*

Proof. Without loss of generality, let $\lambda = 1$. Since the inverse of $f(z) = \ln(z/(1-z))$ on $[0,1]$ is $f^{-1}(z) = \exp(z)/(1+\exp(z))$, by (6.4.1) the cost ϱ_i causes x_i to change to $y_i = 1 - x_i$ with probability $P(y_i \neq x_i | x_i) = f^{-1}(-\varrho_i) = 1 - \tilde{p}_i$. Thus, $P(y_i = 0 | x_i = 1) = f^{-1}(-\varrho_i) = p_i$ and $P(y_i = 0 | x_i = 0) = 1 - f^{-1}(-\varrho_i) = p_i$ as required. □

Now, let $|\mathcal{I}_i| = 2^L$ for some integer $L \geq 0$ and let $\mathcal{P}_1, \dots, \mathcal{P}_L$ be parity functions uniquely describing all 2^L elements in $\mathcal{I}_i$, i.e., $(x_i \neq y_i) \Rightarrow \exists j, \mathcal{P}_j(x_i) \neq \mathcal{P}_j(y_i)$ for all $x_i, y_i \in \mathcal{I}_i$ and all $i \in \{1, \dots, n\}$. For example, $\mathcal{P}_j(x)$ can be defined as the jth LSB of x. The individual sets $\mathcal{I}_i$ can be enlarged to satisfy the size constraint by setting the costs of added elements to ∞.

The optimal algorithm for the PLS (5.2.6) and the DLS (5.2.4) problems sends the stego symbols by sampling from the optimal distribution (6.1.2) with some λ. Let $\mathbf{Y}_i$ be the random variable defined over $\mathcal{I}_i$ representing the ith stego symbol. Due to the assigned parities, $\mathbf{Y}_i$ can be represented as $\mathbf{Y}_i = (Y_i^1, \dots, Y_i^L)$ with Y_i^j corresponding to the jth parity function. We construct the embedding algorithm by induction over L, the number of layers. By the chain rule, for each i the entropy $H(\mathbf{Y}_i)$ can be decomposed into

$$H(\mathbf{Y}_i) = H(Y_i^1) + H(Y_i^2, \dots, Y_i^L | Y_i^1) \tag{6.4.2}$$

This tells us that $H(Y_i^1)$ bits should be embedded by changing the first parity of the ith pixel. In fact, the parities should be distributed according to the marginal distribution $P(Y_i^1)$. Using the Flipping lemma, this task is equivalent to a PLS, which can be realized in practice using STCs as reviewed in Section 6.3. To summarize, in the first step we embed $m_1 = \sum_{i=1}^n H(Y_i^1)$ bits on average.

After the first layer is embedded, we obtain the parities $\mathcal{P}_1(y_i)$ for all stego pixels. This allows us to calculate the conditional probability $P(Y_i^2, \dots, Y_i^L | Y_i^1 = \mathcal{P}_1(y_i))$ and use the chain rule again, for example w.r.t. Y_i^2. In the second layer, we embed $m_2 = \sum_{i=1}^n H(Y_i^2 | Y_i^1 = \mathcal{P}_1(y_i))$ bits on average. In total, we have L such steps fixing one parity value at a time knowing the result of the previous parities. Finally, we send the values y_i corresponding to the obtained parities.

If all individual layers are implemented optimally, we send $m = m_1 + \dots + m_L$ bits on average. By the chain rule, this is exactly $H(\mathbf{Y}_i)$ in every pixel, which proves the optimality of this construction.

Algorithm 6.1 ±1 embedding implemented with 2-layers of STCs and embedding the payload of m bits

```
Require: $\mathbf{x} \in \mathcal{X} = \{\mathcal{I}\}^n \triangleq \{0, \ldots, 255\}^n$
         $\rho_i(\mathbf{x}, z) \in [-K, +K],\ z \in \mathcal{I}_i \triangleq \{x_i - 1, x_i, x_i + 1\}$
define $\mathcal{P}_1(z) = z \bmod 2$, $\mathcal{P}_2(z) = [(z \bmod 4) > 1]$
forbid other colors by $\rho_i(\mathbf{x}, z) = C \gg K,\ z \notin \mathcal{I}_i \cap \mathcal{I}$
find $\lambda \geq 0$ such that distr. $\pi$ over $\mathcal{X}$ satisfies $H(\pi) = m$

define $p''_i = Pr_\pi(\mathcal{P}_2(Y_i) = 0)$, set $m_2 = \sum_i h(p''_i)$, $\mathbf{x}'' \in \{0,1\}^n$ with $x''_i = [p''_i < 1/2]$, and $\varrho''_i = |\ln(p''_i/(1 - p''_i))|$
embed $m_2$ bits with binary STC into $\mathbf{x}''$ with costs $\varrho''_i$ and produce new vector $\mathbf{y}'' = (y''_1, \ldots, y''_n) \in \{0,1\}^n$

define $p'_i = Pr_\pi(\mathcal{P}_1(Y_i) = 0|\mathcal{P}_2(Y_i) = y''_i)$, $\mathbf{x}' \in \{0,1\}^n$ with $x'_i = [p'_i < 1/2]$, and $\varrho'_i = |\ln(p'_i/(1 - p'_i))|$
embed $m - m_2$ bits with binary STC into $\mathbf{x}'$ with costs $\varrho'_i$ and produce a new vector $\mathbf{y}' = (y'_1, \ldots, y'_n) \in \{0,1\}^n$

set $y_i \in \mathcal{I}_i$ such that $\mathcal{P}_2(y_i) = y''_i$ and $\mathcal{P}_1(y_i) = y'_i$
return stego image $\mathbf{y} = (y_1, \ldots, y_n)$
message can be extracted using STCs from $(\mathcal{P}_2(y_1), \ldots, \mathcal{P}_2(y_n))$ and $(\mathcal{P}_1(y_1), \ldots, \mathcal{P}_1(y_n))$
```

In theory, the order in which the parities are being fixed can be arbitrary. As is shown in the following example, the order is important for practical realizations when STCs are used. In all our experiments, we start with the *most* significant bits ending with the LSBs. Algorithm 6.1 describes the necessary steps required to implement ±1 embedding with arbitrary costs using two layers of STCs.

In practice, the number of bits hidden in every layer, m_j, needs to be communicated to the receiver. The number m_j is used as a seed for a pseudo-random permutation used to shuffle all bits in the jth layer. If, due to large payload and wetness, STCs cannot embed a given message, we try a different permutation by embedding a slightly different number of bits.

Example 6.1 (±1 embedding)**.** For simplicity, let $x_i = 2$, $\mathcal{I}_i = \{1, 2, 3\}$, $\rho_i(1) = \rho_i(3) = 1$, and $\rho_i(2) = 0$ for $i \in \{1, \ldots, n\}$ and large n. For such ternary embedding, we use two LSBs as their parities. Suppose we want to solve the problem the PLS problem (5.2.6) with $\alpha = 0.9217$, which leads to $\lambda = 2.08$, $P(Y_i = 1) = P(Y_i = 3) = 0.1$, and $P(Y_i = 2) = 0.8$. To make $|\mathcal{I}_i|$ a power of two, we also include the symbol 0 and define $\rho_i(0) = \infty$ which implies $P(Y_i = 0) = 0$. Let $y_i = (y_i^2, y_i^1)$ be a binary representation of $y_i \in \{0, \ldots, 3\}$, where y_i^1 is the LSB of y_i.

Starting from the LSBs as in [132], we obtain $P(Y_i^1 = 0) = 0.8$. If the LSB needs to be changed, then $P(Y_i^2 = 0|Y_i^1 = 1) = 0.5$ whereas $P(Y_i^2 = 0|Y_i^1 = 0) = 0$. In practice, the first layer can be realized by any syndrome-coding scheme minimizing the number of changes and embedding $m_1 = n \cdot h(0.2)$ bits. The second layer must be implemented with wet paper codes [45], since we need to embed either one bit or leave the pixel unchanged (the relative payload is 1).

If the weights of symbols 1 and 3 were slightly changed, however, we would have to use STCs in the second layer, which causes a problem due to the large relative payload ($\alpha = 1$) combined with large wetness ($\tau = 0.8$) (see Figure 6.3.4). The opposite decomposition starting with the MSB y_i^2 will reveal that $P(Y_i^2 = 0) = 0.1$, $P(Y_i^1 = 0|Y_i^2 = 0) = 0$, and $P(Y_i^1 = 0|Y_i^2 = 1) = 0.8/0.9$. Both layers can now be easily implemented by STCs since here the wetness is not as severe ($\tau = 0.1$).

6.5 Practical Embedding Constructions

In this section, we show some applications of the proposed methodology for spatial and transform domain (JPEG) steganography. In the past, most embedding schemes were constrained by practical ways of how to encode the message so that the receiver can read it. Problems such as "shrinkage" in F5 [125, 47] or in MMx [74] arose from this practical constraint. By being able to solve the PLS and DLS problems close to the bound for an arbitrary additive distortion function,[5] steganographers now have much more freedom in designing new embedding algorithms. They only need to select the distortion function and then apply the proposed framework. The only task left to the steganographer is the choice of the distortion function D. It should be selected so that it correlates with statistical detectability. Instead of delving into the difficult problem of how to select the best D, we provide a few examples of additive distortion measures motivated by recent developments in steganography and show their performance when blind steganalysis is used. The problem of optimizing the distortion function is investigated in Chapter 7.

In the examples below, we tested the embedding schemes using the blind feature-based steganalysis described in Section 2.3.

6.5.1 DCT Domain Steganography

To apply the proposed framework, we first need to design an additive distortion function which can be tested by simulating the embedding as if the best codes were available. Finally, the the most promising approach is implemented using STCs. We assume the cover to be a grayscale bitmap image which we JPEG compress to obtain the cover image. Let $\mathcal{A}$ be a set of indices corresponding to AC DCT coefficients after the block-DCT transform and let c_i be the ith AC coefficient before it is quantized with the quantization step q_i for $i \in \mathcal{A}$. We let $\mathcal{X}$ represent the set of all vectors containing quantized AC DCT coefficients divided by their corresponding quantization step. In ordinary JPEG compression, the values c_i are quantized to $x_i \triangleq [c_i/q_i]$.

6.5.1.1 Proposed Distortion Functions

We define a binary embedding operation $\mathcal{I}_i \triangleq \{x_i, \overline{x}_i\}$ by $\overline{x}_i = x_i + \operatorname{sign}(c_i/q_i - x_i)$, where $\operatorname{sign}(x)$ is 1 if $x > 0$, -1 if $x < 0$ and $\operatorname{sign}(0) \in \{-1, 1\}$ uniformly at random. In simple words, x_i is a quantized AC DCT coefficient and $\overline{x}_i$ is the same coefficient when quantized in the opposite direction. Let $e_i = |c_i/q_i - x_i|$ be the quantization error introduced by JPEG compression. By replacing x_i with $\overline{x}_i$ the error becomes $|c_i/q_i - \overline{x}_i| = 1 - e_i$. If $e_i = 0.5$, then the direction where c_i/q_i is rounded depends on the implementation of the JPEG compressor and only small perturbation of the original image may lead to different results. Let $\mathcal{P}(x) = x \bmod 2$. By construction, $\mathcal{P}$ satisfies the property of a parity function, $\mathcal{P}(x_i) \neq \mathcal{P}(\overline{x}_i)$. The distortion function is assumed to be in the form $D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} \varrho_i \cdot [x_i \neq y_i]$, where $n = |\mathcal{A}|$.

The following four approaches utilizing the values of e_i and q_i were considered. All methods assign $\varrho_i = \infty$ when $c_i/q_i \in (-0.5, 0.5)$ and differ in the definition of the remaining values ϱ_i as follows:

- **S1:** $\varrho_i = 1 - 2e_i$ if $c_i/q_i \notin (-0.5, 0.5)$ (as in perturbed quantization [43]),
- **S2:** $\varrho_i = q_i(1-2e_i)$ if $c_i/q_i \notin (-0.5, 0.5)$ (the same as S1 but ϱ_i is weighted by the quantization step),
- **S3:** $\varrho_i = 1$ if $c_i/q_i \in (-1, -0.5] \cup [0.5, 1)$ and $\varrho_i = 1 - 2e_i$ otherwise, and
- **S4:** $\varrho_i = q_i$ if $c_i/q_i \in (-1, -0.5] \cup [0.5, 1)$ and $\varrho_i = q_i(1 - 2e_i)$ otherwise which is a similar weight assignment as proposed in [102].

To see the importance of the side-information in the form of the uncompressed cover image, we also include in our tests the nsF5 [47] algorithm, which can be represented in our formalism as $x_i = [c_i/q_i]$, $\overline{x}_i = x_i - \operatorname{sign}(x_i)$, and $\varrho_i = \infty$ if $x_i = 0$ and $\varrho_i = 1$ otherwise. This way, we always have $|\overline{x}_i| < |x_i|$. The nsF5 embedding minimizes the number of changes to non-zero AC DCT coefficients.

[5] The additivity constraint can be relaxed and more general distortion measures can be used with the PLS and DLS problems in practice [27].

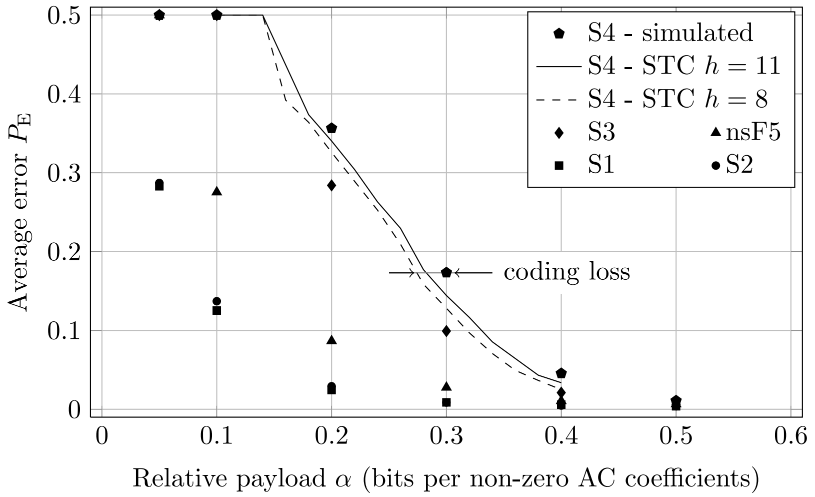

Figure 6.5.1: Comparison of methods with four different weight-assignment strategies S1–S4 and nsF5 as described in Section 6.5.1 when simulated as if the best coding scheme was available. The performance of strategy S4 when practically implemented using STCs with $h = 8$ and $h = 11$ is also shown.

6.5.1.2 Steganalysis Setup and Experimental Results

The proposed strategies were tested on a database of 6,500 digital camera images prepared as described in [79, Sec. 4.1] so that their smaller size was 512 pixels. The JPEG quality factor 75 was used for compression. The steganalyzer employed the 548-dimensional CC-PEV feature set [77]. Figure 6.5.1 shows the minimum average classification error P_E achieved by simulating each strategy on the bound using the PLS formulation. The strategies S1 and S2, which assign zero cost to coefficients $c_i/q_i = 0.5$, were worse than the nsF5 algorithm that does not use any side-information. On the other hand, strategy S4, which also utilizes the knowledge about the quantization step, was the best. By implementing this strategy, we have to deal with a wet paper channel which can be well modeled by a linear profile with relative wetness $\tau \approx 0.6$ depending on the image content. We have implemented strategy S4 using STCs, where wet pixels were handled by setting $\varrho_i = C$ for a sufficiently large C. As seen from the results using STCs, payloads below 0.15 bits per non-zero AC DCT coefficient were undetectable using our steganalyzer.

Note that our strategies utilized only the information obtainable from a single AC DCT coefficient. In reality, ϱ_i will likely depend on the local image content, quantization errors, and quantization steps. We leave the problem of optimizing D w.r.t. statistical detectability for our future research.

6.5.2 Spatial Domain Steganography

To demonstrate the merit of the STC-based multi-layered construction, we present a practical embedding scheme that was largely motivated by [94] and [27]. Single per-pixel distortion function $\rho_{i,j}(y_{i,j})$ should assign the cost of changing i,jth pixel $x_{i,j}$, first, from its neighborhood and then also based on the new value $y_{i,j}$. Changes made in smooth regions often tend to be highly detectable by blind steganalysis which should lead to high distortion values. On the other hand, pixels which are in busy and hard-to-model regions can be changed more often.

6.5.2.1 Proposed Distortion Functions

We design our distortion function based on a model discussed in Section 5.7 built from a set of all straight 4-pixel lines in 4 different orientations (see Figure 6.5.2). Based on the set of all such cliques, we define $\rho_{i,j}(y_{i,j})$ to be an additive approximation (5.7.2) of (5.7.1), i.e., we define the distortion

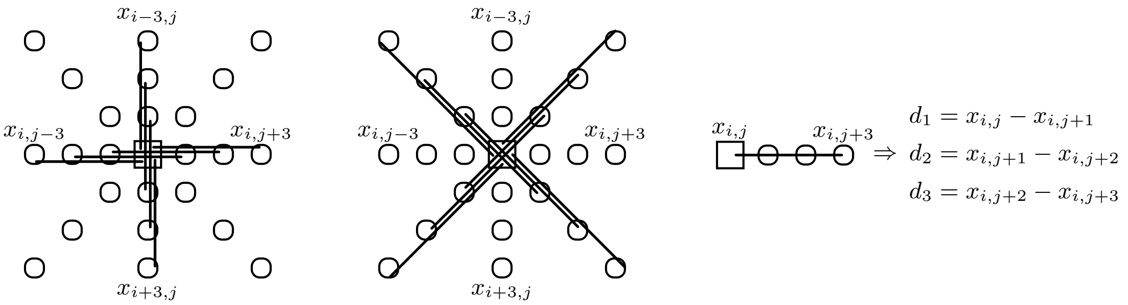
$x_{i-3,j}$
$x_{i,j-3}$
$x_{i,j+3}$
$x_{i+3,j}$
$x_{i,j}$
$d_1 = x_{i,j} - x_{i,j+1}$
$d_2 = x_{i,j+1} - x_{i,j+2}$
$d_3 = x_{i,j+2} - x_{i,j+3}$

Figure 6.5.2: Set of 4-pixel cliques used for calculating the distortion for digital images represented in the spatial-domain. The final distortion $\rho_{i,j}(y_{i,j})$ is obtained as a sum of terms penalizing the change in pixel $x_{i,j}$ measured w.r.t. each clique containing $x_{i,j}$.

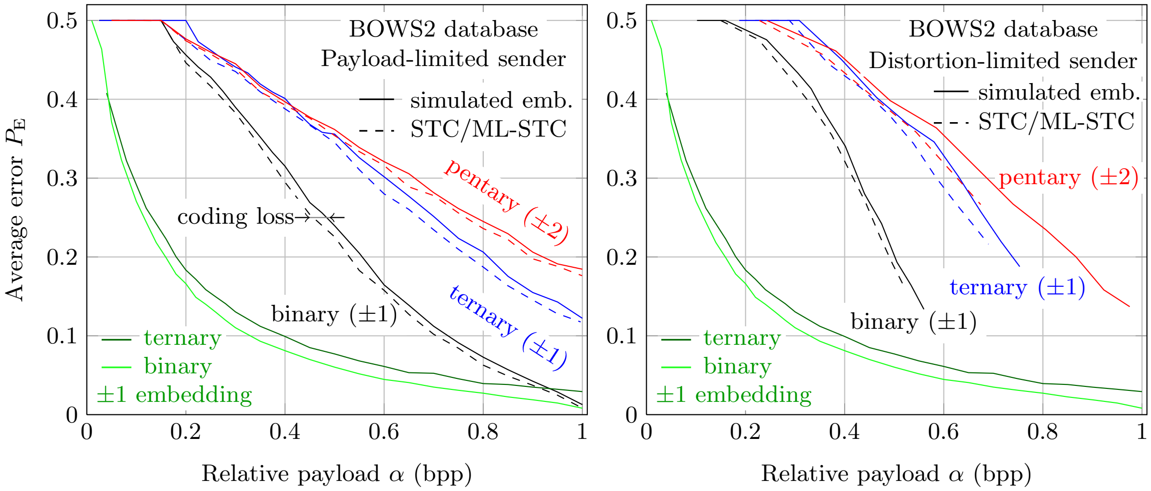

Figure 6.5.3: Comparison of ±1 embedding with optimal binary and ternary coding with embedding algorithms based on the additive distortion measure (6.5.1) using embedding operations of three different cardinalities.

measure $D(\mathbf{y}) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2}\rho_{i,j}(y_{i,j})$ by

$$\rho_{i,j}(y_{i,j}) = \sum_{\substack{k,l,m\in\{-255,\dots,255\}\\ s\in\{\rightarrow,\nearrow,\uparrow,\nwarrow\}}} w_{k,l,m}|g^s_{k,l,m}(\mathbf{x}) - g^s_{k,l,m}(y_{i,j}\mathbf{x}_{\sim i,j})|, \tag{6.5.1}$$

where $w_{k,l,m} = 1/(1+\sqrt{k^2+l^2+m^2})$ are heuristically chosen weights and $\mathbb{G}^s(\mathbf{x}) = (g^s_{k,l,m}(\mathbf{x}))$ defined similarly as in (5.7.11) with the difference vector computed from *four* consecutive pixels $(d^s_{i,j}, d^s_{i,j+1}, d^s_{i,j+2}) = (k,l,m)$ for $s \in \{\rightarrow,\nearrow,\uparrow,\nwarrow\}$.

6.5.2.2 Steganalysis Setup and Experimental Results

All tests were carried out on the BOWS2 database [5] containing approximately 10,800 grayscale images with a fixed size of 512×512 pixels coming from rescaled and cropped natural images of various sizes. Steganalysis was implemented using the second-order SPAM feature set with $T = 3$ [92].

Figure 6.5.3 contains the comparison of embedding algorithms implementing the PLS and DLS with the costs (6.5.1). All algorithms are contrasted with ±1 embedding simulated on the binary and ternary bounds. To compare the effect of practical codes, we first simulated the embedding algorithm as if the best codes were available and then compared these results with algorithms implemented using STCs with $h = 10$. Both types of senders are implemented with binary, ternary

($\mathcal{I}_i = \{x_i - 1, \ldots, x_i + 1\}$), and pentary ($\mathcal{I}_i = \{x_i - 2, \ldots, x_i + 2\}$) embedding operations. Before embedding, the binary embedding operation was initialized to $\mathcal{I}_i = \{x_i, y_i\}$ with y_i randomly chosen from $\{x_i - 1, x_i + 1\}$. The reported payload for the DLS with a fixed D_ϵ was calculated as an average over the whole database after embedding.

The relative horizontal distance between the corresponding dashed and solid lines in Figure 6.5.3 is bounded by the coding loss. Most of the proposed algorithms are undetectable for relative payloads $\alpha \leq 0.2$ bits per pixel (bpp). For payloads $\alpha \leq 0.5$, the DLS is more secure. For larger payloads, the distortion measure seems to fail to capture the statistical detectability correctly and thus the algorithms are more detectable than when implemented in the payload-limited regime.

6.6 Conclusion

The concept of embedding in steganography that minimizes a distortion function is connected to many basic principles used for constructing embedding schemes for complex cover sources today, including the principle of minimal-embedding-impact [47], approximate model-preservation [94], or the Gibbs construction [27]. The current work describes a complete practical framework for constructing steganographic schemes that embed by minimizing an additive distortion function. Once the steganographer specifies the form of the distortion function, the proposed framework provides all essential tools for constructing practical embedding schemes working close to their theoretical bounds. The methods are not limited to binary embedding operations and allow the embedder to choose the amplitude of embedding changes dynamically based on the cover-image content. The distortion function or the embedding operation do not need to be shared with the recipient. In fact, they can even change from image to image. The framework can be thought of as an off-the-shelf method that allows practitioners to concentrate on the problem of designing the distortion measure instead of the problem of how to construct practical embedding schemes.

The merit of the proposed algorithms is demonstrated experimentally by implementing them for the JPEG and spatial domains and showing an improvement in statistical detectability as measured by state-of-the-art blind steganalyzers. We have demonstrated that larger embedding changes provide a significant gain in security when placed adaptively. Finally, the construction is not limited to embedding with larger amplitudes but can be used, e.g., for embedding in color images, where the LSBs of all three colors can be seen as 3-bit symbols on which the cost functions are defined. Applications outside the scope of digital images are possible as long as we know how to define the costs.

The implicit premise of this chapter is the direct relationship between the distortion function D and statistical detectability. Designing (and possibly learning) the distortion measure for a given cover source is an interesting research problem by itself. Examples of distortion measures presented in this work are unlikely to be optimal and we include them here mainly to illustrate the concepts. The problem of designing the distortion function is covered in Chapter 7.

C++ implementation with Matlab wrappers of STCs and multi-layered STCs are available at `http://dde.binghamton.edu/download/syndrome/`.

Chapter 7

Design of Adaptive Embedding Schemes for Digital Images

The last chapter of the minimum-distortion framework is devoted to the problem of optimizing distortion functions so that they better correspond to statistical detectability as measured by blind feature-based steganalyzers. In practice, most distortion functions are obtained heuristically and do not generalize well to other cover sources. Here, we constrain ourselves to independent embedding changes and present practical tools that Alice can use for "learning" the embedding algorithm for a given cover source. The same technique is also applicable for the Gibbs construction. Since syndrome-trellis codes do not require Bob to have the distortion function for extraction, Alice can learn it according to her needs for a specific cover source.

Our motivation for solving the problem of the cost-function design comes from the HUGO algorithm [94] that assigns the costs of individual changes based on the pixel neighborhood. Unfortunately, this approach does not easily generalize to other cover sources, such as JPEG or color bitmap images, neither is it clear how to optimize the design. In this chapter, we open the question of the cost-function design and strive for a robust approach that generalizes well to unseen cover images and unseen steganalytic features to avoid overfitting to a particular cover source and feature space. For example, the Feature Correction Method [76], which is a heuristic approach to embed while approximately preserving the cover-image feature vector, is known to be overly sensitive to the chosen feature set and does not generalize or scale well. The work in [90] has an alternate feature preservation approach and also empirically considers the dynamics between steaganographer and steganlyzer.

The rest of this chapter is organized as follows. Section 7.1 casts the cost-design problem as a function optimization and introduces two new design criteria and a methodology for learning the costs from training images. The methodology developed in Section 7.1 is then applied to grayscale spatial-domain images in Section 7.2. Application to grayscale JPEG images is considered in Section 7.3. The chapter concludes in Section 7.4 with a discussion of possible future directions on how to apply and improve the proposed methodology for designing adaptive embedding schemes.

7.1 Empirical Design of Cost Functions

In this section, we focus on designing adaptive embedding schemes for the payload-limited sender subjected to sequential steganalysis. In this regime, the sender decides on the number of bits he wants to hide in a given cover object, embeds his payload, and sends the stego object through a passively monitored channel. In sequential steganalysis [65], the warden has to decide whether a given image is cover or stego solely based on a single object. We deliberately omit the possibility of intentionally spreading the payload into a group of cover images – a technique known as the batch steganography. This mode can improve the security of the scheme, however, it should no longer be tested with sequential steganalysis. The warden should use pooled steganalysis [65] that allows her to pool the results over a larger group of objects. We leave this direction open for a future research.

A common way of testing steganographic schemes is to report a chosen detection metric (ROC curve, accuracy, minimum error probability under equal priors P_{E}, etc.) empirically estimated from a database of cover and stego images where each stego image carries a fixed relative payload. Whenever possible, we report results obtained from cover images of roughly the same size to reduce the effect of the square root law [32].

Our goal is to design a set of functions ρ_i, $i \in \{1, \dots, n\}$, which, given the original cover image, assign the cost of changing individual cover elements to their new values. For digital images, the dependence between two cover pixels rapidly decreases with their distance. In case of grayscale spatial-domain digital images, the cost of changing a single pixel should mainly depend on its immediate neighborhood. For this reason, we constrain ρ_i to be a real-valued function Θ with small support, $\rho_i(\mathbf{x}, y_i) = \Theta(x_{\sigma(i)}, y_i)$, where $x_{\sigma(i)}$ denotes cover pixels spatially close to pixel i.

From practical experiments, it is possible to identify the quantity that should drive the costs. For example, pixels in busy regions can be changed more frequently (and by a larger amount) than those in smooth regions because they are generally harder to predict (model). On the other hand, pixels in saturated areas should not be modified at all. However, giving exact relationship between predictability of a pixel change given a small neighborhood, i.e., finding a good Θ is not an easy task. For simplicity, we allow Θ to depend on a vector-valued parameter $\theta \in \mathbb{R}^k$ and use our prior knowledge about the cover source to suitably parametrize Θ. With a real-valued measure of statistical detectability (such as the P_{E} error), the problem of finding the best ρ_i's is transformed to an optimization problem over the parameter space of θ – a problem which can be solved by numerical methods.

In the rest of this section, we review several detectability metrics and discuss their suitability for designing the cost function based on the dimensionality of θ. We will illustrate each optimization criterion on a simple problem of designing an adaptive embedding scheme for grayscale spatial-domain digital images with a single-parameter search space. All experiments described in this section were carried out with 10,800 512 × 512 grayscale images from the BOWS2 database [5] described in Section 7.2.

Inverse single-difference cost model: Let $\theta \geq 0$ and $\mathcal{N}_i = \{x_{i,\rightarrow}, x_{i,\nearrow}, x_{i,\uparrow}, \dots, x_{i,\searrow}\}$ be a set of eight pixels from the 3×3 neighborhood of the ith pixel. We use the ±1 embedding operation, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\} \cap \mathcal{I}$, and define

$$\rho_i(\mathbf{x}, y_i) = \Theta(\mathcal{N}_i, y_i) = \begin{cases} 0 & \text{if } y_i = x_i, \\ \infty & \text{if } y_i \notin \mathcal{I}_i, \\ \sum_{z \in \mathcal{N}_i} (1 + \theta|z - x_i|)^{-1} + (1 + \theta|z - y_i|)^{-1} & \text{otherwise.} \end{cases} \tag{7.1.1}$$

At the image boundary, the set of neighboring pixels $\mathcal{N}_i$ is reduced accordingly. This cost assignment penalizes changes in textured areas less than those in smooth regions depending on the differences between neighboring pixels.

7.1.1 Blind steganalysis

The only way of evaluating the security of steganographic schemes for empirical covers is to subject them to a steganalysis test. According to Kerckhoffs' principle, we allow the warden to know all elements of the stegosystem (the cover source statistics, the embedding algorithm and the size of the possible payload) except for the (possibly encrypted) message. Given a single image, the warden has to decide whether it is cover or stego. In this simple binary hypothesis test, the warden can make two types of errors – either detect the cover image as stego (false alarm) or recognize the stego image as cover (missed detection). The corresponding probabilities are denoted P_{FA} and P_{MD}, respectively. The relationship between these two errors is completely described by the ROC curve obtained by plotting $1 - P_{\mathrm{MD}}(P_{\mathrm{FA}})$ as a function of P_{FA}. Unfortunately, ROC curves cannot be directly used for evaluating steganalyzers (embedding algorithms) as they cannot be ordered (they may overlap). Thus, we reduce the ROC curve into a scalar detection measure P_{E} (see Section 2.3).

Due to the lack of exact probability distributions for real digital media, practical steganalyzers for such empirical cover sources are constructed by training a binary classifier on a set of cover and stego images obtained by embedding a pseudo-random message. We follow the blind feature-based approach described in Section 2.3 using soft-margin support-vector machines with Gaussian kernel for binary classification.

Even though blind steganalysis provides the most trustworthy measure of detectability in practice, it requires a large number of images for training and a separate set of images for testing. In practice, many thousands of images are usually processed by the embedding algorithm to create the stego images and extract the features. Since the training can also be very time consuming, evaluating detectability of a specific embedding algorithm at a given payload using machine learning can be prohibitively expensive. For this reason, only a small number of parameters θ can be evaluated and thus this method is impractical for optimizing a high dimensional θ. This complexity issue is the main motivation for developing alternative and much faster optimization criteria. We used the error P_{E} estimated using an SVM-based classifier mainly for validating the results obtained from other optimization criteria or for performing the grid search over a small region of the search space.

7.1.2 `L2R_L2LOSS` - soft-margin optimization criterion

Although there exist many algorithms for binary classification, SVMs are popular for their good ability to generalize to unseen data samples. The success of SVMs lies in the optimization criterion which, for the case of a linear classifier, looks for the separating hyperplane maximizing the distance (often called *margin*) between itself and the closest data points. Intuitively, the larger the margin between two classes, the better they can be separated and the smaller the P_{E} error becomes. We use the *size of the margin* for a linear SVM as the optimization criterion. It is described and studied below.

Let $\mathcal{C}$ be the set of N cover images and $\mathcal{S}$ the set of N stego images obtained from $\mathcal{C}$ by embedding a pseudo-random message into each image. By extracting a d-dimensional feature from each image, we obtain a set of $2N$ vectors $\{\mathbf{f}_i \in \mathbb{R}^d | i \in \{1,\dots,2N\}\}$. We also define the labels g_i, $i \in \{1,\dots,2N\}$, as $g_i = -1$ if $\mathbf{f}_i$ was obtained from a cover image and $g_i = +1$ otherwise. Furthermore, we normalize all cover feature vectors $\mathbf{f}_i$ so that the sample variance of each element is 1. This scaling is then applied to stego features as well. SVMs with a linear kernel [60] classify a new sample $\mathbf{f}$ as cover if $\mathbf{w}^T\mathbf{f} < 0$, where $\mathbf{w} \in \mathbb{R}^d$ is the normal vector of the decision hyperplane obtained by solving the optimization problem:

$$\min_{\mathbf{w}\in\mathbb{R}^d} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{2N}\xi(\mathbf{w};\mathbf{f}_i,g_i). \tag{7.1.2}$$

Here, $\xi(\mathbf{w};\mathbf{f}_i,g_i)$ is a loss function and $C > 0$ is a penalty parameter. By minimizing (7.1.2), we maximize the margin while penalizing the misclassified samples. We focus on the so-called L2-SVM penalty function $\xi(\mathbf{w};\mathbf{f}_i,g_i) = \max(1 - g_i\mathbf{w}^T\mathbf{f}_i, 0)^2$. The optimization problem (7.1.2) can also be formulated in its dual form [60]:

$$\begin{aligned} \min_{\boldsymbol{\alpha}\in\mathbb{R}^{2N}} \quad & h(\boldsymbol{\alpha}) = \frac{1}{2}\boldsymbol{\alpha}^T\bar{\mathbb{Q}}\boldsymbol{\alpha} - \sum_{i=1}^{2N}\alpha_i \\ \text{subject to} \quad & 0 \le \alpha_i, \forall i \in \{1,\dots,2N\}, \end{aligned} \tag{7.1.3}$$

where $\bar{\mathbb{Q}} = \mathbb{Q} + \mathbb{D}$, $\mathbb{D}$ being a diagonal matrix with $D_{ii} = (2C)^{-1}$, and $Q_{ij} = g_i g_j \mathbf{f}_i^T\mathbf{f}_j$, $i,j \in \{1,\dots,2N\}$. Given $\boldsymbol{\alpha}$, the solution to (7.1.2) is $\mathbf{w} = \sum_{i=1}^{2N} g_i\alpha_i\mathbf{f}_i$. From the duality, the value $-h(\boldsymbol{\alpha})$, for any $\boldsymbol{\alpha}$ with $\alpha_i \ge 0$, bounds the optimal solution to the primal problem from below. We call the optimal value of $h(\boldsymbol{\alpha})$ from (7.1.3), the `L2R_L2LOSS` (L_2-regularized L_2-loss) criterion. The smaller the value of this criterion, the larger the optimal value of (7.1.2) is, and the smaller the possible margin between cover and stego samples becomes. Therefore, steganographers should be interested in *minimizing* `L2R_L2LOSS`.

We used a dual coordinate descent method [60] with 10^4 iterations, $C = 0.1$, and $\epsilon = 0.1$ as implemented in the LIBLINEAR [23] package to calculate `L2R_L2LOSS`. Evaluating `L2R_L2LOSS` with second-order SPAM features took 1–2 seconds for $N = 80$ 512×512 cover images on a cluster of 40 CPUs when the message-embedding and feature-extraction parts were distributed using OpenMPI.

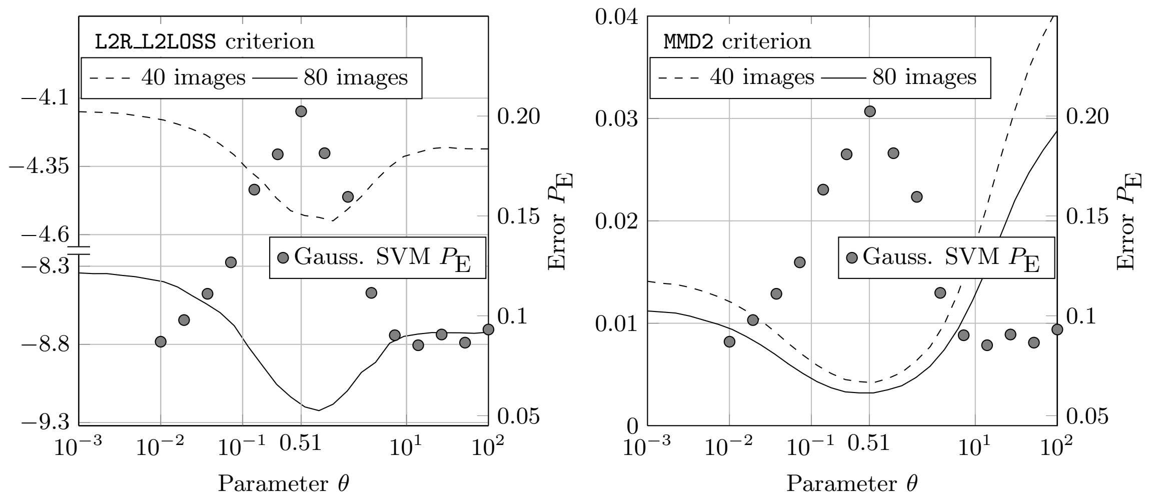

Figure 7.1.1: Comparison of different cost assignments in the inverse single-difference cost model (7.1.1) with a payload-limited sender embedding 0.5 bpp using the L2R_L2LOSS (left) and MMD2 (right) optimization criteria. The results are compared with the P_E error obtained from an SVM-based classifier. All results were produced using the CDF set and the BOWS2 database of 512×512 grayscale images.

When optimizing θ using L2R_L2LOSS, we fix the set of cover images $\mathcal{C}$ and the set of pseudo-random messages we will be embedding. We did this by fixing the seeds used for choosing the cover images and the seed used by the embedding simulator. Although L2R_L2LOSS may have different values when evaluated across different sets $\mathcal{C}$, the minimum w.r.t. θ stays approximately the same. Figure 7.1.1(left) shows the value of the L2R_L2LOSS criterion based on the CDF set when evaluated for different values of $\theta \geq 0$ in (7.1.1) and the number of images in $\mathcal{C}$. We can see that even with 40 images, the optimal value of θ is close to the value obtained from the SVM-based classifier.

Because the L2R_L2LOSS criterion can be evaluated quickly, it can be minimized using numerical methods even for a high dimensional θ. Unfortunately, for higher dimensional θ, the surface obtained by this criterion w.r.t. θ is not smooth enough for gradient-based optimization methods to be used efficiently. Instead, we used the Nelder–Mead simplex-reflection method (exactly as described in [89, Chapter 9.5]) with elements of the initial simplex generated uniformly at random in $[0, 1]$. Due to the non-smooth nature of the optimization criterion, we cannot guarantee that we reached a global minimum (in fact, the solution will be most likely a local minimum).

7.1.3 Other optimization criteria and their relevance to cost design

Due to the non-smooth optimization surface, we may be interested in other metrics. Metrics leading to a smooth optimization surface may produce an embedding algorithm whose cost assignments may be easier to interpret. Here, we present one such metric – the Maximum Mean Discrepancy (MMD) [54, 96]. MMD has been used for comparison of steganographic methods [96] and other machine learning problems, such as feature selection [49]. Originally, MMD was designed as a statistical test for the two-sample problem – to decide whether two data sets were obtained from the same distribution. The theoretical derivation of MMD appears in [96]. Here, we only review the connection between MMD and binary hypothesis testing.

Let $\mathcal{C}'$ and $\mathcal{S}'$ be the sets of N' cover and stego images, respectively. We require the set of cover images used for creating $\mathcal{S}'$ to be disjoint with $\mathcal{C}'$. Let $\mathbf{c}_i, \mathbf{s}_i \in \mathbb{R}^d$, $i \in \{1, \ldots, N'\}$, be the feature vectors representing the ith cover and stego image, respectively. As in Section 7.1.2, we normalize

$\mathbf{c}_i$ and $\mathbf{s}_i$ to unit variance obtained from the cover features. An unbiased estimate of MMD^2 is

$$\mathrm{MMD}(\mathcal{C}', \mathcal{S}')^2 = \frac{1}{N'(N'-1)} \sum_{i \neq j} k_\lambda(\mathbf{c}_i, \mathbf{c}_j) - k_\lambda(\mathbf{c}_i, \mathbf{s}_j) + k_\lambda(\mathbf{s}_i, \mathbf{s}_j) - k_\lambda(\mathbf{s}_i, \mathbf{c}_j), \tag{7.1.4}$$

where $k_\lambda(\mathbf{c}, \mathbf{s}) = \exp(-\gamma \left\| \mathbf{c} - \mathbf{s} \right\|_2^2)$ is the Gaussian kernel with parameter $\gamma \geq 0$. We set the width of the Gaussian kernel to $\lambda = 10^{-3}$, which closely corresponds to the "median rule" [54]. In practice, we used the set of $N \geq 2N'$ cover images from which $\mathcal{C}'$ and $\mathcal{S}'$ were derived using a pseudo-random permutation. For a given set of N cover images, we define the `MMD2` criterion as the sample mean of $\mathrm{MMD}(\mathcal{C}', \mathcal{S}')^2$ calculated over M pseudo-random partitions. For the 1234-dimensional CDF set, evaluating `MMD2` using $N = 80$ 512×512 cover images with $N' = 40$ and $M = 10^5$ took 4 seconds on a 40-CPU computer cluster when all operations were parallelized using OpenMPI.

The `MMD2` criterion is related to binary classification using Parzen windows [57, Chapt. 6.6]. A simple binary hypothesis testing problem (deciding whether a given image is cover or stego) can be solved optimally using the Likelihood Ratio Test (LRT) once the exact probability distributions of cover, P_C, and stego feature vectors, P_S, are available. Given an unknown feature vector $\mathbf{f}$, the LRT calls $\mathbf{f}$ cover if $P_C(\mathbf{f}) > P_S(\mathbf{f})$ and stego otherwise. Because neither P_C or P_S are available, one may want to estimate them from a set of N cover and N stego training samples $\mathbf{f}_i \in \mathbb{R}^d$ with labels g_i, $i \in \{1, \ldots, 2N\}$. The Parzen estimate of $P_C(\mathbf{f})$ defined as

$$\hat{P}_C(\mathbf{f}) = \frac{1}{N} \sum_{g_i = -1} K_\lambda(\mathbf{f}_i, \mathbf{f}) \tag{7.1.5}$$

"counts" the number of training vectors that are close to $\mathbf{f}$. Here, $K_\lambda(\mathbf{f}_i, \mathbf{f})$ is a kernel giving larger weights to vectors closer to $\mathbf{f}$. A popular choice for K_λ is the Gaussian kernel $K_\lambda(\mathbf{f}_i, \mathbf{f}) = k_\lambda(\mathbf{f}_i, \mathbf{f}) = \exp(-\gamma \left\| \mathbf{f}_i - \mathbf{f} \right\|_2^2)$. The Parzen estimate of $P_S(\mathbf{f})$, denoted $\hat{P}_S(\mathbf{f})$, is defined in a similar way. When we substitute $\hat{P}_C(\mathbf{f})$ and $\hat{P}_S(\mathbf{f})$ into the LRT, we obtain the Parzen window classifier. Therefore, $\mathrm{MMD}(\mathcal{C}', \mathcal{S}')^2$ calculates a finite-sample estimate of the average detection criterion with equal-priors:

$$\mathrm{MMD}(P_C, P_S)^2 = E_{\mathbf{f}, \mathbf{f}_{-1} \sim P_C, \mathbf{f}_{+1} \sim P_S} \left[k_\lambda(\mathbf{f}, \mathbf{f}_{-1}) - k_\lambda(\mathbf{f}, \mathbf{f}_{+1}) \right] + E_{\mathbf{f}_{-1} \sim P_C, \mathbf{f}, \mathbf{f}_{+1} \sim P_S} \left[k_\lambda(\mathbf{f}, \mathbf{f}_{+1}) - k_\lambda(\mathbf{f}, \mathbf{f}_{-1}) \right] \tag{7.1.6}$$

obtained using the leave-one-out cross-validation [57, Chapt. 7.10]. Due to the Gaussian kernel k_λ, $\mathrm{MMD}(P_C, P_S)^2 \geq 0$ and $\mathrm{MMD}(P_C, P_S)^2 = 0$ if and only if $P_C = P_S$. For this reason, the steganographer should *minimize* the `MMD2` criterion, which is a bootstrapped version of (7.1.4).

Figure 7.1.1 (right) compares the `MMD2` criterion when calculated from $N = 80$ and $N = 40$ cover images using $N' = N/2$ and $M = 10^5$ over different values of $\theta \geq 0$. The results obtained from the SVM-based classifier are plotted for reference. Due to bootstrapping, the `MMD2` criterion results in a smooth optimization surface even for a high-dimensional θ. We used a simple gradient descent-based optimization technique to minimize `MMD2`.

7.2 Application to Spatial-Domain Digital Images

In this section, we apply the proposed optimization criteria to the problem of optimizing the cost models for grayscale spatial-domain digital images. We first compare the `L2R_L2LOSS` and the `MMD2` criteria on a high-dimensional cost model and validate the results using an SVM-based steganalyzer. `L2R_L2LOSS` is then used for optimizing models similar in nature to those used in the HUGO algorithm [94].

We use the BOWS2 image database [5] containing approximately 10800 grayscale images of size 512 × 512. Images in this database were obtained by rescaling high-resolution photographs of different scenes originally stored as JPEGs and then converted to grayscale. The database was not processed to remove images containing areas with saturated pixels. For comparison, we also use the BOSSBase[1] image database with 9074 grayscale images originally taken by seven different camera models in a RAW format (CR2 or DNG) and converted/resized to grayscale images of size 512 × 512. This database was intentionally formed to not contain images with large regions of saturated pixels.

[1] The latest version of the image database used in the BOSS contest http://boss.gipsa-lab.grenoble-inp.fr/.

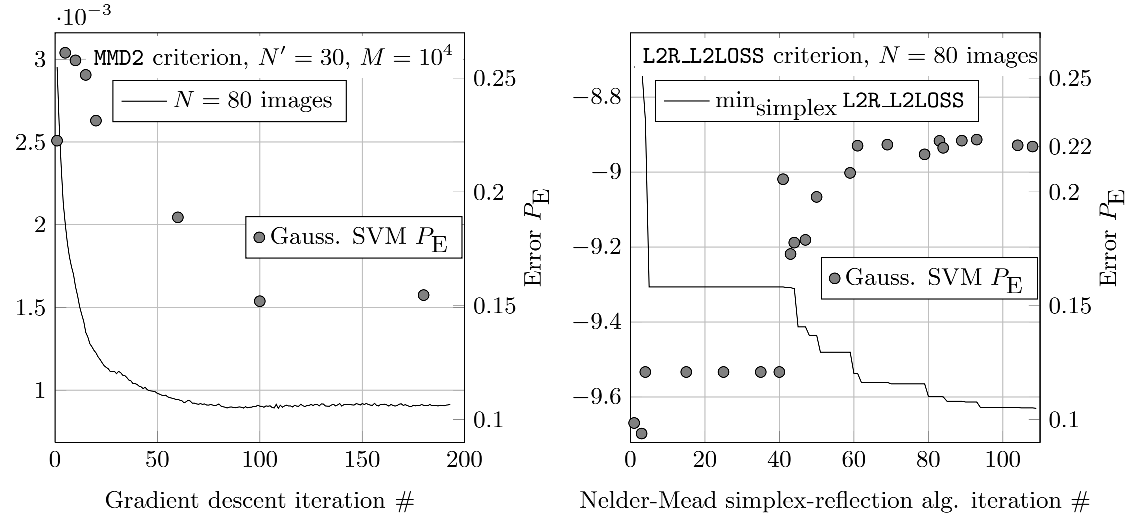

Figure 7.2.1: The value of the optimization criteria MMD2 (left) and L2R_L2LOSS (right) when optimized by their respective algorithms using the generalized single-difference cost model (7.2.1) embedding 0.5 bpp. Selected cost assignments are validated with the P_E error obtained from the SVM-based classifier. All results were produced using the CDF set and the BOWS2 database of 512×512 grayscale images. These results are explained in Section 7.2.1.

7.2.1 Comparing the L2R_L2LOSS and MMD2 criteria for high-dimensional search space

In the single-difference cost model (7.1.1), the cost of changing the ith pixel was forced to follow the inverse model driven by the scalar parameter θ. We now generalize this and associate one parameter with each value of a pixel difference.

Generalized single-difference cost model: Since most pixel differences are concentrated around zero, we define $\boldsymbol{\theta} = (\theta_{-\Delta}, \theta_{-\Delta+1}, \ldots, \theta_{\Delta-1}, \theta_{\Delta}, \theta_{\bullet}) \in \mathbb{R}^{2\Delta+2}$ to be a $2\Delta+2$-dimensional vector, for some fixed parameter $\Delta \in \mathbb{N}$. Again, let $\mathcal{N}_i = \{x_{i,\rightarrow}, x_{i,\nearrow}, x_{i,\uparrow}, \ldots, x_{i,\searrow}\}$ be a set of eight pixels in the 3×3 neighborhood of the ith pixel. Given $\boldsymbol{\theta}$, the cost of changing the ith pixel by ± 1, $\mathcal{I}_i = \{x_i - 1, x_i, x_i + 1\} \cap \mathcal{I}$, is

$$\rho_i(\mathbf{x}, y_i) = \Theta(\mathcal{N}_i, y_i) = \begin{cases} 0 & \text{if } y_i = x_i, \\ \infty & \text{if } y_i \notin \mathcal{I}_i, \\ \sum_{z \in \mathcal{N}_i} \theta_{z-x_i}^2 + \theta_{z-y_i}^2 & \text{otherwise}, \end{cases} \tag{7.2.1}$$

where $\theta_j = \theta_\bullet$ when $|j| > \Delta$. We require $\rho_i(\mathbf{x}, y_i) \geq 0$ and enforce this by squaring. Allowing $\rho_i(\mathbf{x}, y_i) < \rho_i(\mathbf{x}, x_i)$ would lead to cases where it is actually beneficial to make the change instead of keeping the original value. We do not consider such a case here.

Figure 7.2.1 shows the progress of optimizing the generalized single-difference cost model (7.2.1) using the MMD2 (left) and L2R_L2LOSS (right) criteria when embedding a fixed relative payload of 0.5 bpp. We used a simple gradient-descent and the Nelder–Mead simplex-reflection algorithms utilizing the CDF set to minimize MMD2 and L2R_L2LOSS over a fixed set of 80 images, respectively. Selected values of the parameter $\boldsymbol{\theta}$ were also tested using a Gaussian SVM-based steganalyzer utilizing the CDF set. For the final solution, the L2R_L2LOSS criterion provides a more secure embedding algorithm (a higher P_E error) than those obtained from MMD2. As can be seen from the left figure, optimizing the cost assignments w.r.t. the MMD2 criterion does not lead to increasing the P_E error of the SVM-based steganalyzer. Although the final solution obtained from the L2R_L2LOSS criterion does not achieve the best known result (see the leftmost point achieving $P_\mathrm{E} = 26\%$ in the left graph), we consider it to be better connected to the P_E error and use it for all experiments in this chapter.

The discrepancy between the P_{E} error and the MMD2 criterion may be due to the strong relationship between MMD2 and the non-parametric Parzen window classifier, which is believed to be worse than a Gaussian SVM-based steganalyzer. The fact that L2R_L2LOSS does not achieve the maximal known P_{E} is because solution was a local minimum. Restarting the Nelder–Mead algorithm with a different initial simplex lead to different solutions achieving different L2R_L2LOSS values. The gap between the current and optimal solution may be closed in the future using other optimizing criteria or more involved optimization methods.

7.2.2 Cost models based on pixel differences

We further generalize the single-difference cost model by allowing the cost to depend on a larger neighborhood via two or three pixel differences. For better clarity, we represent the cover image $\mathbf{x}$ in a matrix form, where $x_{i,j} \in \mathcal{I}$ denotes the pixel in ith row and jth column.
Two-difference cost model: Let $\mathcal{D}^{\rightarrow}_{i,j}(z) = \{(x_{i,j-2}-x_{i,j-1}, x_{i,j-1}-z), (x_{i,j-1}-z, z-x_{i,j+1}), (z-x_{i,j+1}, x_{i,j+1}-x_{i,j+2})\}$ be a set of two-element vectors describing the differences around the i,jth pixel in the horizontal direction when $x_{i,j}$ is replaced by $z \in \mathcal{I}$. We define $\mathcal{D}_{i,j}(z) = \mathcal{D}^{\rightarrow}_{i,j}(z) \cup \mathcal{D}^{\nearrow}_{i,j}(z) \cup \mathcal{D}^{\uparrow}_{i,j}(z) \cup \mathcal{D}^{\nwarrow}_{i,j}(z)$, where the last three sets are defined similarly as $\mathcal{D}^{\rightarrow}_{i,j}(z)$ except with a different orientation. The cost model is described by $\boldsymbol{\theta} \in \mathbb{R}^{(2\Delta+1)^2+1}$ consisting of $\theta_{k,l} \in \mathbb{R}$ for $-\Delta \leq k, l \leq \Delta$ (this models the cost of disturbing the difference vector (k,l)) and $\theta_{\bullet} \in \mathbb{R}$ for all other values outside Δ. Given $\boldsymbol{\theta}$, the cost of changing the i,jth pixel by ±1, $\mathcal{I}_{i,j} = \{x_{i,j}-1, x_{i,j}, x_{i,j}+1\} \cap \mathcal{I}$, is

$$\rho_{i,j}(\mathbf{x}, y) = \Theta(y) = \begin{cases} 0 & \text{if } y = x_{i,j}, \\ \infty & \text{if } y \notin \mathcal{I}_{i,j}, \\ \sum_{\mathbf{d} \in \mathcal{D}_{i,j}(x_{i,j})} \theta^2_{\mathbf{d}} + \sum_{\mathbf{d} \in \mathcal{D}_{i,j}(y)} \theta^2_{\mathbf{d}} & \text{otherwise}, \end{cases} \tag{7.2.2}$$

where $\theta_{\mathbf{d}} = \theta_{\bullet}$ whenever any element of $\mathbf{d} \in \mathbb{N}^2$ is larger than Δ. We reduce the sum in (7.2.2) accordingly when the i,jth pixel is close to the image boundary.
Three-difference cost model: We extend $\mathcal{D}^{\rightarrow}_{i,j}(z)$ to include all three-element vectors one may obtain from four pixels in the horizontal direction containing $x_{i,j}$, i.e., $|\mathcal{D}^{\rightarrow}_{i,j}(z)| = 4$ and define a $(2\Delta+1)^3+1$-dimensional cost model in the same fashion as above.

Figure 7.2.2 compares the performance of algorithms based on two and three-difference cost models with $\Delta = 4$ optimized using the L2R_L2LOSS criterion for payloads $\alpha' = 0.2$ and $\alpha' = 0.5$ bpp. Both algorithms were simulated on their respective rate–distortion bounds. The performance of a practical implementation of the scheme for $\alpha' = 0.5$ is rather close to the simulated scheme when implemented using the multi-layered STCs [29]. The costs were minimized using the second-order SPAM features with $T = 3$ and tested with a Gaussian SVM-based steganalyzer with the CDF set. This shows the ability of the optimization procedure to produce cost assignments that are not overtrained to a specific feature set despite the fact that the dimensionality of the search space for the three-difference cost model was $(2\Delta+1)^3+1 = 730$. As can be seen from the figure, the algorithm designed for $\alpha' = 0.5$ bpp achieved better results for larger payloads. Increasing the design payload above 0.5 bpp did not bring any further improvement. All algorithms achieve better performance than HUGO [94], because they better utilize the ternary embedding operation for large payloads.

7.3 Application to Digital Images in DCT Domain

Most adaptive embedding schemes for JPEG images [29, 74, 102] embed message bits while quantizing the DCT coefficients during JPEG compression and minimize an additive distortion function (5.4.1) derived from the rounding errors. This approach utilizes the side-information in the form of a never-compressed image, which may not always be available. In this section, we focus on designing adaptive embedding schemes that start directly from a JPEG image and derive the costs of changing a single DCT coefficient from its neighborhood.

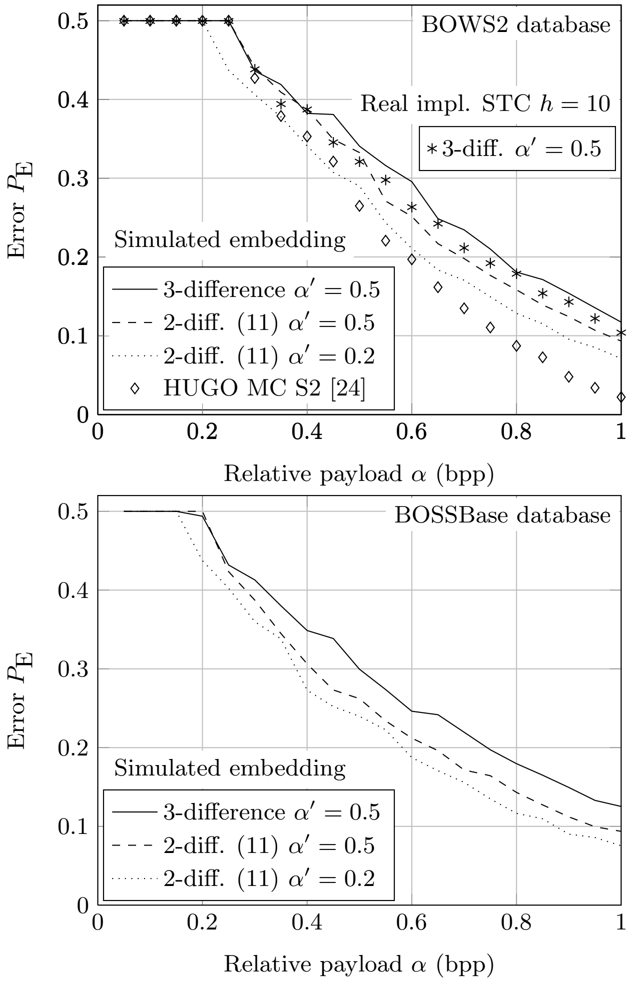

Figure 7.2.2: Performance of embedding algorithms optimized using the `L2R_L2LOSS` criterion with second-order SPAM features with $T = 3$, payload α' bpp, and 80 random images from the BOWS2 database. All algorithms were tested using a Gaussian SVM-based steganalyzer utilizing the CDF set with training and testing images from BOWS2 (left) and BOSSBase (right). Results from the HUGO algorithm [94] when simulated on the rate–distortion bound are shown for comparison.

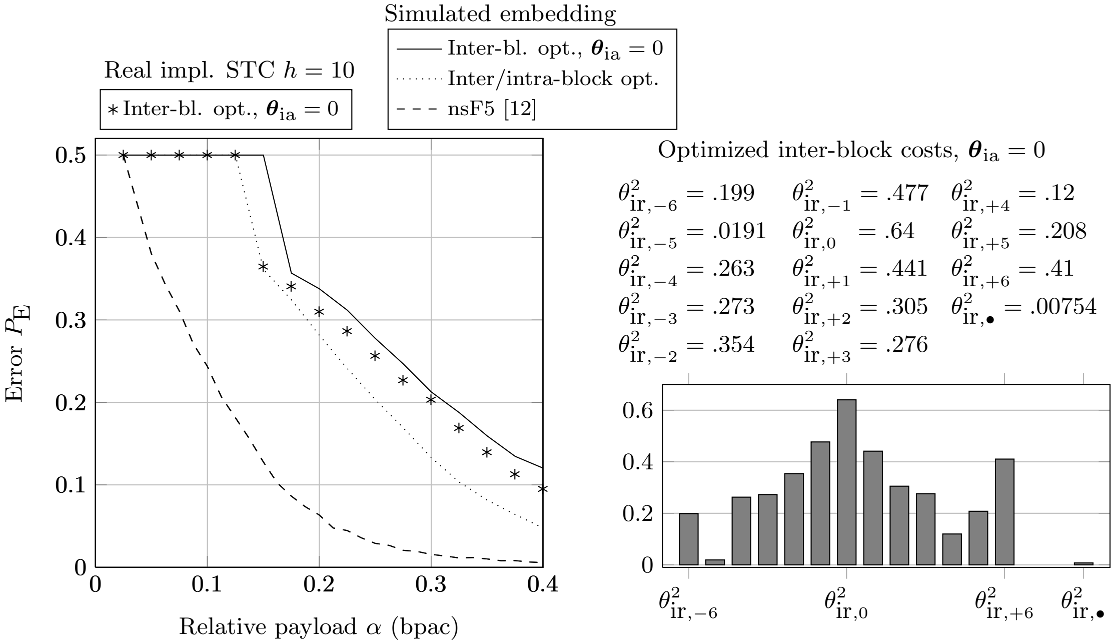

Figure 7.3.1: (Left) Detectability of embedding algorithms for the DCT domain based on the inter/intra-block cost model (7.3.1) optimized using the `L2R_L2LOSS` criterion and CC-PEV features for the payload of 0.5 bpac. The error P_{E} was measured using a Gaussian SVM-based steganalyzer with the CDF set. (Right) The values of $\boldsymbol{\theta}_{\mathrm{ir}}$ for the optimized inter-block model used to generate the plot on the left.

We used a mother database of $6,500$ images obtained from 22 different cameras at their full resolution in a raw format from which a database of $6,500$ grayscale JPEG cover images was created. Each raw image was first converted to grayscale, resized to a smaller size of 512 pixels using bilinear interpolation while preserving the aspect ratio, and finally JPEG compressed using quality factor 75.

A common way of expressing the payload in DCT-domain steganography is the number of bits embedded per non-zero AC DCT coefficient [47], which we denote as "bpac." This is because essentially all embedding schemes for DCT domain never change zero coefficients and some even avoid changing DC coefficients due to their high impact on statistical detectability. According to [47], the most secure algorithm that does not rely on any side-information is the nsF5, which minimizes the number of changed non-zero AC DCT coefficients. Using our terminology, the nsF5 uses a binary embedding operation that decreases the absolute value of a non-zero AC DCT coefficient, i.e., $\mathcal{I}_i = \{x_i, x_i - sign(x_i)\}$ whenever $x_i \neq 0$ is an AC coefficient, and $\mathcal{I}_i = \{x_i\}$ otherwise. Figure 7.3.1 shows the performance of nsF5 when simulated as described in Section 6.1.1. The detection was implemented using the CDF set with a Gaussian SVM-based steganalyzer.

Similar to the spatial domain, we design the costs based on the differences between DCT coefficients either from neighboring blocks or from similar DCT modes in the same 8×8 block. This allows us to express the context in which a single change is made. We represent a JPEG image $\mathbf{x}$ in a matrix notation, where $x_{i,j} \in \mathcal{I} \triangleq \{-1024, \ldots, 1024\}$ denotes the DCT element of mode $(i \bmod 8, j \bmod 8)$ in the $\lceil i/8 \rceil, \lceil j/8 \rceil$th block. The set $\{x_{i,j} | i \bmod 8 \neq 0 \vee j \bmod 8 \neq 0\}$ describes all AC DCT coefficients in $\mathbf{x}$. We define the following cost model, which we use with a ternary embedding operation.

Inter/intra-block cost model: Let $\boldsymbol{\theta} = (\boldsymbol{\theta}_{\mathrm{ir}}, \boldsymbol{\theta}_{\mathrm{ia}}) \in \mathbb{R}^{(2\Delta+1)+1} \times \mathbb{R}^{(2\Delta+1)+1}$ be the model parameters describing the cost of disturbing inter- and intra-block dependencies with $\boldsymbol{\theta}_{\mathrm{ir}} = (\theta_{\mathrm{ir},-\Delta}, \ldots, \theta_{\mathrm{ir},\Delta}, \theta_{\mathrm{ir},\bullet})$ and $\boldsymbol{\theta}_{\mathrm{ia}} = (\theta_{\mathrm{ia},-\Delta}, \ldots, \theta_{\mathrm{ia},\Delta}, \theta_{\mathrm{ia},\bullet})$. The cost of changing *any* (even zero) AC DCT coefficient

$x_{i,j}$ to $y \in \mathcal{I}_{i,j} \triangleq \{x_{i,j}-1, x_{i,j}, x_{i,j}+1\} \cap \mathcal{I}$ is

$$\rho_{i,j}(\mathbf{x}, y) = \Theta(y) = \begin{cases} 0 & \text{if } y = x_{i,j}, \\ \infty & \text{if } y \notin \mathcal{I}_{i,j}, \\ \sum_{z\in\mathcal{N}_{\mathrm{ia}}} \theta^2_{\mathrm{ia},x_{i,j}-z} + \sum_{z\in\mathcal{N}_{\mathrm{ir}}} \theta^2_{\mathrm{ir},x_{i,j}-z} & \text{otherwise,} \end{cases} \tag{7.3.1}$$

where $\mathcal{N}_{\mathrm{ir}} = \{x_{i+8,j}, x_{i,j+8}, x_{i-8,j}, x_{i,j-8}\}$ and $\mathcal{N}_{\mathrm{ia}} = \{x_{i+1,j}, x_{i,j+1}, x_{i-1,j}, x_{i,j-1}\}$ are inter- and intra-block neighborhoods, respectively. As before, $\theta_{\mathrm{ia},z} = \theta_{\mathrm{ia},\bullet}$ and $\theta_{\mathrm{ir},z} = \theta_{\mathrm{ir},\bullet}$ whenever $|z| > \Delta$. We reduced the sum in (7.3.1) accordingly when the required element falled outside of the image boundary.

Figure 7.3.1 (left) compares the performance of embedding algorithms based on the above inter/intra-block cost model when optimized using the `L2R_L2LOSS` criterion with CC-PEV features and payload 0.5 bpac. We report the performance of two algorithms for $\Delta = 6$. In the first version, both $\boldsymbol{\theta}_{\mathrm{ir}}$ and $\boldsymbol{\theta}_{\mathrm{ia}}$ were optimized, while in the second version only the inter-block part $\boldsymbol{\theta}_{\mathrm{ir}}$ was optimized while $\boldsymbol{\theta}_{\mathrm{ia}} = (0, \ldots, 0)$. To show that the optimized algorithms are not over-trained to the CC-PEV features calibrated by cropping by 4×4 pixels, we report the P_{E} error obtained from a Gaussian SVM-based steganalyzer utilizing the CDF set. Similar performance results were obtained using the CC-PEV feature set with calibration by cropping by 2×4 pixels, which suggests that the algorithms are not over-trained to a specific feature set. Unfortunately, the algorithm optimized w.r.t. both inter- and intra-block parts did not achieve a better performance than the algorithm with $\boldsymbol{\theta}_{\mathrm{ia}} = 0$, which is just a special case. This is due to the fact that the Nelder–Mead algorithm converged to a local minimum (the `L2R_L2LOSS` criterion was smaller for the case with $\boldsymbol{\theta}_{\mathrm{ia}} = 0$). When compared with the non-adaptive nsF5 algorithm, both versions increased the payload for the same level of security more than twice. All algorithms can be implemented using the multi-layered STCs [29] in practice. Figure 7.3.1 shows that the loss introduced by such a practical implementation is small when implemented using STCs with constraint height $h = 10$.

We found out experimentally that it is more effective to optimize the cost functions w.r.t. larger payloads. Methods optimized for smaller payloads, such as 0.1 bpac, did not achieve as high performance for higher payloads as methods optimized for larger payloads.

7.4 Conclusion

The basic premise behind steganography designed to embed while minimizing a certain distortion function is that the distortion is related to statistical detectability. In the past, steganographers used heuristically defined distortion functions and focused on the problem of embedding with minimal distortion while no attempt was made to justify the choice of the distortion function or optimize its design. Since the problem of embedding with minimal distortion has been resolved in a near-optimal fashion in Chapters 5 and 6, what remains to be done and where the biggest gain in steganographic security lies is the form of the distortion function.

The main contribution of this chapter is a practical methodology using which one can optimize the distortion to design steganographic schemes with improved security. We do so by representing images in a feature space in which we define a criterion evaluating the separability between the sets of cover and stego features. The distortion function is parametrized and the parameters are found by optimizing them w.r.t. the chosen criterion on a set that is relatively small – 80 cover and stego images. The result is validated on various cover sources using blind steganalyzers. We intentionally use steganalyzers that utilize different feature spaces than the one in which we optimize to demonstrate that our optimized design generalizes to other feature sets as well cover sources.

We work with additive distortion functions that can be written as a sum of costs defined for each pixel, while each pixel cost depends on neighboring cover pixels. After investigating three different choices for the criterion, we selected the margin of a linear SVM as the most suitable one that is computationally efficient yet still closely tied to detectability as determined by a binary classifier trained on a large set of images.

The merit of the proposed work is demonstrated by incorporating the optimized cost for the ±1 embedding operation in the spatial domain and the ±1 operation for the DCT domain. The improvement over current state of the art is especially apparent in the DCT domain where the methods with optimized costs can embed more than twice as large payloads for the same detectability as the nsF5 algorithm. The costs are robust in the sense that the improvement can be observed even when the new method is tested with steganalyzers using a different feature set and even on a slightly different cover source.

Without any doubts, better parametric models for the distortion in the DCT domain can and should be considered. For example, the cost parameters should be dependent on the spatial frequency of DCT coefficients. This would substantially increase the dimensionality of the parameter space which would need to be balanced out by a corresponding increase of the number images. This appears to be a mere issue of increased complexity rather than one that would render our approach inapplicable and we might consider it in our future work. Embedding simulators used in this chapter can be downloaded from http://dde.binghamton.edu/download/stego_design/.

Appendix A

The SRL for Markov Cover Sources - Proofs

In this appendix, we include two auxiliary lemmas needed in the proof of the SRL theorem for Markov cover sources in Section 3.5. In the whole chapter, we assume Assumptions 1–3 to hold. Each lemma is placed in separate section along with other results and derivation required to prove it.

We utilize the Iverson bracket $[S]$, defined as $[S] = 1$ when the statement S is true and zero otherwise and use $[\mathbf{x}_1^2 = (i,j)]$ as a shorthand for $[(x_1 = i) \wedge (x_2 = j)]$. Iverson bracket should not be confused with expectation or variance operators, $E[Z]$, $Var[Z]$ applied on a random variable Z.

A.1 Bound on the Variance of the Test Statistic $\nu_{\beta,n}$

Lemma A.1. *Let $\nu_{\beta,n}$ be the random variable defined in (3.5.1) for a fixed value of the parameter β and number of cover elements n. The variance of this random variable can be bounded by a constant C for every value of β and n*

$$\exists C, \forall\beta, \forall n \quad Var[\nu_{\beta,n}] \le C.$$

Proof. From the definition of $\nu_{\beta,n}$

$$\begin{aligned}\frac{(n-1)^2}{n} Var[\nu_{\beta,n}] &= E\left[\left(\sum_{k=1}^{n-1}[\mathbf{Y}_k^{k+1} = (i,j)]\right)^2\right] - E\left[\sum_{k=1}^{n-1}[\mathbf{Y}_k^{k+1} = (i,j)]\right]^2 \\ &\le \sum_{k=1}^{n-1} Var\Big[[\mathbf{Y}_k^{k+1} = (i,j)]\Big] + 2\Bigg\{\sum_{k+1<\hat{k}} E\Big[[\mathbf{Y}_k^{k+1} = (i,j)][\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)]\Big] - \\ &\quad - E\Big[[\mathbf{Y}_k^{k+1} = (i,j)]\Big]E\Big[[\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)]\Big]\Bigg\} + 2n. \qquad \text{(A.1.1)}\end{aligned}$$

In the last sum, we bounded all terms for $k = \hat{k} - 1$ by 1 and thus obtained the term $2n$ in the last inequality. The variance of $[\mathbf{Y}_k^{k+1} = (i,j)]$ can be calculated as

$$\begin{aligned}Var\Big[[\mathbf{Y}_k^{k+1} = (i,j)]\Big] &= E\Big[[\mathbf{Y}_k^{k+1} = (i,j)]\Big] - E\Big[[\mathbf{Y}_k^{k+1} = (i,j)]\Big]^2 \\ &= Q_\beta\big(\mathbf{Y}_k^{k+1} = (i,j)\big)\Big(1 - Q_\beta\big(\mathbf{Y}_k^{k+1} = (i,j)\big)\Big) \le \frac{1}{4},\end{aligned}$$

because $\big([\mathbf{Y}_k^{k+1} = (i,j)]\big)^2 = [\mathbf{Y}_k^{k+1} = (i,j)]$, and $x(1-x) \le \frac{1}{4}$ for all $x > 0$. Due to the stationarity of the MC, $Q_\beta(\mathbf{Y}_k^{k+1} = (i,j))$ does not depend on index k for all β.

Finally, we find an upper bound on the sum in (A.1.1) in the form of $C_2 n$ for some positive constant C_2. This will give us the proof because $Var[\nu_{\beta,n}] \leq \frac{n}{(n-1)^2}((n-1)\frac{1}{4} + 2C_2 n + 2n) \leq 4(\frac{1}{4} + 2C_2 + 2)$, and $\frac{n^2}{(n-1)^2} \leq 4$ for $n \geq 2$. Thus $C = 8C_2 + 9$.

We start by showing that

$$Q_\beta\Big(\mathbf{Y}_k^{k+1} = (i,j), \mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)\Big) - Q_\beta\Big(\mathbf{Y}_k^{k+1} = (i,j)\Big) Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)\Big)$$
$$= \Big\{\underbrace{Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j) \big| \mathbf{Y}_k^{k+1} = (i,j)\Big) - Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)\Big)}_{\leq N^2 \rho^{\hat{k}-k-2}}\Big\} Q_\beta\Big(\mathbf{Y}_k^{k+1} = (i,j)\Big) \leq N^2 \rho^{\hat{k}-k-2}, \tag{A.1.2}$$

for some $0 \leq \rho < 1$ and $k+1 < \hat{k}$ (N is the number of all possible states of the MC). In other words, the HMC is exponentially forgetting its initial condition. Then, we will be able to bound the sum in (A.1.1) by $N^2 \sum_{\hat{k}=3}^{n} \sum_{k=1}^{\hat{k}-2} \rho^{\hat{k}-k-2} = N^2 \sum_{\hat{k}=3}^{n} \frac{1-\rho^{\hat{k}-2}}{1-\rho} \leq N^2 \sum_{\hat{k}=3}^{n} \frac{1}{1-\rho} = N^2(n-2)\frac{1}{1-\rho} \leq \frac{N^2 n}{1-\rho}$. Thus, $C_2 = \frac{N^2}{1-\rho}$ because $Q_\beta\big(\mathbf{Y}_k^{k+1} = (i,j)\big) \leq 1$.

The term $Q_\beta\big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)\big)$ in (A.1.2) can be written as

$$\begin{aligned} Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j)\Big) &= \sum_{(\hat{i},\hat{j})} Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j) \big| \mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j})\Big) P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j})\Big) \\ &= \sum_{(\hat{i},\hat{j})} b_{\hat{i},i} b_{\hat{j},j} P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j})\Big). \end{aligned} \tag{A.1.3}$$

The term $Q_\beta\big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j) \big| \mathbf{Y}_k^{k+1} = (i,j)\big)$ in (A.1.2) can be written as

$$\begin{aligned} Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j) \big| \mathbf{Y}_k^{k+1} = (i,j)\Big) &= \frac{Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1} = (i,j), \mathbf{Y}_k^{k+1} = (i,j)\Big)}{Q_\beta\Big(\mathbf{Y}_k^{k+1} = (i,j)\Big)} \\ &= \frac{\sum_{(\hat{i},\hat{j})} \sum_{(\tilde{i},\tilde{j})} b_{\hat{i},i} b_{\hat{j},j} b_{\tilde{i},i} b_{\tilde{j},j} P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}), \mathbf{X}_k^{k+1} = (\tilde{i},\tilde{j})\Big)}{Q_\beta\Big(\mathbf{Y}_k^{k+1} = (i,j)\Big)} = (\#). \end{aligned}$$

Finally, $P\big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}), \mathbf{X}_k^{k+1} = (\tilde{i},\tilde{j})\big)$ can be factorized as $P\big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}) | \mathbf{X}_k^{k+1} = (\tilde{i},\tilde{j})\big) P\big(\mathbf{X}_k^{k+1} = (\tilde{i},\tilde{j})\big)$. Due to the Markov property of the random variable $\mathbf{X}_1^n$, $P\big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}) | \mathbf{X}_k^{k+1} = (\tilde{i},\tilde{j})\big) = P\big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}) | X_{k+1} = \tilde{j}\big)$. For each pair of indices $(\hat{i},\hat{j})$, we define index $\tilde{j}_{max} = \arg\max_{\tilde{j}} P\big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}) | X_{k+1} = \tilde{j}\big)$. Then

$$\begin{aligned} (\#) &\leq \frac{\sum_{(\hat{i},\hat{j})} b_{\hat{i},i} b_{\hat{j},j} P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}) \big| X_{k+1} = \tilde{j}_{max}\Big) \sum_{(\tilde{i},\tilde{j})} b_{\tilde{i},i} b_{\tilde{j},j} P\Big(\mathbf{X}_k^{k+1} = (\tilde{i},\tilde{j})\Big)}{Q_\beta\Big(\mathbf{Y}_k^{k+1} = (i,j)\Big)} \\ &\overset{(A.1.3)}{=} \sum_{(\hat{i},\hat{j})} b_{\hat{i},i} b_{\hat{j},j} P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1} = (\hat{i},\hat{j}) \big| X_{k+1} = \tilde{j}_{max}\Big). \end{aligned} \tag{A.1.4}$$

Now, we can combine (A.1.3) and (A.1.4) to prove (A.1.2) as

$$\begin{aligned} &Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1}=(i,j)\Big|\mathbf{Y}_{k}^{k+1}=(i,j)\Big)-Q_\beta\Big(\mathbf{Y}_{\hat{k}}^{\hat{k}+1}=(i,j)\Big) \\ &\overset{(A.1.3),(A.1.4)}{\leq} \sum_{(\hat{i},\hat{j})} b_{\hat{i},i} b_{\hat{j},j}\Big\{P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1}=(\hat{i},\hat{j})\Big|X_{k+1}=\tilde{j}_{max}\Big)-P\Big(\mathbf{X}_{\hat{k}}^{\hat{k}+1}=(\hat{i},\hat{j})\Big)\Big\} \\ &\quad=\sum_{(\hat{i},\hat{j})} b_{\hat{i},i} b_{\hat{j},j} P\Big(X_{\hat{k}+1}=\hat{j}\Big|X_{\hat{k}}=\hat{i}\Big)\Big\{P\Big(X_{\hat{k}}=\hat{i}\Big|X_{k+1}=\tilde{j}_{max}\Big)-P\Big(X_{\hat{k}}=\hat{i}\Big)\Big\}\leq N^2\rho^{\hat{k}-k-2}. \end{aligned} \tag{A.1.5}$$

It is a well known result in MCs that the absolute value of the term $P\big(X_{\hat{k}}=\hat{i}\big|X_{k+1}=\tilde{j}_{max}\big)-P\big(X_{\hat{k}}=\hat{i}\big)$ in (A.1.5) can be bounded by $\rho^{\hat{k}-k-2}$ (exponential forgetting), for some constant $0\leq\rho<1$. This is because the MC is irreducible due to the assumption $a_{i,j}\geq\delta$ (see Equation (2.2) on page 173 in Doob [20]). This bound does not depend on $\tilde{j}_{max}$. The final bound does not depend on β because $b_{\hat{i},i}\leq 1$ and $b_{\hat{j},j}\leq 1$. □

A.2 Normalized KL Divergence under HMC Model

In this section, we formulate and later prove several usefull properties of normalized KL divergence and its derivatives between cover and stego distributions when derived under Assumptions 1–3.

In addition to the notation developed before, we use the following symbols. We use $\mathcal{P}_\epsilon(\mathcal{I})$ to denote set of probability distributions on set $\mathcal{I}=\{1,\dots,N\}$ lower-bounded by ϵ, i.e., $\mathbf{p}=(p_1,\dots,p_N)^T\in\mathcal{P}_\epsilon(\mathcal{I})\Rightarrow p_i\geq\epsilon$ for all i. We define $\mathbb{B}(y)=(b_{i,j}(y))$ as diagonal matrix with $b_{i,i}(y)=b_{i,y}$ and vectors $\mathbf{b}(y)=(b_{1,y},\dots,b_{N,y})^T$, $\mathbf{e}=(1,\dots,1)^T$, $\mathbf{e}_i$ as ith standard basis vector. Sometimes we write $\mathbb{B}_\beta(y)$ and $\mathbf{b}_\beta(y)$ to stress the dependency on parameter β. We write ∂f as a shorthand for $\frac{\partial}{\partial\beta}f$. For vector $\mathbf{x}$ and matrix $\mathbb{M}$, we denote $\|\mathbf{x}\|_1$ the L_1 norm, $\|\mathbf{x}\|_1=\sum_i|x_i|$,$\|\mathbf{x}\|$ the L_2 norm, $\|\mathbf{x}\|=(\sum_i x_i^2)^{1/2}$, and $\|\mathbb{M}\|$ the 2-norm of matrix $\mathbb{M}$, i.e., $\|\mathbb{M}\|=\sup_{\mathbf{x}\neq 0}\frac{\|\mathbb{M}\mathbf{x}\|}{\|\mathbf{x}\|}=\sup_{\|\mathbf{x}\|=1}\|\mathbb{M}\mathbf{x}\|$ (See for example [53, Sect. 2.2, p. 14–15]).

As its result, we will have the fact, that $\frac{1}{n}d_n(\beta)$ and its derivatives are continuous and uniformly bounded functions of β.

Lemma A.2. *Every derivative of normalized KL divergence $\frac{1}{n}d_n(\beta)=D_{KL}\big(P^{(n)}||Q_\beta^{(n)}\big)$ between n-element distributions of $\mathbf{X}_1^n$ distributed accordding to $P^{(n)}$ and $\mathbf{Y}_1^n$ distributed according to $Q_\beta^{(n)}$ embedded with parameter β is uniformly bounded,*

$$\forall k\geq 0,\exists C_k<\infty,\forall n,\forall\beta\in[0,\beta_0],\left|\frac{1}{n}\frac{\partial^k}{\partial\beta^k}d_n(\beta)\right|<C_k, \tag{A.2.1}$$

and is Lipschitz-continuous (shortly Lipschitz) w.r.t. parameter β, i.e.,

$$\forall k\geq 0,\exists L_k<\infty,\forall n,\forall\beta,\beta'\in[0,\beta_0],\frac{1}{n}\left|\frac{\partial^k}{\partial\beta^k}d_n(\beta)-\frac{\partial^k}{\partial\beta^k}d_n(\beta')\right|<L_k|\beta-\beta'|. \tag{A.2.2}$$

Constant $\beta_0>0$ is given in the proof.

The problem of bounding normalized derivatives of KL divergence for the case of HMC was studied by Mevel et al. [85]. Their results, namely Theorem 4.4 and Theorem 4.7, however, cannot be directly applied to our case because our assumptions are different. In particular, Assumption C on page 1124 is not satisfied because we allow zeros in matrix $\mathbb{B}$. Motivated by this work, we need to derive a more general result about the normalized KL divergence and its derivatives. Intuitively, we can expect the normalized KL divergence to be arbitrarily smooth and bounded due to the smooth transition from P to Q_β and the fact that $d_n(0)=0$.

Before proving Lemma A.2, we introduce a concept of a prediction filter and prove several of its properties. Some definitions were adopted from the work of Mevel and Finesso [85] and are considered classic for Hidden Markov Chains (HMCs).

A.3 Properties of Prediction Filters

We view an HMC as a stochastic process $(X_n, Y_n)_{n=1}^{\infty}$ (sequence of random variables), with $(X_n)_{n=1}^{\infty}$ being Markov Chain (MC), $X_n \in \mathcal{I} = \{1, \dots, N\}$, and each Y_n non-deterministic function of internal state X_n. Only Y_n, $n \in \mathbb{N}$, are observable. The following corollary is a simple consequence of Assumption 2 and defines certain constants and bounds which will be used later.

Corollary A.1. *By the Perron-Frobenius theorem $\|\mathbb{A}^T\| = 1$. By $a_{i,j} \geq \delta > 0$, MC $(X_n)_{n=1}^{\infty}$ is irreducible and $\pi_i \geq \delta$ (see [20, p. 173, Eq. 2.1]), $\boldsymbol{\pi} \in \mathcal{P}_\delta(\mathcal{I})$. If $\mathbf{p} \in \mathcal{P}_\delta(\mathcal{I})$, then $\mathbf{b}^T(y)\mathbf{p} \geq \delta \sum_i b_{i,y} \geq \delta(1 + \beta c_{y,y}) \geq \delta(1 + \beta_1 \min_y c_{y,y}) = \delta_1 > 0$ for $\beta \in [0, \beta_1]$, where $1 + \beta_1 \min_y c_{y,y} > 0$. We will need the following bounds, $\|\mathbf{b}_\beta(y)\| \leq S_0$, $\|\partial \mathbf{b}_\beta(y)\| = \|\mathbb{C}_{\bullet,y}\| \leq S_1$. By the assumption, $S_0 < \infty$ and $S_1 < \infty$.*

For some fixed output $\mathbf{y}_1^{n-1} \in \mathcal{I}^{n-1}$, we define a column vector $\mathbf{p}^{(n)} = (p_1^{(n)}, \dots, p_N^{(n)})^T$, called *prediction filter*, as $p_i^{(n)} = P(X_n = i | \mathbf{Y}_1^{n-1} = \mathbf{y}_1^{n-1})$. Sometimes we use $\mathbf{p}_\beta^{(n)}$ to stress the dependency on β. Filter $\mathbf{p}^{(n+1)}$ can be recursively calculated from $\mathbf{p}^{(n)}$, given observation y_n, by using the so-called forward Baum equation as[1]

$$\mathbf{p}^{(n+1)} = \frac{\mathbb{A}^T \mathbb{B}(y_n)\mathbf{p}^{(n)}}{\mathbf{b}^T(y_n)\mathbf{p}^{(n)}}. \tag{A.3.1}$$

Similarly as in [85], we define *approximate prediction filter* as

$$f_\beta(y, \mathbf{p}) \triangleq \mathbb{A}^T \frac{\mathbb{B}(y)\mathbf{p}}{\mathbf{e}^T \mathbb{B}(y)\mathbf{p}} = \mathbb{A}^T \frac{p_y \mathbf{e}_y + \beta \mathbb{C}(y)\mathbf{p}}{p_y + \beta \mathbf{e}^T \mathbb{C}(y)\mathbf{p}} = \mathbb{A}^T \frac{\mathbf{e}_y + \frac{\beta}{p_y}\mathbb{C}(y)\mathbf{p}}{1 + \frac{\beta}{p_y}\mathbf{e}^T \mathbb{C}(y)\mathbf{p}}, \tag{A.3.2}$$

where $\mathbb{C}(y) = diag(\mathbb{C}_{\bullet,y})$. Important case of this expression is $\beta = 0$, then $f_0(y, \mathbf{p}) = (\mathbb{A}_{y,\bullet})^T$ regardless of $\mathbf{p}$. This reflects the fact that the distribution of X_{n+1} is exactly given by y_nth row of matrix $\mathbb{A}$ since $y_n = x_n$, because the case $\beta = 0$ represents pure MC.

For given observation sequence $\mathbf{y}_1^n$, we define the *normalized log-likelihood function* as $l_n(\beta, \mathbf{y}_1^n) = \frac{1}{n} \ln Q_\beta(\mathbf{Y}_1^n = \mathbf{y}_1^n)$. This can be written in terms of prediction filter $\mathbf{p}_\beta^{(i)}$ as

$$l_n(\beta, \mathbf{y}_1^n) = \frac{1}{n} \sum_{i=1}^{n} \ln \left(\mathbf{b}_\beta^T(y_i)\mathbf{p}_\beta^{(i)}\right), \tag{A.3.3}$$

because $\ln Q_\beta(\mathbf{Y}_1^n = \mathbf{y}_1^n) = \ln \prod_{i=1}^n \mathbf{b}^T(y_i) Q_\beta(X_i | \mathbf{Y}_1^{i-1} = \mathbf{y}_1^{i-1})$.

Under the assumption that the initial distribution on MC $(X_n)_{n=1}^{\infty}$ is chosen to be stationary distribution $\boldsymbol{\pi}$, then $\mathbf{p}^{(1)} = \boldsymbol{\pi}$. If $\beta = 0$, then from (A.3.1) we have $\mathbf{p}^{(n)} = \boldsymbol{\pi}$.

One of the key property of the approximate prediction filter is expressed in the following lemma. It states that for all $\beta \in [0, \beta_2]$ the approximate prediction filter satisfies contraction property in $\mathbf{p}$ and in β, independently of the choice of $y \in \mathcal{I}$.

Lemma A.3. *Approximate prediction filter $f_\beta(y, \mathbf{p})$ satisfies the following contraction properties*

$$\|f_\beta(y, \mathbf{p}) - f_\beta(y, \mathbf{q})\| \leq \lambda_1 \|\mathbf{p} - \mathbf{q}\| \tag{A.3.4}$$

$$\|f_\beta(y, \mathbf{p}) - f_{\beta'}(y, \mathbf{p})\| \leq \lambda_2 |\beta - \beta'| \tag{A.3.5}$$

for all values of $\beta, \beta' \in [0, \beta_2]$, $\mathbf{p}, \mathbf{q} \in \mathcal{P}_\delta(\mathcal{I})$ and output $y \in \mathcal{I}$, where constants $\lambda_1 < 1$ and $\lambda_2 < \infty$ depend only on δ and matrix $\mathbb{C}$.

Proof. We use the vector form of the mean value theorem (V-MVT)[56] to derive both inequalities

$$\|f_\beta(y, \mathbf{p}) - f_\beta(y, \mathbf{q})\| \leq \sup_{t \in [0,1]} \left\| \frac{\partial}{\partial \mathbf{p}} f(y, \mathbf{p} + t(\mathbf{q} - \mathbf{p})) \right\| \|\mathbf{p} - \mathbf{q}\|,$$

[1] This can be easily proved and is considered as a classical description of HMC. For details see [22, p. 1538].

where $\mathbb{J}(\tilde{\mathbf{p}}) = (j_{i,l}) \triangleq \frac{\partial}{\partial \mathbf{p}} f(y, \tilde{\mathbf{p}})$ is the Jacobian matrix of function $f_\beta(y, \mathbf{p})$ w.r.t. $\mathbf{p}$ calculated at point $\tilde{\mathbf{p}} = \mathbf{p} + t(\mathbf{q} - \mathbf{p})$. We consider the following bound for matrix 2-norm (see [53, 2.2-15, p. 15]) $\|\mathbb{M}\| \leq \sqrt{N} \max_j \sum_i |m_{i,j}| = \sqrt{N} \max_j \|\mathbb{M}_{\bullet,j}\|_1$ and calculate the jth column of the Jacobian matrix $\mathbb{J}$ by differentiating (A.3.2). If $j \neq y$, then (differentiate the last but one term in (A.3.2))

$$\mathbb{J}(\tilde{\mathbf{p}})_{\bullet,j} = \mathbb{A}^T \left(\frac{\beta \mathbb{C}(y) \mathbf{e}_i}{\tilde{p}_y + \beta \mathbf{e}^T \mathbb{C}(y) \tilde{\mathbf{p}}} - \frac{\beta(\tilde{p}_y \mathbf{e}_y + \beta \mathbb{C}(y) \tilde{\mathbf{p}}) c_{i,y}}{(\tilde{p}_y + \beta \mathbf{e}^T \mathbb{C}(y) \tilde{\mathbf{p}})^2} \right) = \mathbb{A}^T \beta \mathbb{M}_j(\beta, y, \mathbf{p}),$$

if $j = y$, then (differentiate the last term in (A.3.2))

$$\begin{aligned}
\mathbb{J}(\tilde{\mathbf{p}})_{\bullet,y} &= \mathbb{A}^T \beta \left(\frac{\mathbb{C}(y)}{1 + \beta \mathbf{e}^T \mathbb{C}(y) \tilde{\mathbf{p}} / \tilde{p}_y} - \frac{\mathbf{e}^T \mathbb{C}(y)(\mathbf{e}_y + \beta \mathbb{C}(y) \tilde{\mathbf{p}} / \tilde{p}_y)}{(1 + \beta \mathbf{e}^T \mathbb{C}(y) \tilde{\mathbf{p}} / \tilde{p}_y)^2} \right) \left(\frac{\tilde{p}_y \mathbf{e}_y - \mathbf{p}}{(\tilde{p}_y)^2} \right) \\
&= \mathbb{A}^T \beta \mathbb{M}_y(\beta, y, p),
\end{aligned}$$

where $\mathbb{C}(y) = diag(\mathbb{C}_{\bullet,y})$. We know that $\|\mathbb{A}\| = 1$. By the Assumption 2 and by $\mathbf{e}^T \mathbb{B}(y) \mathbf{p} \geq \delta_1$ for $\beta \in [0, \beta_1]$, we can find $C < \infty$, such that $\|\mathbb{M}_j(\beta, y, \mathbf{p})\|_1 \leq C$ for all $j \in \mathcal{I}$ and thus we set $\lambda_1 = \sqrt{N} C \beta_2$, where β_2 satisfies $\beta_2 < (\sqrt{N} C)^{-1}$. If $\beta_2 > \beta_1$, then we set $\beta_2 = \beta_1$. Constant $\lambda_1 < 1$ does not depend on the choice of y, $\beta \in [0, \beta_2]$ and $p \in \mathcal{P}_\delta(\mathcal{I})$.

In order to prove the second statement, we find an upper bound for $\|\partial f(\tilde{\beta}) / \partial \beta\|$, $\tilde{\beta} \in [\beta, \beta']$ by using V-MVT. Partial derivative of (A.3.2) w.r.t. β can be written as

$$\frac{\partial}{\partial \beta} f_{\tilde{\beta}}(y, \mathbf{p}) = \mathbb{A}^T \left(\frac{\mathbb{C}(y) \mathbf{p}}{\mathbf{e}^T \mathbb{B}(y) \mathbf{p}} - \frac{(p_y \mathbf{e}_y + \tilde{\beta} \mathbb{C}(y) \mathbf{p}) \mathbf{e}^T \mathbb{C}(y) \mathbf{p}}{(\mathbf{e}^T \mathbb{B}(y) \mathbf{p})^2} \right).$$

Since $\beta, \beta' \in [0, \beta_2] \subset [0, \beta_1]$, $\tilde{\beta} \in [0, \beta_1]$ and thus $\mathbf{e}^T \mathbb{B}(y) \mathbf{p} \geq \delta_1$. By $\|\mathbb{A}\| = 1$, we can prove that $\|\partial f(\tilde{\beta}) / \partial \beta\|$ is finite and can be bounded by λ_2. □

By using the above lemma, we can prove Lipschitz property of approximate prediction filter w.r.t. parameter β.

Lemma A.4. *The functions $\beta \to f_\beta(\mathbf{y}_1^n, \mathbf{p})$, such as $f_\beta(\mathbf{y}_1^n, \mathbf{p}) \triangleq f_\beta(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p}))$ are Lipschitz on $\mathcal{P}_\delta(\mathcal{I})$ w.r.t. $\beta \in [0, \beta_2]$, i.e., if $\beta, \beta' \in [0, \beta_2]$ then*

$$\omega(n) \triangleq \sup_{p \in \mathcal{P}_\delta(\mathcal{I})} \|f_\beta(\mathbf{y}_1^n, \mathbf{p}) - f_{\beta'}(\mathbf{y}_1^n, \mathbf{p})\| \leq Lip(f)|\beta - \beta'|.$$

Constant $Lip(f)$ does not depend on the choice of $\mathbf{y}_1^n \in \mathcal{I}^n$.

Proof. We prove $\omega(n) \leq \big(\lambda_2 + \lambda_2 \sum_{i=1}^{n-1} \lambda_1^i\big)|\beta - \beta'|$ for $\beta \in [0, \beta_2]$ by induction on n. By using (A.3.5), we have

$$\big\|f_\beta(y, \mathbf{p}) - f_{\beta'}(y, \mathbf{p})\big\| \leq \lambda_2 |\beta - \beta'|.$$

For $n > 1$ we have

$$\begin{aligned}
\|f_\beta(\mathbf{y}_1^n, \mathbf{p}) - f_{\beta'}(\mathbf{y}_1^n, \mathbf{p})\| \leq{}& \|f_\beta(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p})) - f_{\beta'}(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p}))\| + \\
&+ \|f_{\beta'}(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p})) - f_{\beta'}(y_n, f_{\beta'}(\mathbf{y}_1^{n-1}, \mathbf{p}))\|.
\end{aligned}$$

By definition of prediction filter (A.3.2), $f_\beta(y_1, \bullet) : \mathcal{P}_\delta(\mathcal{I}) \to \mathcal{P}_\delta(\mathcal{I})$, because (A.3.2) can be seen as convex combination of rows of $\mathbb{A}$. By Lemma A.3 , $\|f_\beta(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p})) - f_{\beta'}(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p}))\| \leq \lambda_2 |\beta - \beta'|$. By (A.3.4) and by the induction hypothesis, we can bound the second term as

$$\begin{aligned}
\|f_{\beta'}(y_n, f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p})) - f_{\beta'}(y_n, f_{\beta'}(\mathbf{y}_1^{n-1}, \mathbf{p}))\| &\leq \lambda_1 \|f_\beta(\mathbf{y}_1^{n-1}, \mathbf{p}) - f_{\beta'}(\mathbf{y}_1^{n-1}, \mathbf{p})\| \\
&\leq \lambda_1 \big(\lambda_2 + \lambda_2 \sum_{i=1}^{n-2} \lambda_1^i\big)|\beta - \beta'|
\end{aligned}$$

and thus $\omega(n) \leq \big(\lambda_2 + \lambda_2 \sum_{i=1}^{n-1} \lambda_1^i\big)|\beta - \beta'|$. By Lemma A.3, $\lambda_1 < 1$ for $\beta \in [0, \beta_2]$ and thus the whole bound is convergent and $Lip(f) = \lim_{n \to \infty} \lambda_2 + \lambda_2 \sum_{i=1}^{n-1} \lambda_1^i = \lambda_2 + \frac{\lambda_1}{1 - \lambda_1}$. □

The following lemma will be useful for proving Lipschitz property of some class of functions.

Lemma A.5. *Let g_1, g_2 be real Lipschitz functions, then the following holds: (A) function $g_1 \pm g_2$ is Lipschitz; (B) if $|g_1|$, $|g_2|$ are upper bounded, then function $g_1 \cdot g_2$ is Lipschitz; (C) if $|g_1|$, $|g_2|$ are bounded from above and below, respectively and if $1/g_2$ is differentiable, then function $\frac{g_1}{g_2}$ is Lipschitz; (D) if g_1' and g_2' are Lipschitz and $|g_1'|$, $|g_2'|$ bounded, then $(g_1 \cdot g_2)'$ and $(g_1/g_2)'$ are Lipschitz.*

Proof. Let $|g_i(x)-g_i(x')| \leq G_i|x-x'|$ for $i \in \{1,2\}$. (A) $|(g_1 \pm g_2)(x)-(g_1 \pm g_2)(x')| \leq (G_1+G_2)|x-x'|$. Let $G_1^- \leq |g_i(x)| \leq G_1^+$ for all possible x. (B) $|(g_1 \cdot g_2)(x)-(g_1 \cdot g_2)(x')| \leq |(g_1(x)||g_2(x)-g_2(x')|+|g_2(x')||g_1(x)-g_1(x')| \leq (G_1^+G_2+G_2^+G_1)|x-x'|$. (C) $|\frac{g_1(x)}{g_2(x)}-\frac{g_1(x')}{g_2(x')}| \leq \frac{1}{|g_2(x)|}|g_1(x)-g_1(x')|+|g_1(x')||\frac{1}{g_2(x)}-\frac{1}{g_2(x')}|$. By the MVT for function $1/g_2$, $\frac{1}{g_2(x)}-\frac{1}{g_2(x')}=\frac{-g_2'(\tilde{x})}{(g_2(\tilde{x}))^2}(x-x')$. From the Lipschitz property of g_2, we obtain $|g_2'(\tilde{x})| \leq G_2$ and hence $|\frac{g_1(x)}{g_2(x)}-\frac{g_1(x')}{g_2(x')}| \leq \big(\frac{G_1}{G_2^-}+\frac{G_1^+}{(G_2^-)^2}G_2\big)|x-x'|$. Case (D) holds, because $(g_1 \cdot g_2)' = g_1' \cdot g_2 + g_1 \cdot g_2'$ is Lipschitz by using (A),(B). Same holds for $(g_1/g_2)'$. □

Boundedness and Lipschitz property of $\|\partial^k p^{(i)}\|$ are stated and proved below.

Lemma A.6. *The functions $\beta \to \partial^l f_\beta(\mathbf{y}_1^n,\mathbf{p})$ are bounded and Lipschitz on $\mathcal{P}_\delta(\mathcal{I})$ w.r.t. $\beta \in [0,\beta_3]$ for some $0 < \beta_3 \leq \beta_2$, i.e., if $\beta,\beta' \in [0,\beta_3]$ then*

$$\sup_{p\in\mathcal{P}_\delta(\mathcal{X})} \|\partial^l f_\beta(\mathbf{y}_1^n,\mathbf{p})\| \quad \leq \quad P_l, \tag{A.3.6}$$

$$\sup_{p\in\mathcal{P}_\delta(\mathcal{I})} \|\partial^l f_\beta(\mathbf{y}_1^n,\mathbf{p}) - \partial^l f_{\beta'}(\mathbf{y}_1^n,\mathbf{p})\| \quad \leq \quad Lip(\partial^l f)|\beta-\beta'|. \tag{A.3.7}$$

Constants $Lip(\partial^l f)$ and P_l does not depend on the choice of $\mathbf{y}_1^n \in \mathcal{I}^n$.

Proof. We prove (A.3.6) and (A.3.7) for $l=1$ and show how to generalize this approach for higher derivatives. First derivative of prediction filter can be written as

$$\partial\mathbf{p}^{(n+1)} = \partial f_\beta(\mathbf{y}_1^n,\mathbf{p}) = \mathbb{A}^T\partial\frac{\mathbb{B}(y_n)\mathbf{p}^{(n)}}{\mathbf{b}^T(y_i)\mathbf{p}^{(n)}} = \mathbb{A}^T\mathbb{F}\partial\mathbf{p}^{(n)} + \mathbb{A}^T\mathbb{G}\mathbf{p}^{(n)}, \tag{A.3.8}$$

where

$$\mathbb{F} = \frac{\mathbb{B}(y_n)}{\mathbf{b}^T(y_n)\mathbf{p}^{(n)}}\left(\mathbb{I}-\frac{\mathbf{p}^{(n)}\mathbf{b}^T(y_n)}{\mathbf{b}^T(y_n)\mathbf{p}^{(n)}}\right) \qquad \mathbb{G} = \frac{\partial\mathbb{B}(y_n)}{\mathbf{b}^T(y_n)\mathbf{p}^{(n)}} - \frac{\mathbb{B}(y)\mathbf{p}^{(n)}\partial\mathbf{b}^T(y_n)}{(\mathbf{b}^T(y_n)\mathbf{p}^{(n)})^2}. \tag{A.3.9}$$

In the rest of this proof, we will need $\|\mathbb{A}^T\mathbb{F}\| < 1$ for $\beta \in [0,\beta_2]$ which we prove now. If $\mathbb{C}(y) = diag(\mathbb{C}_{\bullet,y})$, then by $\mathbf{b}^T(y)\mathbf{p} \geq \delta_1$ and $\|\mathbb{A}\| = 1$

$$\begin{aligned}\|\mathbb{A}^T\mathbb{F}\| &\leq \delta_1^{-1}\left\|\mathbb{A}^T\left(\mathbf{e}_y - \frac{p_y\mathbf{e}_y+\beta\mathbb{C}(y)\mathbf{p}}{p_y+\beta\mathbf{e}^T\mathbb{C}(y)\mathbf{p}}\right)\mathbf{e}_y^T + \beta\mathbb{A}^T\left(\mathbb{C}(y) - \frac{\mathbf{p}_y\mathbf{e}_y+\beta\mathbb{C}(y)\mathbf{p}}{p_y+\beta\mathbf{e}^T\mathbb{C}(y)\mathbf{p}}\mathbb{C}_{\bullet,y}^T\right)\right\| \\ &\leq \delta_1^{-1}\big\|f_0(y,\mathbf{p}) - f_\beta(y,\mathbf{p})\big\| + \beta\big\|\mathbb{C}(y) - f_\beta(y,\mathbf{p})\mathbb{C}_{\bullet,y}^T\big\| \leq \beta\delta_1^{-1}(\lambda_2+2S_1),\end{aligned}$$

where $\mathbf{p} = \mathbf{p}^{(n)}$, $y = y_n$ and thus we can find $0 < \beta_3 \leq \beta_2$ such that $\|\mathbb{A}^T\mathbb{F}_\beta\| \leq \beta_3\delta_1^{-1}(\lambda_2+2S_1) = \lambda_3 < 1$ for $\beta \in [0,\beta_3]$. We call this "*contraction property*" of $\mathbb{A}^T\mathbb{F}$.

By Assumption 2, $\|\mathbb{G}\|$ is upper bounded. By this and by contraction property of $\mathbb{A}^T\mathbb{F}$, $\|\partial\mathbf{p}^{(n+1)}\| \leq \|\mathbb{A}^T\mathbb{F}\|\|\partial\mathbf{p}^{(n)}\| + \|\mathbb{A}^T\|\|\mathbb{G}\|\|\mathbf{p}^{(n)}\|$ is recurrent expression for an upper bound on $\|\partial\mathbf{p}^{(n+1)}\|$. This upper bound converges to finite number P_1, because $\partial\mathbf{p}^{(1)} = 0$ — initial distribution does not depend on β, it is equal to $\boldsymbol{\pi}$. This bound does not depend on $\mathbf{p} \in \mathcal{P}_\delta(\mathcal{I})$, $y \in \mathcal{I}$.

By Lemma A.5, $\mathbb{F}$ and $\mathbb{G}$ are Lipschitz in 2-norm w.r.t. β, because they were obtained by combination of Lipschitz and bounded terms, remember $\mathbf{b}^T(y)\mathbf{p} \geq \delta_1$ and $\|\partial\mathbb{B}(y)\|$ if finite. Now we can prove (A.3.7), because by (A.3.8) and by adding and subtracting $\mathbb{A}^T\mathbb{F}_{\beta'}\partial\mathbf{p}_\beta^{(n)}$

$$\begin{aligned}\|\partial\mathbf{p}_\beta^{(n+1)} - \partial\mathbf{p}_{\beta'}^{(n+1)})\| &\leq \|\mathbb{A}^T\mathbb{F}_{\beta'}\|\|\partial\mathbf{p}_\beta^{(n)} - \partial\mathbf{p}_{\beta'}^{(n)}\| + \|\mathbb{A}^T\|\|\mathbb{F}_\beta - \mathbb{F}_{\beta'}\|\|\partial\mathbf{p}_\beta^{(n)}\| + \\ &+ \|\mathbb{A}^T\|\|\mathbb{G}_\beta\mathbf{p}_\beta^{(n)} - \mathbb{G}_{\beta'}\mathbf{p}_{\beta'}^{(n)}\| \leq \lambda_3\|\partial\mathbf{p}_\beta^{(n)} - \partial\mathbf{p}_{\beta'}^{(n)}\| + (Lip(\mathbb{F}) + Lip(\mathbb{G}\mathbf{p}))|\beta-\beta'|,\end{aligned}$$

where we used Lipschitz property of $\mathbb{G}_\beta \mathbf{p}_\beta^{(n)}$ w.r.t. β (use Lemma A.5) and Lipschitz property of $\mathbb{F}$. Again, this recurrent bound converges to finite limit $Lip(\partial f)$ because of contraction property of $\mathbb{A}^T\mathbb{F}_{\beta'}$ and $\|\partial \mathbf{p}_\beta^{(1)} - \partial \mathbf{p}_{\beta'}^{(1)}\| = 0|\beta - \beta'|$.

By a closer look at higher derivatives of (A.3.2), we can realize that

$$\partial^l \mathbf{p}^{(n+1)} = \mathbb{A}^T \mathbb{F} \partial^l \mathbf{p}^{(n)} + R_{n,\beta}\Big(y_n, \mathbf{p}^{(n)}, \partial \mathbf{p}^{(n)}, \ldots, \partial^{l-1}\mathbf{p}^{(n)}\Big),$$

where $R_{n,\beta}(\cdots)$ is Lipschitz w.r.t. derivatives of $\mathbf{p}^{(n)}$ up to order $l-1$. Same observation was mentioned and used by Mevel and Finesso in [85, p. 1127]. This observation is possible, since $\mathbb{B}(y)$, $\partial\mathbb{B}(y)$ are bounded and Lipschitz and $\partial^l \mathbb{B}(y) = 0$ for $l \geq 2$. By this recursion, induction hypothesis ((A.3.6) and (A.3.7) holds up to $l-1$) and the fact that $\mathbb{A}^T\mathbb{F}$ is contracting, we can find finite upper bound P_l for $\|\partial^l \mathbf{p}^{(n)}\|$. Same approach applies to (A.3.7). □

The following lemmas are related to the problem of sub-exponential forgetting of the derivatives of the prediction filter. From Lemma A.3, we know that prediction filter is forgetting its initial condition with exponential rate. By this result, sequence of realizations of prediction filters can be seen as nearly mutually independent and thus classical laws such as Central Limit Theorem (CLT) and Law of Large Numbers (LLN) can be proved. In the next, we will show that derivatives of prediction filter have similar property and thus as a result, we can prove the CLT for first derivative and LLN for second derivative of log-likelihood function. This is because from (A.3.3) the log-likelihood can be written as a sum of terms of prediction filters and its derivatives. The CLT for first derivative allows us to prove the LAN (local asymptotic normality) of log-likelihood ratio test statistics.

First we show some simple properties of matrices $\mathbb{F}$ and $\mathbb{G}$ from (A.3.8) which will be necessary.

Lemma A.7. *Let matrix $\mathbb{F}(\mathbf{p})$ and $\mathbb{G}(\mathbf{p})$ be defined as in (A.3.9) for fixed prediction filter $\mathbf{p}$, then these matrices are continuous and bounded in L_2 norm w.r.t. $\mathbf{p}$ for all $\beta \in [0, \beta_3]$, i.e., for $\mathbf{p}, \mathbf{p}' \in \mathcal{P}_\delta(\mathcal{I})$*

$$\begin{aligned} \|\mathbb{F}(\mathbf{p}) - \mathbb{F}(\mathbf{p}')\| &\leq C_f \|\mathbf{p} - \mathbf{p}'\|, & \|\mathbb{F}(\mathbf{p})\| &\leq D_f < 1, \\ \|\mathbb{G}(\mathbf{p}) - \mathbb{G}(\mathbf{p}')\| &\leq C_g \|\mathbf{p} - \mathbf{p}'\|, & \|\mathbb{G}(\mathbf{p})\| &\leq D_g, \end{aligned}$$

for some finite constants C_f, C_g, and D_g.

Proof. Boundedness of both matrices was proved and mentioned in previous lemmas (use $\|\mathbb{A}\| \leq 1$), thus we prove the continuity only by using V-MVT [56]. By the same approach as in Lemma A.3, it is sufficient to be interested in an upper bound on $\|\mathbb{J}(\tilde{\mathbf{p}})_{\bullet,j}\|_1$, where $\mathbb{J}(\tilde{\mathbf{p}}) = (j_{(i,l),k}) \triangleq (\partial \mathbb{F}_{il}(\tilde{\mathbf{p}})/\partial \mathbf{p}_k)$ is Jacobian matrix of size $N^2 \times N$ calculated at point $\tilde{\mathbf{p}}$ on line between $\mathbf{p}$ and $\mathbf{p}'$. We start with matrix $\mathbb{F}$ and calculate the kth column of the Jacobian matrix as

$$\mathbb{J}(\tilde{\mathbf{p}})_{\bullet,k} = -\frac{\mathbb{B}(y)\mathbf{b}^T(y)\mathbf{e}_k}{(\mathbf{b}^T(y)\mathbf{p})^2} - \mathbb{B}(y)\frac{\mathbf{e}_k \mathbf{b}^T(y)(\mathbf{b}^T(y)\mathbf{p})^2 - 2\mathbf{p}\mathbf{b}^T(y)\mathbf{b}^T(y)\mathbf{p}\mathbf{b}^T(y)\mathbf{e}_k}{(\mathbf{b}^T(y)\mathbf{p})^4}.$$

By Assumption **2** and Corollary A.1, the above matrix (think of it as big vector) is bounded in L_1 norm by some constant C_f. The same steps can be done to show the upper bound in the case of matrix $\mathbb{G}$. □

Now we can prove the fact that sequence $(\mathbf{p}^{(n)}, \partial \mathbf{p}^{(n)})_{n=1}^\infty$ and possible extensions to higher order derivatives are exponentially forgetting their initial values $(\mathbf{p}^{(1)}, \partial \mathbf{p}^{(1)})$.

Lemma A.8. *Function $(f, \partial f)_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial \mathbf{p})$ defined as*

$$\big(f, \partial f\big)_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial\mathbf{p}) \triangleq \big(f_\beta(\mathbf{y}_1^n, \mathbf{p}), \partial f_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial \mathbf{p})\big)$$

is forgetting its initial values $\mathbf{p} \in \mathcal{P}_\delta(\mathcal{I})$ and $\partial\mathbf{p} \in \mathbb{R}^N$ on $\beta \in [0, \beta_3]$ with exponential rate, i.e., if $\mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}_\delta(\mathcal{I})$ and $\partial \mathbf{p}, \partial\hat{\mathbf{p}} \in \mathbb{R}^N$ then

$$\|(f, \partial f)_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial\mathbf{p}) - (f, \partial f)_\beta(\mathbf{y}_1^n, \hat{\mathbf{p}}, \partial\hat{\mathbf{p}})\| \leq C\rho^n \|\mathbf{p} - \hat{\mathbf{p}}\| + \rho^n \|\partial \mathbf{p} - \partial \hat{\mathbf{p}}\|,$$

where $\rho < 1$ and C are constants independent of $\mathbf{y}_1^n \in \mathcal{I}^n$ and choice of $\beta \in [0, \beta_3]$.

Proof. For fixed $\mathbf{y}_1^n \in \mathcal{I}^n$ and $\beta \in [0, \beta_3]$, define

$$\mathbf{p}^{(n+1)} = \begin{cases} \mathbf{p} & \text{if } n = 0 \\ f_\beta(\mathbf{y}_1^n, \mathbf{p}) & \text{otherwise} \end{cases} \qquad \hat{\mathbf{p}}^{(n+1)} = \begin{cases} \hat{\mathbf{p}} & \text{if } n = 0 \\ f_\beta(\mathbf{y}_1^n, \hat{\mathbf{p}}) & \text{otherwise} \end{cases}$$
$$\partial\mathbf{p}^{(n+1)} = \begin{cases} \partial\mathbf{p} & \text{if } n = 0 \\ \partial f_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial\mathbf{p}) & \text{otherwise} \end{cases} \qquad \partial\hat{\mathbf{p}}^{(n+1)} = \begin{cases} \partial\hat{\mathbf{p}} & \text{if } n = 0 \\ \partial f_\beta(\mathbf{y}_1^n, \hat{\mathbf{p}}, \partial\hat{\mathbf{p}}) & \text{otherwise} \end{cases}$$

and sequences Δ_n and δ_n as $\Delta_n = \|\partial\mathbf{p}^{(n)} - \partial\hat{\mathbf{p}}^{(n)}\|$ and $\delta_n = \|\mathbf{p}^{(n)} - \hat{\mathbf{p}}^{(n)}\|$. By expanding $\partial\mathbf{p}^{(n+1)}$ as in (A.3.8), we can find an upper bound on Δ_{n+1}

$$\begin{aligned} \Delta_{n+1} &\le \|\mathbb{A}^T\| \|\mathbb{F}(\mathbf{p}^{(n)})\partial\mathbf{p}^{(n)} - \mathbb{F}(\hat{\mathbf{p}}^{(n)})\partial\hat{\mathbf{p}}^{(n)} + \mathbb{G}(\mathbf{p}^{(n)})\mathbf{p}^{(n)} - \mathbb{G}(\hat{\mathbf{p}}^{(n)})\hat{\mathbf{p}}^{(n)}\| \\ &\le \|\mathbb{F}(\mathbf{p}^{(n)})\|\Delta_n + \|\mathbb{F}(\mathbf{p}^{(n)}) - \mathbb{F}(\hat{\mathbf{p}}^{(n)})\| \|\partial\hat{\mathbf{p}}^{(n)}\| + \\ &\quad + \|\mathbb{G}(\mathbf{p}^{(n)})\|\delta_n + \|\mathbb{G}(\mathbf{p}^{(n)}) - \mathbb{G}(\hat{\mathbf{p}}^{(n)})\| \|\hat{\mathbf{p}}^{(n)}\| \\ &\le D_f\Delta_n + P_1C_f\delta_n + D_g\delta_n + C_g\delta_n = D_f\Delta_n + C_1\delta_n, \end{aligned}$$

where we used continuity and boundedness proved in Lemma A.7, the fact that $\|\partial\mathbf{p}\|$ is bounded (see Lemma A.6) and $C_1 = P_1C_f + D_g + C_g$. By recursion, we obtain

$$\Delta_{n+1} \le D_f\Delta_n + C_1\delta_n \le \cdots \le D_f^n\Delta_1 + C_1\sum_{i=0}^{n-1} D_f^i\delta_{n-i}.$$

From (A.3.4), we have $\delta_{n+1} \le \lambda_1\|\mathbf{p}^{(n)} - \hat{\mathbf{p}}^{(n)}\| \le \lambda_1^n\delta_1$ and thus we can get rid of the sum

$$\begin{aligned} \Delta_{n+1} &\le D_f^n\Delta_1 + C_1\Big(\sum_{i=0}^{n-1}(D_f/\lambda_1)^i\Big)\lambda_1^{n-1}\delta_1 \\ &\le D_f^n\Delta_1 + C_1\Big(\sum_{i=0}^{n-1}(\hat{D}_f/\lambda_1)^i\Big)\lambda_1^{n-1}\delta_1 \\ &\le \hat{D}_f^n\Delta_1 + C_1\frac{\hat{D}_f^n - \lambda_1^n}{\hat{D}_f - \lambda_1}\delta_1 \\ &\le \hat{D}_f^n\Delta_1 + C_2\hat{D}_f^n\delta_1, \end{aligned}$$

where[2] $C_2 = C_1/(\hat{D}_f - \lambda_1)$. Finally, we have

$$\begin{aligned} \left\|(\mathbf{p}^{(n+1)}, \partial\mathbf{p}^{(n+1)}) - (\hat{\mathbf{p}}^{(n+1)}, \partial\hat{\mathbf{p}}^{(n+1)})\right\| &= \sqrt{\delta_{n+1}^2 + \Delta_{n+1}^2} \\ &\le \delta_{n+1} + \Delta_{n+1} \\ &\le (\lambda_1^n + C_2\hat{D}_f^n)\delta_1 + \hat{D}_f^n\Delta_1 \\ &\le 2C_2\hat{D}_f^n\delta_1 + \hat{D}_f^n\Delta_1. \end{aligned}$$

□

From the proof, we can see that exponential forgetting of $\partial\mathbf{p}$ is a consequence of exponential forgetting of $\mathbf{p}$, continuity of matrices $\mathbb{F}$ and $\mathbb{G}$ and contraction of matrix $\mathbb{A}^T\mathbb{F}$ (forgetting previous $\partial\mathbf{p}$). When we consider (A.3.8) and its higher order derivatives w.r.t. β, the same result (exponential forgetting) can be proved for vectors of higher order derivatives of the prediction filter $(\mathbf{p}, \partial\mathbf{p}, \ldots, \partial^l\mathbf{p})$. We formulate this in the next corollary which is presented without the proof, because all the asumptions (continuity, boundedness and contraction) of respective matrices are satisfied and thus same approach can be used in the proof.

[2] We choose $\hat{D}_f$ in order to avoid case $\lambda_1 = D_f$. If $\lambda_1 \ge D_f$, then we choose $\lambda_1 < \hat{D}_f < 1$ and $\hat{D}_f = D_f$ otherwise.

Corollary A.2. *Function* $(f, \partial f, \dots, \partial^l f)_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial\mathbf{p}, \dots, \partial^l \mathbf{p})$ *defined as*

$$\left(f, \dots, \partial^l f\right)_\beta(\mathbf{y}_1^n, \mathbf{p}, \dots, \partial^l \mathbf{p}) \triangleq \left(f_\beta(\mathbf{y}_1^n, \mathbf{p}), \partial f_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial\mathbf{p}), \dots, \partial^l f_\beta(\mathbf{y}_1^n, \mathbf{p}, \partial\mathbf{p}, \dots \partial^l \mathbf{p})\right)$$

is forgetting its initial values $\mathbf{p} \in \mathcal{P}_\delta(\mathcal{I})$ *and* $\partial\mathbf{p}, \dots, \partial^l\mathbf{p} \in \mathbb{R}^N$ *on* $\beta \in [0, \beta_3]$ *with exponential rate, i.e., if* $\mathbf{p}, \hat{\mathbf{p}} \in \mathcal{P}_\delta(\mathcal{I})$ *and* $\partial\mathbf{p}, \partial\hat{\mathbf{p}}, \dots, \partial^l\mathbf{p}, \partial^l\hat{\mathbf{p}} \in \mathbb{R}^N$ *then*

$$\begin{aligned}\|(f, \dots, \partial^l f)_\beta(\mathbf{y}_1^n, \mathbf{p}, \dots, \partial^l \mathbf{p}) - (f, \dots, \partial^l f)_\beta(\mathbf{y}_1^n, \hat{\mathbf{p}}, \dots, \partial^l \hat{\mathbf{p}})\| \\ \le C_1\rho^n\|\mathbf{p} - \hat{\mathbf{p}}\| + C_2\rho^n\|\partial\mathbf{p} - \partial\hat{\mathbf{p}}\| + \dots + C_{l+1}\rho^n\|\partial^l\mathbf{p} - \partial^l\hat{\mathbf{p}}\|\end{aligned}$$

where $\rho < 1$ *and* C_i *are constants independent of* $\mathbf{y}_1^n \in \mathcal{I}^n$ *and choice of* $\beta \in [0, \beta_3]$.

A.4 Proof of Lemma A.2

We use prediction filter to calculate normalized KL divergence and its derivatives.

First, approximate prediction filter is closed to $\mathcal{P}_\delta(\mathcal{I})$, i.e., if $\mathbf{p} \in \mathcal{P}_\delta(\mathcal{I})$, then $f_\beta(y, \mathbf{p}) \in \mathcal{P}_\delta(\mathcal{I})$. This holds, because Equation (A.3.2) can be seen as a convex combination of rows of matrix $\mathbb{A}$ which are in $\mathcal{P}_\delta(\mathcal{I})$. Therefore, if $\mathbf{p}^{(1)} = \boldsymbol{\pi}$, then $\mathbf{p}^{(n)} \in \mathcal{P}_\delta(\mathcal{I})$. From this we obtain the proof of (A.2.1) for $k = 0$, because by using $\ln P(\mathbf{X}_1^n) \le 0$ and $\sum_{\mathbf{y}_1^n} P(\mathbf{X}_1^n = \mathbf{y}_1^n) = 1$ it is sufficient to bound normalized log-likelihood $|l_n(\beta, \mathbf{y}_1^n)| \le C_0$. This can be done, because $\mathbf{p}_\beta^{(n)} \in \mathcal{P}_\delta(\mathcal{I})$ and $\mathbf{b}_\beta^T(y)\mathbf{p}_\beta^{(n)} \ge \delta_1$ for $\beta \in [0, \beta_1]$ and by (A.3.3) $C_0 = -\log\delta_1$.

To prove (A.2.2) for $k = 0$, it is sufficient to prove Lipschitz property for function $\ln(b_\beta^T(y_i)p_\beta^{(i)})$. By the Mean Value Theorem (MVT) used on function $\beta \to \ln(v(\beta)^T z)$, for some vectors $\mathbf{v}$ and $\mathbf{z}$, $\ln(\mathbf{v}(\beta)^T\mathbf{z})/(v(\beta')^T z)| \le \max|\frac{(\partial v(\tilde{\beta})^T)z}{v(\tilde{\beta})^T z}||\beta - \beta'|$ and thus

$$\begin{aligned}|\ln(\mathbf{b}_\beta^T(y_i)\mathbf{p}_\beta^{(i)}) - \ln(\mathbf{b}_{\beta'}^T(y_i)\mathbf{p}_{\beta'}^{(i)})| &\le \left|\ln\frac{\mathbf{b}_\beta^T(y_i)\mathbf{p}_\beta^{(i)}}{\mathbf{b}_{\beta'}^T(y_i)\mathbf{p}_\beta^{(i)}}\right| + \left|\ln\frac{\mathbf{b}_{\beta'}^T(y_i)\mathbf{p}_\beta^{(i)}}{\mathbf{b}_{\beta'}^T(y_i)\mathbf{p}_{\beta'}^{(i)}}\right| \\ &\le \frac{S_1}{\delta_1}|\beta - \beta'| + \frac{Lip(f)S_0}{\delta_1}|\beta - \beta'|.\end{aligned}$$

We use the fact that Lipschitz property of $\mathbf{p}_\beta^{(i)}$ w.r.t. β (see Lemma A.4), i.e., $\|f_\beta[y_1^n, \mathbf{p}] - f_{\beta'}[y_1^n, \mathbf{p}]\| \le Lip(f)|\beta - \beta'|$ implies $\|\partial f_\beta[\mathbf{y}_1^n, \mathbf{p}]\| \le Lip(f)$. This completes the proof of Theorem A.2 for $k = 0$.

Now we show that if we have Lipschitz property and upper bound for derivatives of prediction filter up to order k, then we can prove (A.2.1) and (A.2.2) for k, i.e., we need $\|\partial^j\mathbf{p}_\beta^{(i)} - \partial^j\mathbf{p}_{\beta'}^{(i)}\| \le Lip(\partial^j f)|\beta - \beta'|$ and $\|\partial^j\mathbf{p}_\beta^{(i)}\| \le P_j < \infty$ for $j \le k$. Result for $k = 0$ has been established already. To prove (A.2.1) and (A.2.2) for $k > 0$, it is sufficient to study the derivatives of normalized log-likelihood. First derivative of $l_n(\beta)$ w.r.t. β can be written as

$$\frac{\partial}{\partial\beta}l_n(\beta) \quad = \quad \frac{1}{n}\sum_{i=1}^n \frac{(\partial\mathbf{b}^T(y_i))\mathbf{p}^{(i)} + \mathbf{b}^T(y_i)(\partial\mathbf{p}^{(i)})}{\mathbf{b}^T(y_i)\mathbf{p}^{(i)}}. \tag{A.4.1}$$

Derivatives of $l_n(\beta)$ of order k can be expressed as an average of terms of the form $g_1/(\mathbf{b}^T(y_i)\mathbf{p}^{(i)})^{2^{k-1}}$, where g_1 is linear combination of dot-products of the following vectors $\mathbf{b}^T(y_i)$, $\partial\mathbf{b}^T(y_i)$, $\mathbf{p}^{(i)}$, $\partial\mathbf{p}^{(i)}, \dots, \partial^k\mathbf{p}^{(i)}$. By Lemma A.5, we need upper bound on L_2 norm and Lipschitz property of these vectors to prove boundedness and Lipschitz property for this type of functions, because $|\mathbf{b}^T(y_i)\mathbf{p}^{(i)}| \ge \delta_1 > 0$ for $\beta \in [0, \beta_1]$. Vectors $\mathbf{b}^T(y_i)$ and $\partial\mathbf{b}^T(y_i)$ are bounded and Lipschitz in L_2 norm by Assumption 2 and thus we need to prove the same for $\partial^k\mathbf{p}^{(i)}$. Uniform upper bound and Lipschitz property of $\partial^k\mathbf{p}^{(i)}$ in L_2 norm are stated and proved in Lemma A.6. Finally, we set $\beta_0 = \beta_3$, because $0 < \beta_3 \le \beta_2 \le \beta_1$ and thus all bounds are valid for $\beta \in [0, \beta_3]$. This completes the proof.

Appendix B

Fisher Information - Proofs

In this appendix, we present several lemmas needed in the proof of Theorem 4.2.

Lemma B.1. *Derivatives of log-likelihood of Q_β (as a function of variables $\{b_{i,j}|i,j \in \mathcal{X}\}$) can be written as*

$$\frac{\partial^2}{\partial b_{i,j} b_{k,l}} \ln Q_\beta(\mathbf{Y}_1^n = \mathbf{y}_1^n) = L_1(\mathbf{y}_1^n, i, j, k, l) - [j = l] L_2(\mathbf{y}_1^n, i, j, k),$$

where $i, j, k, l \in \mathcal{I}$, $\mathbf{y}_1^n \in \mathcal{I}^n$ and $L_1(\mathbf{y}_1^n, i, j, k, l)$ and $L_2(\mathbf{y}_1^n, i, j, k)$ are defined in the proof.

Proof. The derivative of $\ln Q_\beta(\mathbf{y}_1^n)$ for a fixed $\mathbf{y}_1^n \in \mathcal{I}^n$ can be written as

$$\frac{\partial^2}{\partial b_{i,j} b_{k,l}} \ln Q_\beta(\mathbf{y}_1^n) = \frac{\frac{\partial^2}{\partial b_{i,j} b_{k,l}} Q_\beta(\mathbf{y}_1^n)}{Q_\beta(\mathbf{y}_1^n)} - \frac{\frac{\partial}{\partial b_{i,j}} Q_\beta(\mathbf{y}_1^n)}{Q_\beta(\mathbf{y}_1^n)} \frac{\frac{\partial}{\partial b_{k,l}} Q_\beta(\mathbf{y}_1^n)}{Q_\beta(\mathbf{y}_1^n)}. \tag{B.0.1}$$

By the independence of embedding operations (MI embedding), $Q_\beta(\mathbf{y}_1^n)$ can be written as

$$Q_\beta(\mathbf{y}_1^n) = \sum_{\mathbf{x}_1^n \in \mathcal{I}^n} P(\mathbf{x}_1^n) \prod_{v=1}^{n} b_{x_v, y_v}. \tag{B.0.2}$$

For a fixed $\mathbf{y}_1^n \in \mathcal{I}^n$, Equation (B.0.2) can be seen as a polynomial w.r.t. the fixed term $b_{i,j}$. The derivative of such a polynomial w.r.t. a given $b_{i,j}$ can be written in the following general form (see Example B.1 for more details)

$$\frac{\partial Q_\beta(\mathbf{y}_1^n)}{\partial b_{i,j}} = \sum_{t \in J(j)} S_y(t, i), \tag{B.0.3}$$

where $J(j) = \{1 \le t \le n | y_t = j\}$ and

$$S_y(t, i) = \sum_{\mathbf{x}_1^n \in \mathcal{I}^n, x_t = i} P(\mathbf{x}_1^n) \prod_{v=1, v \ne t}^{n} b_{x_v y_v}. \tag{B.0.4}$$

In the derivative of (B.0.2), it is sufficient to sum only over the products that contain $b_{i,j}$. If the term is in the form $C b_{i,j}^k$ for some constants k and C, then its derivative is $C k b_{i,j}^{k-1}$. This is achieved by summing over all elements from the set $J(j)$, fixing $x_t = i$ for each $t \in J(j)$, and putting 1 instead of $b_{i,j}$ in the product.

Similarly, we obtain a general form for $(\partial^2/\partial b_{i,j}b_{k,l})Q_\beta(\mathbf{y}_1^n)$ as

$$\begin{aligned}
\frac{\partial^2 Q_\beta(\mathbf{y}_1^n)}{\partial b_{i,j}b_{k,l}} &= \frac{\partial}{\partial b_{k,l}} \sum_{t\in J(j)} S_y(t,i) \\
&= \sum_{t\in J(j)} \sum_{\substack{\mathbf{x}_1^n\in\mathcal{I}^n \\ x_t=i}} P(\mathbf{x}_1^n) \frac{\partial}{\partial b_{k,l}} \prod_{\substack{v=1 \\ v\neq t}}^{n} b_{x_v,y_v} \\
&= \sum_{t\in J(j)} \sum_{\substack{\mathbf{x}_1^n\in\mathcal{I}^n \\ x_t=i}} P(\mathbf{x}_1^n) \sum_{t'\in J(l)\setminus\{t\}} [x_{t'}=k] \prod_{\substack{v=1 \\ v\notin\{t,t'\}}}^{n} b_{x_v,y_v} \qquad \text{(B.0.5)} \\
&= \sum_{t\in J(j)} \sum_{t'\in J(l)\setminus\{t\}} \sum_{\substack{\mathbf{x}_1^n\in\mathcal{I}^n \\ x_t=i, x_{t'}=k}} P(\mathbf{x}_1^n) \prod_{\substack{v=1 \\ v\notin\{t,t'\}}}^{n} b_{x_v,y_v} \\
&= \sum_{t\in J(j)} \sum_{t'\in J(l)\setminus\{t\}} S_y(t,t',i,k), \qquad \text{(B.0.6)}
\end{aligned}$$

where

$$S_y(t,t',i,k) = \sum_{\substack{\mathbf{x}_1^n\in\mathcal{I}^n \\ x_t=i, x_{t'}=k}} P(\mathbf{x}_1^n) \prod_{\substack{v=1 \\ v\notin\{t,t'\}}}^{n} b_{x_v,y_v}.$$

In (B.0.5), we used the fact that $(d/dx)Cx^k = Ckx^{k-1} = \sum_{v=1}^{k} Cx^{k-1}$ again.

We now substitute (B.0.3) and (B.0.6) into (B.0.1) and obtain

$$\begin{aligned}
\frac{\partial^2}{\partial b_{i,j}b_{k,l}} \ln Q_\beta(\mathbf{y}_1^n) &= \sum_{t\in J(j)} \sum_{t'\in J(l)\setminus\{t\}} \frac{S_y(t,t',i,k)}{Q_\beta(y_1^n)} - \sum_{t\in J(j)} \frac{S_y(t,i)}{Q_\beta(y_1^n)} \sum_{t'\in J(l)} \frac{S_y(t',k)}{Q_\beta(y_1^n)} \\
&= L_1(\mathbf{y}_1^n,i,j,k,l) - [j=l] L_2(\mathbf{y}_1^n,i,j,k),
\end{aligned}$$

where

$$\begin{aligned}
L_1(\mathbf{y}_1^n,i,j,k,l) &= \sum_{t\in J(j)} \sum_{t'\in J(l)\setminus\{t\}} \left(\frac{S_y(t,t',i,k)}{Q_\beta(y_1^n)} - \frac{S_y(t,i)}{Q_\beta(y_1^n)} \frac{S_y(t,k)}{Q_\beta(y_1^n)} \right) \qquad \text{(B.0.7)} \\
L_2(\mathbf{y}_1^n,i,j,k) &= \sum_{t\in J(j)} \frac{S_y(t,i)}{Q_\beta(\mathbf{y}_1^n)} \frac{S_y(t,k)}{Q_\beta(\mathbf{y}_1^n)}. \qquad \text{(B.0.8)}
\end{aligned}$$

□

Example B.1. $\mathcal{I} = \{1,2\}$, $n=3$, $\mathbf{y}_1^3 = (y_1,y_2,y_3) = (2,2,1)$

$$\begin{aligned}
Q(\mathbf{y}_1^3) &= \sum_{\mathbf{x}_1^3\in\mathcal{I}^3} Q(\mathbf{y}_1^3|\mathbf{x}_1^3)P(\mathbf{x}_1^3) \\
&= \Big(P(1,1,1)b_{1,1} + P(1,1,2)b_{2,1}\Big)b_{1,2}^2 + \Big(P(1,2,1)b_{2,2}b_{1,1} + \\
&\quad + P(1,2,2)b_{2,2}b_{2,1} + P(2,1,1)b_{2,2}b_{1,1} + P(2,1,2)b_{2,2}b_{2,1}\Big)b_{1,2} + \\
&\quad + \Big(P(2,2,1)b_{2,2}b_{2,2}b_{1,1} + P(2,2,2)b_{2,2}b_{2,2}b_{2,1}\Big)
\end{aligned}$$

If $x = b_{1,2}$, then the previous result can be represented as $Ax^2 + Bx + C$. The partial derivative of

$Q(\mathbf{y}_1^3)$ w.r.t. $b_{1,2}$ accepts the following form

$$\begin{aligned}\frac{\partial Q(\mathbf{y}_1^3)}{\partial b_{1,2}} &= 2P(1,1,1)b_{1,2}b_{1,1} + 2P(1,1,2)b_{1,2}b_{2,1}\\ &\quad + P(1,2,1)b_{2,2}b_{1,1} + P(1,2,2)b_{2,2}b_{2,1} + P(2,1,1)b_{2,2}b_{1,1} + P(2,1,2)b_{2,2}b_{2,1}\\ &= P(1,1,1)b_{1,2}b_{1,1} + P(1,1,2)b_{1,2}b_{2,1} + P(1,2,1)b_{2,2}b_{1,1} + P(1,2,2)b_{2,2}b_{2,1}\\ &\quad + P(1,1,1)b_{1,2}b_{1,1} + P(1,1,2)b_{1,2}b_{2,1} + P(2,1,1)b_{2,2}b_{1,1} + P(2,1,2)b_{2,2}b_{2,1},\end{aligned}$$

where in the last step we sum all terms for $x_1 = 1$ and $x_2 = 1$. We do not need to sum the terms with $\mathbf{x}_1^2 = (2,2)$, because they are zero after the derivation (they do not contain $b_{1,2}$). This can be written in a general form as

$$\frac{\partial Q(\mathbf{y}_1^3)}{\partial b_{1,2}} = \sum_{t \in J(2)} S_y(t,1),$$

where $J(2) = \{1,2\}$ (the set of indices t such that $y_t = 2$) and $S_y(t,i)$ is defined by (B.0.4) and

$$\begin{aligned}S_y(1,1) &= P(1,1,1)b_{1,2}b_{1,1} + P(1,1,2)b_{1,2}b_{2,1} + P(1,2,1)b_{2,2}b_{1,1} + P(1,2,2)b_{2,2}b_{2,1},\\ S_y(2,1) &= P(1,1,1)b_{1,2}b_{1,1} + P(1,1,2)b_{1,2}b_{2,1} + P(2,1,1)b_{2,2}b_{1,1} + P(2,1,2)b_{2,2}b_{2,1}.\end{aligned}$$

The second derivative, e.g., if $i = 1$, $j = 2$, $k = 1$, $l = 1$

$$\frac{\partial^2 Q(\mathbf{y}_1^3)}{\partial b_{1,2} b_{1,1}} = 2\Big(P(1,1,1)b_{1,2} + P(1,2,1)b_{2,2}\Big),$$

can be derived in a similar manner and written in a general form as in (B.0.6), where $J(j) = \{1,2\}$, $J(l) = \{3\}$, and

$$\begin{aligned}S_y(1,3,1,1) &= P(1,1,1)b_{1,2} + P(1,2,1)b_{2,2}\\ S_y(2,3,1,1) &= P(1,1,1)b_{1,2} + P(1,2,1)b_{2,2}.\end{aligned}$$

Lemma B.2. *Let $L_1(\mathbf{y}_1^n, i, j, k, l)$ be function of $\mathbf{y}_1^n \in \mathcal{I}^n$ and let matrix $\mathbb{B} = (b_{ij})$ be defined by* (B.0.7)*. Then, for all $i, j, k, l \in \mathcal{I}$ the following limit exists*

$$\lim_{n\to\infty} \frac{1}{n} E_P\big[L_1(\mathbf{Y}_1^n, i, j, k, l)\big|_{\mathbb{B}=\mathbb{I}}\big]$$

and is equal to $U(i,j,k,l)$ as defined by (4.1.12)*. The series converges to the limit with rate $1/n$.*

Proof. First, we show some properties of the terms in (B.0.7). Assuming $|t - t'| > 1$, by $\mathbb{B} = \mathbb{I}$, $y_t = j$, and $y_{t'} = l$ (remember $t \in J(j)$ and $t' \in J(l)$), we have

$$\begin{aligned}&\frac{S_y(t,t',i,k)}{Q_\beta(\mathbf{y}_1^n)} - \frac{S_y(t,i)}{Q_\beta(\mathbf{y}_1^n)}\frac{S_y(t',k)}{Q_\beta(\mathbf{y}_1^n)}\Big|_{\mathbb{B}=\mathbb{I}} =\\ &\qquad = \frac{a_{y_{t-1},i}a_{i,y_{t+1}}}{a_{y_{t-1},j}a_{j,y_{t+1}}}\frac{a_{y_{t'-1},k}a_{k,y_{t'+1}}}{a_{y_{t'-1},l}a_{l,y_{t'+1}}} - \frac{a_{y_{t-1},i}a_{i,y_{t+1}}}{a_{y_{t-1},j}a_{j,y_{t+1}}}\frac{a_{y_{t'-1},k}a_{k,y_{t'+1}}}{a_{y_{t'-1},l}a_{l,y_{t'+1}}} = 0.\end{aligned}$$

This means that the only non-zero terms in (B.0.7) can be the terms for $|t - t'| = 1$. If $t = t' - 1$, $t \notin \{1, n-1\}$, then for $t \in J(j)$ and $t' \in J(l)$

$$\begin{aligned}\frac{S_y(t,t',i,k)}{Q_\beta(\mathbf{y}_1^n)} - \frac{S_y(t,i)}{Q_\beta(\mathbf{y}_1^n)}\frac{S_y(t',k)}{Q_\beta(\mathbf{y}_1^n)}\Big|_{\mathbb{B}=\mathbb{I}} &= \frac{a_{y_{t-1},i}}{a_{y_{t-1},j}}\left(\frac{a_{i,k}}{a_{j,l}} - \frac{a_{i,y_{t+1}}}{a_{j,y_{t+1}}}\frac{a_{y_{t'-1},k}}{a_{y_{t'-1},l}}\right)\frac{a_{k,y_{t'+1}}}{a_{l,y_{t'+1}}}\\ &= \frac{a_{y_{t-1},i}}{a_{y_{t-1},j}}\left(\frac{a_{i,k}}{a_{j,l}} - \frac{a_{i,l}}{a_{j,l}}\frac{a_{j,k}}{a_{j,l}}\right)\frac{a_{k,y_{t+2}}}{a_{l,y_{t+2}}},\end{aligned}$$

because $y_{t'-1} = y_t = j$, and $y_{t+1} = y_{t'} = l$. If $t = t' + 1$, $t \notin \{2, n\}$, then

$$
\begin{aligned}
\frac{S_y(t,t',i,k)}{Q_\beta(\mathbf{y}_1^n)} - \frac{S_y(t,i)}{Q_\beta(\mathbf{y}_1^n)}\frac{S_y(t',k)}{Q_\beta(\mathbf{y}_1^n)}\Big|_{\mathbb{B}=\mathbb{I}} &= \frac{S_y(t',t,k,i)}{Q_{\beta=0}(\mathbf{y}_1^n)} - \frac{S_y(t',k)}{Q_{\beta=0}(\mathbf{y}_1^n)}\frac{S_y(t,i)}{Q_{\beta=0}(\mathbf{y}_1^n)} \\
&= \frac{a_{y_{t-2},k}}{a_{y_{t-2},l}}\left(\frac{a_{k,i}}{a_{l,j}} - \frac{a_{k,j}}{a_{l,j}}\frac{a_{l,i}}{a_{l,j}}\right)\frac{a_{i,y_{t+1}}}{a_{j,y_{t+1}}}.
\end{aligned}
$$

By using both results, we can write

$$
\begin{aligned}
&\frac{1}{n}\sum_{\mathbf{y}_1^n\in\mathcal{I}^n} P(\mathbf{y}_1^n)L_1(\mathbf{Y}_1^n,i,j,k,l)\Big|_{\mathbb{B}=\mathbb{I}} = \\
&\quad = \frac{1}{n}\sum_{t=2}^{n-2}\sum_{\mathbf{y}_1^n\in\mathcal{I}^n} P(\mathbf{y}_1^n)\left(\left[\mathbf{y}_t^{t+1}=(j,l)\right]\frac{a_{y_{t-1},i}}{a_{y_{t-1},j}}\left(\frac{a_{i,k}}{a_{j,l}} - \frac{a_{i,l}}{a_{j,l}}\frac{a_{j,k}}{a_{j,l}}\right)\frac{a_{k,y_{t+2}}}{a_{l,y_{t+2}}}\right) + \\
&\qquad + \frac{1}{n}\sum_{t=3}^{n-1}\sum_{\mathbf{y}_1^n\in\mathcal{I}^n} P(\mathbf{y}_1^n)\left(\left[\mathbf{y}_{t-1}^{t}=(l,j)\right]\frac{a_{y_{t-2},k}}{a_{y_{t-2},l}}\left(\frac{a_{k,i}}{a_{l,j}} - \frac{a_{k,j}}{a_{l,j}}\frac{a_{l,i}}{a_{l,j}}\right)\frac{a_{i,y_{t+1}}}{a_{j,y_{t+1}}}\right) + g_n,
\end{aligned}
$$

where g_n is the sum for $(t,t')\in\{(1,2),(2,1),(n-1,n),(n,n-1)\}$. The series g_n can be sandwiched by $0\le g_n\le C\frac{1}{n}$ for some constant C and thus $\lim_{n\to\infty} g_n = 0$ with rate $O(1/n)$. This constant depends only on elements of matrix $\mathbb{A}$. We can continue and write

$$
\begin{aligned}
&\frac{1}{n}\sum_{\mathbf{y}_1^n\in\mathcal{I}^n} P(\mathbf{y}_1^n)L_1(\mathbf{Y}_1^n,i,j,k,l)\Big|_{\mathbb{B}=\mathbb{I}} - g_n \\
&\quad = \frac{1}{n}\sum_{t=2}^{n-2}\sum_{z_1,z_2\in\mathcal{I}}\frac{a_{z_1,i}}{a_{z_1,j}}\left(\frac{a_{i,k}}{a_{j,l}} - \frac{a_{i,l}}{a_{j,l}}\frac{a_{j,k}}{a_{j,l}}\right)\frac{a_{k,z_2}}{a_{l,z_2}}\underbrace{P\big(\mathbf{y}_{t-1}^{t+2}=(z_1,j,l,z_2)\big)}_{\pi_{z_1}a_{z_1,j}a_{j,l}a_{l,z_2}} + \\
&\qquad + \frac{1}{n}\sum_{t=3}^{n-1}\sum_{z_2,z_1\in\mathcal{I}}\frac{a_{z_2,k}}{a_{z_2,l}}\left(\frac{a_{k,i}}{a_{l,j}} - \frac{a_{k,j}}{a_{l,j}}\frac{a_{l,i}}{a_{l,j}}\right)\frac{a_{i,z_1}}{a_{j,z_1}}\underbrace{P\big(\mathbf{y}_{t-2}^{t+1}=(z_2,l,j,z_1)\big)}_{\pi_{z_2}a_{z_2,l}a_{l,j}a_{j,z_1}} \\
&\quad = \frac{n-3}{n}\sum_{z_1,z_2\in\mathcal{I}}\left\{\pi_{z_1}a_{z_1,i}\left(a_{i,k} - a_{i,l}\frac{a_{j,k}}{a_{j,l}}\right)a_{k,z_2} + \pi_{z_2}a_{z_2,k}\left(a_{k,i} - a_{k,j}\frac{a_{l,i}}{a_{l,j}}\right)a_{i,z_1}\right\} \\
&\quad = \frac{n-3}{n}\left\{\left(a_{i,k} - a_{i,l}\frac{a_{j,k}}{a_{j,l}}\right)\sum_{z_1\in\mathcal{I}}\pi_{z_1}a_{z_1,i} + \left(a_{k,i} - a_{k,j}\frac{a_{l,i}}{a_{l,j}}\right)\sum_{z_2\in\mathcal{I}}\pi_{z_2}a_{z_2,k}\right\} \\
&\quad = \frac{n-3}{n}\left\{\pi_i\left(a_{i,k} - a_{i,l}\frac{a_{j,k}}{a_{j,l}}\right) + \pi_k\left(a_{k,i} - a_{k,j}\frac{a_{l,i}}{a_{l,j}}\right)\right\}.
\end{aligned}
$$

Finally, the limit for $n\to\infty$ is

$$
\begin{aligned}
U(i,j,k,l) &\triangleq \lim_{n\to\infty}\frac{1}{n}E_P\left[L_1(\mathbf{Y}_1^n,i,j,k,l)\Big|_{\mathbb{B}=\mathbb{I}}\right] \\
&= \pi_i\left(a_{i,k} - a_{i,l}\frac{a_{j,k}}{a_{j,l}}\right) + \pi_k\left(a_{k,i} - a_{k,j}\frac{a_{l,i}}{a_{l,j}}\right). \qquad \text{(B.0.9)}
\end{aligned}
$$

□

Lemma B.3. *Let $L_2(\mathbf{y}_1^n,i,j,k)$ be function of $\mathbf{y}_1^n\in\mathcal{I}^n$, and matrix $\mathbb{B}=(b_{ij})$ as defined by (B.0.8), then for all $i,j,k\in\mathcal{I}$ the following limit exists*

$$
\lim_{n\to\infty}\frac{1}{n}E_P\left[L_2(\mathbf{Y}_1^n,i,j,k)\Big|_{\mathbb{B}=\mathbb{I}}\right]
$$

and is equal to $V(i,j,k)$ as defined by (4.1.11). The series converges to the limit with rate $1/n$.

Proof. Let $\mathbf{y}_1^n \in \mathcal{I}^n$ be a fixed realization of random variable $\mathbf{Y}_1^n \in \mathcal{I}^n$. By substituting $\mathbb{B} = \mathbb{I}$, we simplify the term $L_2(\mathbf{y}_1^n, i, j, k)$

$$\begin{aligned}
L_2(\mathbf{y}_1^n, i, j, k)\big|_{\mathbb{B}=\mathbb{I}} &= \sum_{t \in J(j)} \frac{S_y(t,i)}{Q_\beta(\mathbf{y}_1^n)} \frac{S_y(t,k)}{Q_\beta(\mathbf{y}_1^n)}\Big|_{\mathbb{B}=\mathbb{I}} \\
&= \sum_{t \in J(j)} \frac{P\big((\mathbf{y}_1^{t-1}, i, \mathbf{y}_{t+1}^n)\big)}{P(\mathbf{y}_1^n)} \frac{P\big((\mathbf{y}_1^{t-1}, k, \mathbf{y}_{t+1}^n)\big)}{P(\mathbf{y}_1^n)} \\
&= \sum_{t \in J(j)} \frac{a_{y_{t-1},i} a_{i,y_{t+1}}}{a_{y_{t-1},j} a_{j,y_{t+1}}} \frac{a_{y_{t-1},k} a_{k,y_{t+1}}}{a_{y_{t-1},j} a_{j,y_{t+1}}}.
\end{aligned}$$

Now, we can rewrite the series $\frac{1}{n} E_P\big[L_2(\mathbf{Y}_1^n, i, j, k)\big|_{\mathbb{B}=\mathbb{I}}\big]$ to calculate the limit

$$\begin{aligned}
\frac{1}{n} E_P\big[L_2(\mathbf{Y}_1^n, i, j, k)\big|_{\mathbb{B}=\mathbb{I}}\big] &= \frac{1}{n} \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P(\mathbf{y}_1^n) L_2(\mathbf{y}_1^n, i, j, k)\big|_{\mathbb{B}=\mathbb{I}} = \\
&= \frac{1}{n} \sum_{t=1}^{n} \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P(\mathbf{y}_1^n) \bigg([t \in J_y(j)] \frac{a_{y_{t-1},i} a_{i,y_{t+1}}}{a_{y_{t-1},j} a_{j,y_{t+1}}} \frac{a_{y_{t-1},k} a_{k,y_{t+1}}}{a_{y_{t-1},j} a_{j,y_{t+1}}} \bigg) \\
&= \frac{1}{n} \sum_{t=2}^{n-1} \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P(\mathbf{y}_1^n) \bigg([t \in J_y(j)] \frac{a_{y_{t-1},i} a_{i,y_{t+1}}}{a_{y_{t-1},j} a_{j,y_{t+1}}} \frac{a_{y_{t-1},k} a_{k,y_{t+1}}}{a_{y_{t-1},j} a_{j,y_{t+1}}} \bigg) + \\
&\quad + \underbrace{\frac{1}{n} \sum_{\mathbf{y}_1^n \in \mathcal{I}^n} P(\mathbf{y}_1^n) \bigg([1 \in J_y(j)] \frac{\pi_i a_{i,y_2}}{\pi_j a_{j,y_2}} \frac{\pi_k a_{k,y_2}}{\pi_j a_{j,y_2}} + [n \in J_y(j)] \frac{a_{y_{n-1},i}}{a_{y_{n-1},j}} \frac{a_{y_{n-1},k}}{a_{y_{n-1},j}} \bigg) =}_{\triangleq f_n} \\
&= \frac{1}{n} \sum_{t=2}^{n-1} \sum_{z_1,z_2 \in \mathcal{I}} \frac{a_{z_1,i} a_{i,z_2}}{a_{z_1,j} a_{j,z_2}} \frac{a_{z_1,k} a_{k,z_2}}{a_{z_1,j} a_{j,z_2}} \sum_{y_1^n \in \mathcal{X}^n} P(\mathbf{y}_1^n) \Big[\mathbf{y}_{t-1}^{t+1} = (z_1, j, z_2) \Big] + f_n \\
&= \frac{1}{n} \sum_{t=2}^{n-1} \sum_{z_1,z_2 \in \mathcal{I}} \frac{a_{z_1,i} a_{i,z_2}}{a_{z_1,j} a_{j,z_2}} \frac{a_{z_1,k} a_{k,z_2}}{a_{z_1,j} a_{j,z_2}} P\Big(\mathbf{Y}_{t-1}^{t+1} = (z_1, j, z_2) \Big) + f_n \\
&= \frac{1}{n} \sum_{t=2}^{n-1} \sum_{z_1,z_2 \in \mathcal{I}} \frac{a_{z_1,i} a_{i,z_2}}{a_{z_1,j} a_{j,z_2}} \frac{a_{z_1,k} a_{k,z_2}}{a_{z_1,j} a_{j,z_2}} \pi_{z_1} a_{z_1,j} a_{j,z_2} + f_n \\
&= \frac{n-2}{n} \sum_{z_1,z_2 \in \mathcal{I}} \pi_{z_1} a_{z_1,i} a_{i,z_2} \frac{a_{z_1,k} a_{k,z_2}}{a_{z_1,j} a_{j,z_2}} + f_n.
\end{aligned}$$

Finally, we can calculate the limit

$$\begin{aligned}
V(i,j,k) &\triangleq \lim_{n \to \infty} \frac{1}{n} E_P\Big[L_2(\mathbf{Y}_1^n, i, j, k)\big|_{\mathbb{B}=\mathbb{I}} \Big] \\
&= \sum_{z_1,z_2 \in \mathcal{I}} \pi_{z_1} a_{z_1,i} a_{i,z_2} \frac{a_{z_1,k} a_{k,z_2}}{a_{z_1,j} a_{j,z_2}} \\
&= \bigg(\sum_{z \in \mathcal{I}} \pi_z a_{z,i} \frac{a_{z,k}}{a_{z,j}} \bigg) \bigg(\sum_{z \in \mathcal{I}} a_{i,z} \frac{a_{k,z}}{a_{j,z}} \bigg)
\end{aligned}$$

because $0 \leq f_n \leq C/n$ for some constant C and thus the rate of convergence is $O(1/n)$. The constant C depends only on elements of matrix $\mathbb{A}$. □

Bibliography

[1] R. Anderson. Stretching the limits of steganography. In R. J. Anderson, editor, *Information Hiding, 1st International Workshop*, volume 1174 of Lecture Notes in Computer Sc., pages 39–48, Cambridge, UK, May 30–June 1, 1996. Springer-Verlag, Berlin.

[2] E. Arikan. Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. *IEEE Transactions on Information Theory*, 55(7):3051–3073, July 2009.

[3] R. J. Barron, B. Chen, and G. W. Wornell. The duality between information embedding and source coding with side information and some applications. *IEEE Transactions on Information Theory*, 49(5):1159–1180, 2003.

[4] P. Bas, T. Filler, and T. Pevný. "Break Our Steganographic System" — the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. D. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Sc., Prague, Czech Republic, May 18–20, 2011. Submitted.

[5] P. Bas and T. Furon. BOWS-2. http://bows2.gipsa-lab.inpg.fr/BOWS2OrigEp3.tgz, July 2007.

[6] J. Bierbrauer. On Crandall's problem. Personal communication available from http://www.ws.binghamton.edu/fridrich/covcodes.pdf, 1998.

[7] J. Bierbrauer and J. Fridrich. Constructing good covering codes for applications in steganography. *LNCS Transactions on Data Hiding and Multimedia Security*, 4920:1–22, 2008.

[8] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Sc., pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[9] R. Böhme. *Advanced statistical steganalysis.* Springer-Verlag, Heidleberg, 2010.

[10] R. Böhme and A. Westfeld. Breaking Cauchy model-based JPEG steganography with first order statistics. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *Computer Security - ESORICS 2004. Proceedings 9th European Symposium on Research in Computer Security*, volume 3193 of Lecture Notes in Computer Sc., pages 125–140, Sophia Antipolis, France, September 13–15, 2004. Springer, Berlin.

[11] C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of Lecture Notes in Computer Sc., pages 306–318, Portland, OR, April 14–17, 1998. Springer-Verlag, New York.

[12] A.R. Calderbank, P.C. Fishburn, and A. Rabinovich. Covering properties of convolutional codes and associated lattices. *IEEE Transactions on Information Theory*, 41(3):732–746, 1995.

[13] G. Cancelli and M. Barni. MPSteg-color: A new steganographic technique for color images. In *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Sc., pages 1–15, Saint Malo, France, June 11–13, 2007.

[14] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[15] C. Chen and Y. Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, 2008. ISCAS 2008. IEEE Intern. Symposium on*, pages 3029–3032, May 2008.

[16] P. Comesana and F. Pérez-Gonzáles. On the capacity of stegosystems. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[17] T. M. Cover and J. A. Thomas. *Elements of Information Theory.* New York: John Wiley & Sons, Inc., 2006.

[18] M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, June 1996.

[19] R. Crandall. Some notes on steganography. *Steganography Mailing List*, available from http://os.inf.tu-dresden.de/~westfeld/crandall.pdf, 1998.

[20] J. L. Doob. *Stochastic processes.* Wiley, New York, 1st edition, 1953.

[21] S. Dumitrescu, X. Wu, and Z. Wang. Detection of LSB steganography via Sample Pairs Analysis. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Sc., pages 355–372, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[22] Y. Ephraim and N. Merhav. Hidden Markov processes. *Information Theory, IEEE Transactions on*, 48(6):1518–1569, June 2002.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. Software available at http://www.csie.ntu.edu.tw/~cjlin/liblinear.

[24] T. Filler and J. Fridrich. Complete characterization of perfectly secure stegosystems with mutually independent embedding. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan, April 19–24, 2009.

[25] T. Filler and J. Fridrich. Fisher information determines capacity of ϵ-secure steganography. In S. Katzenbeisser and A.-R. Sadeghi, editors, *Information Hiding, 11th International Workshop*, volume 5806 of Lecture Notes in Computer Sc., pages 31–47, Darmstadt, Germany, June 7–10, 2009. Springer-Verlag, New York.

[26] T. Filler and J. Fridrich. Wet ZZW construction for steganography. In *First IEEE International Workshop on Information Forensics and Security*, London, UK, December 6–9 2009.

[27] T. Filler and J. Fridrich. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4):705–720, December 2010.

[28] T. Filler and J. Fridrich. Using non-binary embedding operation to minimize additive distortion functions in steganography. In *Second IEEE International Workshop on Information Forensics and Security*, Seattle, WA, 2010.

[29] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using Syndrome-Trellis Codes. *IEEE Transactions on Information Forensics and Security*, 2010. Submitted. See http://dde.binghamton.edu/filler/publications.php.

[30] T. Filler, J. Judas, and J. Fridrich. Minimizing embedding impact in steganography using trellis-coded quantization. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 05–01–05–14, San Jose, CA, January 17–21, 2010.

[31] T. Filler and A. D. Ker. Fisher information: Two approaches. Rump session talk at 11th Information Hiding, Darmstadt, Germany, June 7-10, 2009. http://dde.binghamton.edu/filler/pdf/FillKer09ihw-fisher-comparison.pdf.

[32] T. Filler, A. D. Ker, and J. Fridrich. The Square Root Law of steganographic capacity for Markov covers. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 08 1–08 11, San Jose, CA, January 18–21, 2009.

[33] E. Franz. Embedding considering dependencies between pixels. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages D 1–D 12, San Jose, CA, January 27–31, 2008.

[34] E. Franz, S. Rönisch, and R. Bartel. Improved embedding based on a set of cover images. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 141–150, Princeton, NJ, September 7–8, 2009.

[35] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Sc., pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.

[36] J. Fridrich. Asymptotic behavior of the ZZW embedding construction. *IEEE Transactions on Information Forensics and Security*, 4(1):151–153, March 2009.

[37] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.

[38] J. Fridrich and T. Filler. Practical methods for minimizing embedding impact in steganography. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 02–03, San Jose, CA, January 29–February 1, 2007.

[39] J. Fridrich and M. Goljan. Digital image steganography using stochastic modulation. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V*, volume 5020, pages 191–202, Santa Clara, CA, January 21–24, 2003.

[40] J. Fridrich, M. Goljan, and R. Du. Reliable detection of LSB steganography in grayscale and color images. In J. Dittmann, K. Nahrstedt, and P. Wohlmacher, editors, *Proceedings of the ACM, Special Session on Multimedia Security and Watermarking*, pages 27–30, Ottawa, Canada, October 5, 2001.

[41] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography using wet paper codes. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 6th ACM Multimedia & Security Workshop*, pages 4–15, Magdeburg, Germany, September 20–21, 2004.

[42] J. Fridrich, M. Goljan, and D. Soukal. Efficient wet paper codes. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, Lecture Notes in Computer Sc., pages 204–218, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[43] J. Fridrich, M. Goljan, and D. Soukal. Perturbed quantization steganography. *ACM Multimedia System Journal*, 11(2):98–107, 2005.

[44] J. Fridrich, M. Goljan, and D. Soukal. Wet paper codes with improved embedding efficiency. *IEEE Transactions on Information Forensics and Security*, 1(1):102–110, 2006.

[45] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk. Writing on wet paper. In *IEEE Transactions on Signal Processing, Special Issue on Media Security*, volume 53, pages 3923–3935, October 2005. (journal version).

[46] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk. Writing on wet paper. In *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 328–340, San Jose, CA, January 16–20, 2005.

[47] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[48] J. Fridrich and D. Soukal. Matrix embedding for large payloads. *IEEE Transactions on Information Forensics and Security*, 1(3):390–394, 2006.

[49] K. Fukumizu, F. R. Bach, and M. I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.

[50] S. I. Gel'fand and M. S. Pinsker. Coding for channel with random parameters. *Problems of Control and Information Theory*, 9(1):19–31, 1980.

[51] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[52] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 1–13, San Jose, CA, January 16–19, 2006.

[53] G. Golub and C. Van Loan. *Matrix Computations.* The John Hopkins University Press, 1st edition, 1983.

[54] A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, Cambridge, MA, 2007.

[55] G. Gul, A. E. Dirik, and I. Avcibas. Steganalytic features for JPEG compression-based perturbed quantization. *IEEE Signal Processing Letters*, 14(3):205–208, March 2007.

[56] W. Hall and M. Newell. The mean value theorem for vector valued functions: a simple proof. *Mathematics Magazine*, 52(3):157–158, May 1979.

[57] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer-Verlag, Heidleberg, 2nd edition, 2009.

[58] I. Hen and N. Merhav. On the error exponent of trellis source coding. *IEEE Transactions on Information Theory*, 51(11):3734–3741, 2005.

[59] S. Hetzl and P. Mutzel. A graph-theoretic approach to steganography. In *Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005*, volume 3677 of Lecture Notes in Computer Sc., pages 119–128, Salzburg, Austria, September 19–21, 2005.

[60] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 408–415. ACM, 2008.

[61] S. M. Kay. *Fundamentals of Statistical Signal Processing, Volume II: Detection Theory*, volume II. Upper Saddle River, NJ: Prentice Hall, 1998.

[62] A. D. Ker. Improved detection of LSB steganography in grayscale images. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Sc., pages 97–115, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[63] A. D. Ker. A general framework for structural analysis of LSB replacement. In M. Barni, J. Herrera, S. Katzenbeisser, and F. Pérez-González, editors, *Information Hiding, 7th International Workshop*, volume 3727 of Lecture Notes in Computer Sc., pages 296–311, Barcelona, Spain, June 6–8, 2005. Springer-Verlag, Berlin.

[64] A. D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, June 2005.

[65] A. D. Ker. Batch steganography and pooled steganalysis. In *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Sc., pages 265–281, Alexandria, VA, July 10–12, 2006.

[66] A. D. Ker. Fourth-order structural steganalysis and analysis of cover assumptions. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 25–38, San Jose, CA, January 16–19, 2006.

[67] A. D. Ker. A capacity result for batch steganography. *IEEE Signal Processing Letters*, 14(8):525–528, 2007.

[68] A. D. Ker. A fusion of maximal likelihood and structural steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Sc., pages 204–219, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[69] A. D. Ker. Steganalysis of embedding in two least significant bits. *IEEE Transactions on Information Forensics and Security*, 2:46–54, 2007.

[70] A. D. Ker. Perturbation hiding and the batch steganography problem. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Sc., pages 45–59, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.

[71] A. D. Ker. Estimating steganographic Fisher information in real images. In S. Katzenbeisser and A.-R. Sadeghi, editors, *Information Hiding, 11th International Workshop*, volume 5806 of Lecture Notes in Computer Sc., pages 73–88, Darmstadt, Germany, June 7–10, 2009. Springer-Verlag, New York.

[72] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–5 17, San Jose, CA, January 27–31, 2008.

[73] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[74] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Sc., pages 314–327, Alexandria, VA, July 10–12, 2006.

[75] G. Kipper. *Investigator's Guide to Steganography.* Boca Raton, FL: CRC Press, 2004.

[76] J. Kodovský and J. Fridrich. On completeness of feature spaces in blind steganalysis. In *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 123–132, Oxford, UK, September 22–23, 2008.

[77] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.

[78] J. Kodovský and J. Fridrich. Quantitative structural steganalysis of jsteg. *IEEE Transactions on Information Forensics and Security*, 5(4):681–693, 2010.

[79] J. Kodovský, T. Pevný, and J. Fridrich. Modern steganalysis can detect YASS. In *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7541, pages 02–01–02–11, San Jose, CA, January 17–21, 2010.

[80] S. B. Korada and R. L. Urbanke. Polar codes are optimal for lossy source coding. *IEEE Transactions on Information Theory*, 56(4):1751–1768, April 2010.

[81] K. Lee and A. Westfeld. Generalized category attack – improving histogram-based attack on JPEG LSB embedding. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Sc., pages 378–392, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[82] K. Lee, A. Westfeld, and S. Lee. Category attack for LSB embedding of JPEG images. In Y.-Q. Shi, B. Jeon, Y.Q. Shi, and B. Jeon, editors, *Digital Watermarking, 5th International Workshop*, volume 4283 of Lecture Notes in Computer Sc., pages 35–48, Jeju Island, Korea, November 8–10, 2006. Springer-Verlag, Berlin.

[83] D. MacKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003. Can be obtained from http://www.inference.phy.cam.ac.uk/mackay/itila/.

[84] D. J. C. McKay. *Information Theory, Inference, and Learning Algorithms.* Cambridge: Cambridge University Press, 2003.

[85] L. Mevel and L. Finesso. Asymptotical statistics of misspecified hidden Markov models. *Automatic Control, IEEE Transactions on*, 49(7):1123–1132, July 2004.

[86] P. Moulin and R. Koetter. Data-hiding codes. *Proceedings of the IEEE*, 93(12):2083–2126, 2005.

[87] P. Moulin and Y. Wang. New results on steganographic capacity. In *Proceedings of the Conference on Information Sciences and Systems, CISS*, Princeton, NJ, March 17–19, 2004.

[88] R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, University of Toronto, September 25 1993.

[89] J. Nocedal and S. Wright. *Numerical Optimization.* Springer, 2nd edition, 2006.

[90] A. Orsdemir, O. Altun, G. Sharma, and M. Bocko. Steganalysis-aware steganography: Statistical indistinguishability despite high distortion. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, San Jose, CA, January 27–31, 2008.

[91] T. Pevný. Detecting messages of unknown length. In N. D. Memon, E. J. Delp, A. Alattar, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XII*, volume 7880, San Jose, CA, January 17–21, 2010.

[92] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. In *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 75–84, Princeton, NJ, September 7–8, 2009.

[93] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, June 2010.

[94] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In P. W. L. Fong, R. Böhme, and R. Safavi-Naini, editors, *Information Hiding, 12th International Conference*, Lecture Notes in Computer Sc., pages 161–177, Calgary, Canada, June 28–30, 2010.

[95] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29–February 1, 2007.

[96] T. Pevný and J. Fridrich. Benchmarking for steganography. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Sc., pages 251–267, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.

[97] T. Pevný, J. Fridrich, and A. D. Ker. From blind to quantitative steganalysis. In N. D. Memon, E. J. Delp, P. W. Wong, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Security and Forensics of Multimedia XI*, volume 7254, pages 0C 1–0C 14, San Jose, CA, January 18–21, 2009.

[98] S.S. Pradhan and K. Ramchandran. Distributed source coding using syndromes (discus): design and construction. *IEEE Transactions on Information Theory*, 49(3):626–643, 2003.

[99] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[100] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, January 2009.

[101] B.Y. Ryabko and D.B. Ryabko. Asymptotically optimal perfect steganographic systems. *Problems of Information Transmission*, 45(2):184–190, 2009.

[102] V. Sachnev, H. J. Kim, and R. Zhang. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 131–140, Princeton, NJ, Sept. 2009.

[103] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[104] A. Sarkar, L. Nataraj, B. S. Manjunath, and U. Madhow. Estimation of optimum coding redundancy and frequency domain analysis of attacks for YASS - a randomized block based hiding scheme. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 1292–1295, 2008.

[105] A. Sarkar, K. Solanki, U. Madhow, and B. S. Manjunath. Secure steganography: Statistical restoration of the second order dependencies for improved security. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE Intern. Conf. on*, volume 2, pages II–277–II–280, April 15–20, 2007.

[106] D. Schönfeld and A. Winkler. Embedding with syndrome coding based on BCH codes. In S. Voloshynovskiy, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 8th ACM Multimedia & Security Workshop*, pages 214–223, Geneva, Switzerland, September 26–27, 2006.

[107] N. Shachtman. Fbi: Spies hid secret messages on public websites. http://www.wired.com/dangerroom/2010/06/alleged-spies-hid-secret-messages-on-public-websites/, June 2010.

[108] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.*, 4:142–163, 1959.

[109] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Sc., pages 249–264, Alexandria, VA, July 10–12, 2006.

[110] V. Sidorenko and V. Zyablov. Decoding of convolutional codes using a syndrome trellis. *IEEE Transactions on Information Theory*, 40(5):1663–1666, 1994.

[111] M. Sidorov. Hidden Markov models and steganalysis. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 6th ACM Multimedia & Security Workshop*, pages 63–67, Magdeburg, Germany, September 20–21, 2004.

[112] G. J. Simmons. The prisoner's problem and the subliminal channel. In D. Chaum, editor, *Advances in Cryptology, CRYPTO '83*, pages 51–67, Santa Barbara, CA, August 22–24, 1983. New York: Plenum Press.

[113] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Sc., pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.

[114] K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Provably secure steganography: Achieving zero K–L divergence using statistical restoration. In *Image Processing, 2006 IEEE International Conference on*, pages 125–128, October 8–11, 2006.

[115] D. Soukal, J. Fridrich, and M. Goljan. Maximum likelihood estimation of secret message length embedded using $\pm k$ steganography in spatial domain. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 595–606, San Jose, CA, January 16–20, 2005.

[116] D. Upham. Steganographic algorithm JSteg. http://zooid.org/ paul/crypto/jsteg.

[117] M. van Dijk and F. Willems. Embedding information in grayscale images. In *Proceedings of the 22nd Symposium on Information and Communication Theory*, pages 147–154, Enschede, The Netherlands, May 15–16, 2001.

[118] S. Verdú and T. Weissman. Erasure entropy. In *Proc. of ISIT*, Seattle, WA, July 9–14, 2006.

[119] S. Verdú and T. Weissman. The information lost in erasures. *IEEE Transactions on Information Theory*, 54(11):5030–5058, November 2008.

[120] A. Viterbi and J. Omura. Trellis encoding of memoryless discrete-time sources with a fidelity criterion. *IEEE Transactions on Information Theory*, 20(3):325–332, May 1974.

[121] C. Wang, X. Li, B. Yang, X. Lu, and C. Liu. A content-adaptive approach for reducing embedding impact in steganography. In *Proceedings IEEE, International Conference on Acoustics, Speech, and Signal Processing*, pages 1762–1765, March 2010.

[122] C. K. Wang, G. Doërr, and I. Cox. Trellis coded modulation to improve dirty paper trellis watermarking. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, page 6505 0G, San Jose, CA, January 29–February 1, 2007.

[123] F. Wang and D. P. Landau. Determining the density of states for classical statistical models: A random walk algorithm to produce a flat histogram. *Phys. Rev. E*, 64(5):056101, 2001. arXiv:cond-mat/0107006v1.

[124] Y. Wang and P. Moulin. Perfectly secure steganography: Capacity, error exponents, and code constructions. *IEEE Transactions on Information Theory, Special Issue on Security*, 55(6):2706–2722, June 2008.

[125] A. Westfeld. High capacity despite better steganalysis (F5 – a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of Lecture Notes in Computer Sc., pages 289–302, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[126] A. Westfeld and R. Böhme. Exploiting preserved statistics for steganalysis. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Sc., pages 82–96, Toronto, Canada, May 23–25, 2004. Springer-Verlag, Berlin.

[127] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 2003.

[128] R. Zhang, V. Sachnev, and H. J. Kim. Fast BCH syndrome coding for steganography. In S. Katzenbeisser and A.-R. Sadeghi, editors, *Information Hiding, 11th International Workshop*, volume 5806 of Lecture Notes in Computer Sc., pages 31–47, Darmstadt, Germany, June 7–10, 2009. Springer-Verlag, New York.

[129] W. Zhang and X. Wang. Generalization of the ZZW embedding construction for steganography. *Information Forensics and Security, IEEE Transactions on*, 4(3):564–569, September 2009.

[130] W. Zhang, X. Zhang, and S. Wang. Maximizing steganographic embedding efficiency by combining Hamming codes and wet paper codes. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Sc., pages 60–71, Santa Barbara, CA, June 19–21, 2008. Springer-Verlag, New York.

[131] W. Zhang and X. Zhu. Improving the embedding efficiency of wet paper codes by paper folding. *IEEE Signal Processing Letters*, 16(9):794–797, September 2009.

[132] X. Zhang, W. Zhang, and S. Wang. Efficient double-layered steganographic embedding. *Electronics Letters*, 43:482–483, April 2007.