Design of Robust Blind Detector with Application to Watermarking

by

Ernest Sopuru Anamalu

A Thesis
submitted to the Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements for the degree of
Master of Science

Department of Electrical and Computer Engineering
University of Manitoba
Winnipeg, Manitoba, R3T 5V6 Canada

Dedicated to the loving memory of my late father Gabriel
and
my mother Veronica.

Abstract

One of the difficult issues in detection theory is to design a robust detector that takes into account the actual distribution of the original data. The most commonly used statistical detection model for blind detection is Gaussian distribution. Specifically, linear correlation is an optimal detection method in the presence of Gaussian distributed features. This has been found to be sub-optimal detection metric when density deviates completely from Gaussian distributions. Hence, we formulate a detection algorithm that enhances detection probability by exploiting the true characterises of the original data. To understand the underlying distribution function of data, we employed the estimation techniques such as parametric model called approximated density ratio logistic regression model and semiparameric estimations. Semiparametric model has the advantages of yielding density ratios as well as individual densities. Both methods are applicable to signals such as watermark embedded in spatial domain and outperform the conventional linear correlation non-Gaussian distributed.

Keywords: Signal detections, parametric and nonparametric estimations, K-means, expectation maximization, maximum likelihood estimations, density ratio estimation, Gaussian mixture model, Logistic regression model.

Acknowledgements

The completion of this thesis would not have been possible without acknowledging the enormous help I received from the following people.

- My academic supervisor, Professor Dr. Miroslaw Pawlak for accepting me as his graduate student. His deep knowledge and exceptional analytical problem solving skills immensely helped me a lot throughout my MSc program. Also, I would like thank him for his support, advice, patience and detailed examination and correction of my thesis.
- I am grateful for the financial support from Faculty of Graduate Studies.
- I would also like to express my gratitude to Professor Pradeepa Yahampath for his help and suggestions on many occasions I ran into problem.
- Finally, I would like to thank my friends here in Winnipeg, my family for their uptmost unfading love and support.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 The Concept of Watermark Detection

Watermarking is a type of digital communication where the aim is to transmit a watermark message reliably over a noisy channel. It is mainly used for different security reasons in applications such as images, copyright protection, fingerprinting, device control and data authentication [1], and has recently been employed to provide sensory data authentication in wireless sensor networks (WSNs) [2].

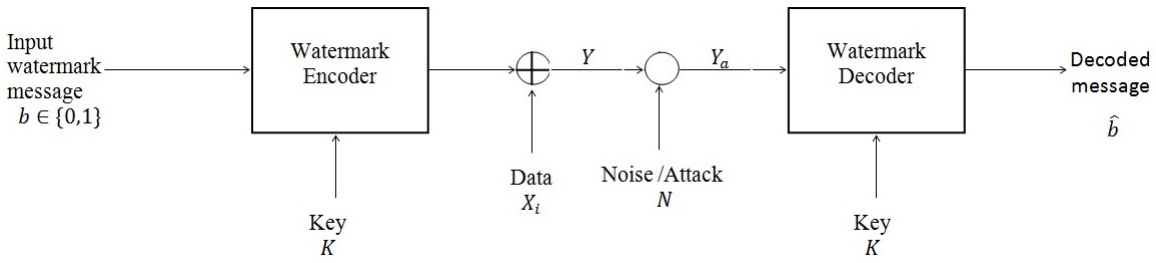


Figure 1.1: General model of watermarking process.

In Figure (1.1), we present a typical watermarking process where the input message to be embedded as watermark signal is b, K is the secret key usually shared by the embedder and decoder. The watermark signal is either additively or multiplicatively inserted into the original data X_i such as images or sensory data. Sometimes, the watermarked data can encounter either intentional or unintentional attacks. Finally, a watermark decoder is applied to decode the embedded signal. It is our intention to state that decoder and

detector are used interchangeably throughout this thesis. Furthermore, in most applications it is required that the original data and watermarked version be perceptually similar, meaning that addition of watermark should not affect the original data significantly.
In general, most researchers have focused on different strategies to embed watermark in a cover work such as image data so that watermarked work is robust to some well known attacks. Such attacks like lossy compression, cropping, lowpass filtering and additive noise can easily be dealt with by embedding watermark in spread spectrum like fashion [1][3]. Others have focused on robustness to geometric distortions such as rotation, scaling and flipping caused to the watermarked image by applying Zernike moments [4][5]. Watermark detection is usually associated with the type of embedding algorithm used. This means that if watermark is embedded in frequency domain or spatial domain, then to detect the watermark, the watermarked data has to also be transformed in frequency or spatial domain. In all these watermark embedding strategies developed, the detection algorithms used are mostly sub-optimal [6].

Developing an optimal watermark detection method is very difficult task. There are two common factors that contribute to the difficulties while developing watermark detection strategy. The first is that a watermarked data may undergo series of unpredictable attacks that affects the signal detection. Secondly, the probability density function (PDF) of the original data may not be available during watermark detection especially in blind watermark detection [6]. These two factors, attack and data features are mainly the sources of noise to the watermark signal. Generally, watermarks embedded in spatial domain of the original data have counterpart detection algorithm as linear correlation (LC). Linear correlation is always optimal for Gaussian distributed features [1], but since most of image data are not Gaussian distributed as we shall see, applying LC as detection metric may not perform optimally [6] . Therefore, the general assumption of using central limit theory (CLT) to approximate the distribution of features of data may not work very well since according to CLT, the data has to be very large for this to hold . This is not always the case in watermark applications.

Also, for watermark embedded into frequency domain host features such as discrete cosine transform (DCT) and discrete wavelet transforms (DWT), the generalized Gaussian distribution (GGD) is proven to be better statistical model for the host features during detection [6]. The frequency domain host features are not approximated by the

Gaussian distribution because the PDF of frequency domain host features is slightly deviated from the Gaussian distribution. In fact, for DCT features, it is proven that the statistical distribution that best fits the model is Laplacian distribution [7]. The Gaussian and Laplacian distributions are the special cases of generalized Gaussian distributions. For details on frequency domain detection metric using GGD see [8][9] [10]. We skipped the design of new frequency domain detector in this thesis since GGD has adaptational and flexible properties that enable it to fit the unknown frquency domain characteristics.

1.2 Description of the Techniques

To overcome these drawbacks, it is proven that better modelling of the host features PDF during watermark detection improves the detection performance [11]. We propose to estimate the underlying density of data with the following density estimation methods.

- **Gaussian mixture model (GMM)**. GMM is a parametric density function that comprises of two or more Gaussian distributions added up to form a statistical density, and it is termed multimodal density. Usually, it has a finite number of Gaussian densities which is totally characterized by their estimated parameters. As we shall see, most image data are distributed according to this model GMM, so estimating the parameters of this PDF model accurately can improve the probability of watermark detection. One of the advantages of GMM distribution is that computational load required to estimate the density compared to other methods such as nonparametric method is much lower. The disadvantage of this estimation method is that if number of components is not estimated correctly, the original PDF may be inaccurately estimated [12].

- **Density ratio estimation (Logistic Regression Model)**. In watermark detection design, likelihood ratio test (LRT) is usually compared to a predetermined threshold so as to determine the presence of watermark in a received data. The universal method employed to determine the LRT is simply to estimate the watermarked and non-watermark densities separately and then take their ratio. It is proven that sometimes estimating individual densities and taking their ratio is more difficult and increases the error probability [13][14]. Moreover, in LC detection individual densitites need not be estimated since central limit theory is invoked. Therefore, to avoid density estimation, parametric logistic regression model based

on exponential family is employed which directly models the density ratio of the two distributions [15].

- **Semiparametric and Nonparametric Estimation**. In some applications, the distribution of the original data is not known a priori during detection so assuming univariate Gaussian may be wrong assumption and costly. Therefore, the obvious choice is to estimate the underlying density either nonparametrically [16] or exploit the compromise of semiparametric approach offers. The advantage of nonparametric based detector is that modelling error is eliminated and with semiparametric approach the computational load maybe reduced [17].

- **Informed and Blind watermark detection**. Generally, there are two types of watermark detection schemes which can be classified as informed and blind detection. The former requires original PDF of the image during detection process whereas the later does not require this PDF. In some applications where the original data is available during detection, the informed detection is appropriate. The original data can easily be subtracted from the received data to obtain the embedded watermark [1]. In other applications, it is not always possible to have original data during detection, so the underlying data has to be estimated before watermark detection is applied. In this thesis, the blind watermarking assumption is made throughout unless otherwise stated.

Chapter 2

Review of Binary Hypothesis Testing of Watermarked Data

The basic watermark detection problem is to decide whether watermark is present or absent in the received data. This is called binary hypothesis testing because we are deciding between two hypotheses[18][19][20].

$$Decision = \begin{cases} H_0 : Watermark\ \ is\ \ absent \\ H_1 : Watermark\ \ is\ \ present. \end{cases} \tag{2.1}$$

The objective is to use receive watermarked data to make correct decisions about the existence of watermark signal. Sometimes we encounter errors due to bad decisions, and try minimizing these errors as possible as we can thereby improving watermark detection probability. The four possible conditional probabilities of decision criteria encountered in binary hypotheses testing are

1. $P(Decide\ \ H_0|H_0)$ is the probability that correct decision is made when watermark is absent.
2. $P(Decide\ \ H_1|H_1)$ is the probability that correct decision is made when watermark is present.
3. $P(Decide\ \ H_1|H_0)$ is the probability of wrong decision when watermark is absent.
4. $P(Decide\ \ H_0|H_1)$ is the probability of wrong decision when watermark is present.

The conditional probabilities $P(Decide\ \ H_1|H_0)$ and $P(Decide\ \ H_0|H_1)$ are called the probability of false alarm P_{fa} and probability of miss detection P_m respectively. Therefore, the objective is to minimize these probabilities so that probability of detection $P(Decide\ \ H_1|H_1)$ is maximized. In some applications, it is more severe or catastrophic to have P_m occur regularly than P_{fa}. For example, an image watermarked for copy right control would be more catastrophic when detector fails to detect watermark given that watermark is actually embedded than when P_{fa} occurs. As an example of a simple binary hypothesis testing of watermarked data, if we assume that a set of data has been watermarked by simply adding watermark to the original data as.

$$Y_i = W_i + d_i. \tag{2.2}$$

The original data is given as i.i.d of Gaussian distribution $d_i \backsim \mathcal{N}(\mu_d, \sigma_d^2)$ and watermark signal is $W_i \backsim \mathcal{N}(\mu_w, \sigma_w^2)$, where $i = 1, ..., N$. Hence, $Y_i \backsim \mathcal{N}(\mu_d + \mu_w, \sigma_d^2 + \sigma_w^2)$ is the convolution of both random independent variables which takes continuous value over domain $\mathcal{Y}$. The assumption in this case is that data feature is the only source of noise encountered. This is not the case in many applications where other sources of noise such as lossy compression or additive white Gaussian noise (AWGN) may also be encountered. Based on the observed values Y, we decide the presence or absent of watermark as

$$Decide = \begin{cases} H_0 : Y \backsim g(Y) \\ H_1 : Y \backsim f(Y). \end{cases} \tag{2.3}$$

.

In equation (2.3), there are two cases that possibly led to hypothesis H_0, either that the original data is not watermarked or that a wrong watermark is embedded instead of required watermark W_i. If we let $\mathcal{Y}$ domain to be the set of possible observations, then

$$\mathcal{Y} = \mathcal{Y}_{H_0} \cup \mathcal{Y}_{H_1}. \tag{2.4}$$

We accept the hypotheses H_0, if $Y \in \mathcal{Y}_{H_0}$ and accept hypothesis H_1, if $Y \in \mathcal{Y}_{H_1}$.The false alarm probability P_{fa} and probability of miss detection in this case are derived respectively as in [19]

$$\begin{aligned} P_{fa} &= P(Y \in \mathcal{Y}_{H_1}|H_0) = \int_{\mathcal{Y}_{H_1}} g(Y)dY \\ P_m &= P(Y \in \mathcal{Y}_{H_0}|H_1) = \int_{\mathcal{Y}_{H_0}} f(Y)dY. \end{aligned} \tag{2.5}$$

The two most common detection approaches in hypotheses testing are the Bayesian and Neyman-Pearson approaches[18][19][20]. Usage of any of the approach depends on the application intended for. For instance, it is common practice to use Bayesian approach for pattern recognition applications and Neyman-Pearson approach in radar detection applications [19]. In watermarking system, both approaches can be used, but their usage depends on the attack watermarked data undergoes and our willingness to assign cost and prior probabilities to the hypotheses.

2.1 Bayes Theory of Hypothesis Testing of Watermark

In Bayes theory of hypothesis, prior probabilities of hypotheses are assigned during detection of watermark. These probabilities express the knowledge of likelihood occurring of each hypothesis. The objective in Bayes decision theory is to minimize the risk which is defined as the average cost assigns to the decision made during detection, and it is given as [19]

$$R = C_{00}(1 - P_{fa})P(H_0) + C_{10}P_{fa}P(H_0) + C_{11}(1 - P_m)P(H_1) + C_{01}P_mP(H_1), \quad (2.6)$$

where $P(H_0)$ and $P(H_1)$ are the a priori probabilities and $C_{i,j}$, $i, j = \{0, 1\}$ are the cost functions assigned due to a particular decision. The assumption mostly made when using Bayes criterion in watermark application is that the cost of making a wrong decision during detection is always greater than the cost of making a correct decision. Hence we made wrong decisions when

$$\begin{cases} P_{fa}, & C_{10} > C_{00} \\ P_m, & C_{01} > C_{11}. \end{cases} \quad (2.7)$$

Therefore, to minimize R, the decision boundary $\mathcal{Y}_{H_1}$ has to be selected appropriately. Substituting the integrations of (2.5) into (2.6) and selecting the values that are functions of $\mathcal{Y}_{H_1}$ gives

$$P(H_0)(C_{10} - C_{11})f(yH_0) - P(H_1)(C_{01} - C_{00})f(yH_1) < 0. \quad (2.8)$$

Re-arranging equation (2.8) gives the Bayes risk detector [19] given as

$$\frac{f(Y)}{g(Y)} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{P(H_0)(C_{10} - C_{11})}{P(H_1)(C_{01} - C_{00})}, \quad (2.9)$$

where $L(Y) = \frac{f(Y)}{g(Y)}$ is the one dimensional random variable known as likelihood ratio test (LRT) and $\gamma = \frac{P(H_0)(C_{10}-C_{11})}{P(H_1)(C_{01}-C_{00})}$ is the threshold of the LRT. Finally, the Bayes detection equation is written as

$$L(Y) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma. \tag{2.10}$$

Sometimes, it is convenient to compute the natural logarithm of (2.10) since natural logarithm is a monotonic function

$$L(Y)^{'} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma^{'}, \tag{2.11}$$

where $L(Y)^{'} = \{\log L(Y)\}$ and $\gamma^{'} = \log\{\gamma\}$. If we let $C_{11} = C_{00} = 0$ and $C_{10} = C_{01} = 1$ such that γ is computed as the ratio of prior probabilities, we obtain the error probability P_e from equation (2.6) as

$$\begin{aligned} P_e &= P_{fa}P(H_0) + P_m P(H_1) \\ &= P(L(Y) \geq \gamma | H_0)P(H_0) + P(L(Y) < \gamma | H_1)P(H_1). \end{aligned} \tag{2.12}$$

When the two a priori probability hypotheses $P(H_0) = P(H_1) = 0.5$, the threshold $\log\{\gamma\} = 0$. This is illustrated in Figure 2.1. This procedure is called *maximum likelihood detection (ML)* and it is from the fact that equation (2.9) reduces to

$$f(Y) > g(Y). \tag{2.13}$$

That is, we choose the hypothesis with larger conditional probability density function.

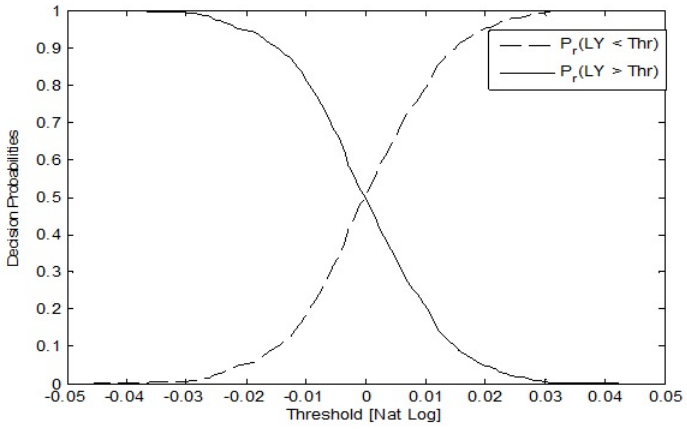


Figure 2.1: Maximum likelihood ratio decision plot.

2.2 Neyman-Pearson Hypothesis Testing of Watermark

Neyman-Pearson (NP) testing is widely used as a detection criterion in watermark applications because unlike Bayes approach, it does not rely on the cost function C_{ij} and prior hypothesis probabilities $P(H_i)$ or $P(H_j)$. This is helpful in situations where neither these costs nor probabilities are known and difficult to estimate. Also, another significant difference between Bayes and NP tests is how the threshold γ is determined. In Bayes criterion, costs and prior probabilities must be determined to obtain the threshold, whereas in NP criterion, the threshold is obtained in such a way that a pre-chosen false alarm probability is achieved. The NP tests has the advantage that it allows detector to fixed false alarm whilst maximizing the probability of detection $P_D = (1 - P_m)$ [16]. This simply means that during detection that the detector keeps P_{fa} below the constraint regardless of probability of missing the watermark P_m. Mathematically

$$P(Y \in \mathcal{Y}_{H_1}|H_0) \leq \alpha. \tag{2.14}$$

We can also re-write and evaluate equation (2.14) as

$$\int_{\mathcal{Y}_{H_1}} g(Y)dY \leq \alpha. \tag{2.15}$$

The likelihood ratio test (LRT) for NP is written as

$$L(Y) = \frac{f(Y)}{g(Y)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma. \tag{2.16}$$

In logarithm form, we have

$$L(Y)^{'} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma^{'}, \tag{2.17}$$

where $L(Y)^{'} = \log\{L(Y)\}$ and $\gamma^{'} = \log\{\gamma\}$. The probability of false alarm and probability of miss detection are respectively written as

$$P_{fa} = P(L(Y)^{'} > \gamma^{'}|H_0) = \int_{\gamma^{'}}^{+\infty} g(L(Y)^{'})dL,$$

$$P_M = P(L(Y)^{'} < \gamma^{'}|H_1) = \int_{-\infty}^{\gamma^{'}} f(L(Y)^{'})dL. \tag{2.18}$$

For Gaussian distributed features in watermarking applications, it is easier to apply NP test as the detection algorithm because test statistics can be obtained in a close form. In other features that are not Gaussian, test statistics might be challenging to obtain.

2.3 Optimal Detection of an Additive Watermark

Watermark detection in the presence of an additive white Gaussian noise (AWGN) is an example detection considered in this thesis [6]. It is simply adding a watermark signal to the original data as follow

$$Y_i = d_i + \alpha W_i + n_i, \tag{2.19}$$

where d_i is the original data features, αW_i is the watermark with scaling factor α and n_i is the uncertainty AWGN attack introduced in the channel. The original features and AWGN attack are the two sources of noise experienced by the watermark signal, and we may assume that both are drawn from the Gaussian distribution which is not always the case. If we let noise sources to be $X_i = d_i + n_i$, then (2.19) can be re-written as

$$Y_i = X_i + \alpha W_i. \tag{2.20}$$

It is worth noting that original data features such as natural image d_i are not necessary Gaussian distributed [21][22], this is the main goal of this thesis. That is, to exploit the actual distribution of the original data so as to improve the watermark detection.

2.3.1 Spatial Domain Features Distribution

In many watermark applications, the watermark is directly embedded in the spatial domain of the original data. This is the simplest embedding method. It is not always the optimal embedding strategy as slight modification to the watermarked data can remove the embedded watermark. In this type of embedding, the underlying data is mostly modelled as Gaussian distribution. Though, this is not always the case, but for the time being, we assume this to be true. It is proven [1] that the optimal detection metric for the watermark embedded in Gaussian distributed features is linear correlation between the received data and the watermark. The mathematical illustration of this assumption using Neyman Pearson criterion which can also be found in [6] is given as

$$L(Y) = \frac{\prod_{i=1}^{N} f(Y_i)}{\prod_{i=1}^{N} g(Y_i)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma. \tag{2.21}$$

The random variables Y_i coming from conditonal PDFs $f(Y_i|H_k)$, $k = \{0, 1\}$ are independent and identically distributed (iid) due to the fact that $X_i = d_i + n_i$ are also iid, then we have

$$\prod_{i=1}^{N} f(Y_i) = \left(\frac{1}{\sqrt{2\pi}\sigma_X}\right)^N \exp\left(-\sum_{i=1}^{N} \frac{(Y_i - \mu_X - \alpha W_i)^2}{2\sigma_X^2}\right), \tag{2.22}$$

$$\prod_{i=1}^{N} g(Y_i) = \left(\frac{1}{\sqrt{2\pi}\sigma_X}\right)^N \exp\left(-\sum_{i=1}^{N} \frac{(Y_i - \mu_X)^2}{2\sigma_X^2}\right), \tag{2.23}$$

where μ_X and σ_X^2 are the mean and variance of $X_i = d_i + n_i$ respectively and W_i is fixed parameter representing the watermarks. Substituting (2.22) and (2.23) into (2.21) yields

$$L(Y) = \frac{\exp\left(-\sum_{i=1}^{N} \frac{(Y_i - \mu_X - \alpha W_i)^2}{2\sigma_X^2}\right)}{\exp\left(-\sum_{i=1}^{N} \frac{(Y_i - \mu_X)^2}{2\sigma_X^2}\right)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma. \tag{2.24}$$

Simplifying (2.24) gives log form as

$$L(Y)' = \frac{1}{2\sigma_X^2}\left(\sum_{i=1}^{N} 2Y_i\alpha W_i - \sum_{i=1}^{N} 2\mu_X\alpha W_i - \sum_{i=1}^{N} \alpha^2 W_i^2\right) \underset{H_0}{\overset{H_1}{\gtrless}} \gamma'. \tag{2.25}$$

Noticing that the last two terms do not depend on the received data y_i and that they are absorbed by the NP threshold, we can ignore them to obtain the final statistics as

$$L(Y, W)' = \frac{1}{N}\sum_{i=1}^{N} Y_i W_i \underset{H_0}{\overset{H_1}{\gtrless}} \gamma'. \tag{2.26}$$

This is the test statistics for this watermark detection problem. Obtaining this test statistics is all we need to detect the presence of watermark in the received data. It is also called the linear correlation detection between watermark and received data. The detector in (2.26) is average product vectors of the received data and watermark because sometimes it is convenient to work with. Since Y_i follows Gaussian distribution, linear correlation is the optimal detection metric for this model. The ease of implementation for this detection metric and the fact that Y_i is approximated with central limit theory are what make this detector widely acceptable. One of the drawbacks for this detector as noted earlier is that Y_i is not always Gaussian distributed and cannot be modelled assuming central limit theory especially when the data is not sufficiently large enough[21][22]. Therefore, linear detector may work sub-optimally.

Next, the statistical parameters of the detector which is the mean and variance are calculated for different hypotheses [6]. In general, if the mean of watermark is zero, then the mean of output detector will also be zero. But if the mean of watermark is different from zero, the output detector will have a specified mean as well. Two cases commonly considered in watermarking are deterministic and random cases of signal W_i.

Case 1: Deterministic Signal

As stated before, since we have the test statistics, it is enough to obtain the statistical parameters of it and use it to check the presence of watermark.

- **Hypothesis H_0 (Watermark absent)** Since there is no watermark embedded, the received data is actually the original data which includes the noise as well, i.e $Y_i = X_i$, with mean $\mathbb{E}(Y_i) = \mathbb{E}(X_i) = \mu_X$ and variance $\mathbb{V}ar(Y_i) = \sigma_X^2$ where $X_i = d_i + n_i$.

  The sample mean of sufficient statistics becomes

$$\begin{aligned} \mu_{L0} &= \mathbb{E}L(Y)' \\ &= \mathbb{E}[\tfrac{1}{N}\textstyle\sum_{i=1}^{N} Y_i W_i] \\ &= \mu_X \tfrac{1}{N}\textstyle\sum_{i=1}^{N} W_i. \end{aligned} \tag{2.27}$$

  The variance is calculated as

$$\begin{aligned} \sigma_{L0}^2 &= \mathbb{V}arL(Y)' \\ &= \mathbb{V}ar[\tfrac{1}{N}\textstyle\sum_{i=1}^{N} Y_i W_i] \\ &= \tfrac{\sigma_X^2}{N^2}\textstyle\sum_{i=1}^{N} W_i^2. \end{aligned} \tag{2.28}$$

- **Hypothesis H_1 (Watermark present)**

  Similarly, for statistical parameters of linear correlation when watermark is present $Y_i = X_i + \alpha W_i$, the mean and variance of received data are $\mathbb{E}(Y_i) = \mu_X + \alpha W_i$ and $\mathbb{V}ar(Y_i) = \sigma_X^2$ respectively. The statistical parameters are

  Mean

$$\begin{aligned} \mu_{L1} &= \mathbb{E}L(Y)' \\ &= \mathbb{E}[\tfrac{1}{N}\textstyle\sum_{i=1}^{N} (X_i + \alpha W_i) W_i] \\ &= \tfrac{\mu_X}{N}\textstyle\sum_{i=1}^{N} W_i + \tfrac{\alpha}{N}\textstyle\sum_{i=1}^{N} W_i^2. \end{aligned} \tag{2.29}$$

  Variance

$$\begin{aligned} \sigma_{L1}^2 &= \tfrac{\sigma_X^2}{N^2}\textstyle\sum_{i=1}^{N} W_i^2 \\ &= \sigma_{L0}^2. \end{aligned} \tag{2.30}$$

Case 2: Random Signal

In some applications, applying a watermark signal as random variables enhances the robustness of embedded signal against certain attacks. It is worth noting that product of two Gaussian random variables is not Guassian, therefore test statistics in real sense is not Gaussian distributed. Hence, central limit theory is invoked which approximates the test statistics as Gaussian when the sample size is sufficiently large [6]. In this case, the statistical parameters differ slighly from the case of determistic signal.

- **Hypothesis** H_0
  Mean

$$\mu_{L0} = \mu_X \mu_W. \tag{2.31}$$

  Let U be the watermark signal known to the detector which will be correlated with W if present, also the assumption is that U and Y are independent then
  Variance

$$\begin{aligned} \sigma^2_{L0} &= \tfrac{1}{N}\{\mathbb{V}ar(Y)\mathbb{V}ar(U) \\ &\quad +\mathbb{V}ar(Y)\mathbb{E}^2(U) \\ &\quad +\mathbb{V}ar(U)\mathbb{E}^2(Y)\}, \end{aligned} \tag{2.32}$$

  where $\mathbb{V}ar(Y) = \sigma^2_X + \sigma^2_n$, see derivation in Appendix (A.1). To avoid confusion, U is used specifically for only calculation of variance.

- **Hypothesis** H_1 .
  Mean

$$\begin{aligned} \mu_{L1} &= \mathbb{E}(Y)\mathbb{E}(W) \\ &= (\mu_X + \alpha\mu_W)\mu_W \\ &= \mu_X\mu_W + \alpha\mu^2_W. \end{aligned} \tag{2.33}$$

  Variance,
  Under H_1, the task is to find the correlation of U and W when we recieved $Y = X + W$. In general, there are two scenarios that can be encountered. First, wrong watermark U is used for correlation during detection and second, correct watermark is used for correlation. If we consider the first case of using wrong watermark which assumes that X, U and W are all independent on each other that is $W \neq U$, then the variance of linear correlator becomes

$$\begin{aligned} \sigma^2_{L1} &= \tfrac{1}{N}\{(\mathbb{V}ar(X) + \alpha^2\mathbb{V}ar(W))\mathbb{V}ar(U) \\ &\quad +(\mathbb{V}ar(X) + \alpha^2\mathbb{V}ar(W))\mathbb{E}^2(U) \\ &\quad +\mathbb{V}ar(U)(\mathbb{E}(X) + \alpha\mathbb{E}(U))\}. \end{aligned} \tag{2.34}$$

  When correct watermark signal U is used for detection, that is X is independent on U and W whereas U and W are correlated by the relaionship $U = \gamma W$, where γ is the correlation coefficient, the variance is re-calculated as

$$\begin{aligned} \sigma^2_{L1} &= \tfrac{1}{N}\{\mathbb{V}ar(X)\mathbb{V}ar(U) \\ &\quad +\mathbb{E}^2(U)\mathbb{V}ar(X) + \mathbb{E}^2(X)\mathbb{V}ar(U) \\ &\quad +\alpha^2\mathbb{E}(W^4) - \mathbb{E}^2(W^2) \\ &\quad +2\alpha\mathbb{E}(X)\alpha^2(\mathbb{E}(W^3) - \mathbb{E}(W)\mathbb{E}(W^2))\}, \end{aligned} \tag{2.35}$$

where the derivation is in Appendix (A.2).

In summary, the test statistics $L(Y)^{'}$ is asymptotically Gaussian such that

$$\frac{L-\mathbb{E}(L)}{\sqrt{\mathbb{V}ar(L)}} \overset{N}{\simeq} \mathcal{N}(0,1), \tag{2.36}$$

where $\mathbb{E}(L)$ and $\mathbb{V}ar(L)$ are given by the above formulas depending on whether we have H_1 or H_0. Therefore, the distribution of linear correlation between the received data and the watermark is summarized as

$$L(Y,W)^{'} \simeq \begin{cases} \mathcal{N}(\mu_{L0}, \sigma^2_{L0}) \\ \mathcal{N}(\mu_{L1}, \sigma^2_{L1}). \end{cases} \tag{2.37}$$

The threshold of NP test can be obtained by using NP equation derived in previous Section as

$$\begin{aligned} \alpha &= \int_{\gamma^{'}}^{+\infty} f(L(Y)^{'}|H_0)dL \\ &= \int_{\gamma^{'}}^{+\infty} \tfrac{1}{\sigma_{L0}}\phi(\tfrac{L-\mu_{L0}}{\sigma_{L0}})dL \\ &= \int_{\frac{\gamma^{'}-\mu_{L0}}{\sigma_{L0}}}^{+\infty} \phi(u)du \\ &= 1-\Phi(\tfrac{\gamma^{'}-\mu_{L0}}{\sigma_{L0}}). \end{aligned} \tag{2.38}$$

Re-arranging (2.38) and solving for $\gamma^{'}$ yields the threshold solution

$$\gamma_\alpha = \Phi^{-1}(1-\alpha)\sigma_{L0} + \mu_{L0}, \tag{2.39}$$

where $\phi(.)$ is the PDF of standard normal distribution $\mathcal{N}(0,1)$, $\Phi(.)$ is the CDF of $\mathcal{N}(0,1)$ and $\Phi^{-1}(1-\alpha) = Q_{1-\alpha}$ is the quantile of $\mathcal{N}(0,1)$. Equation (2.39) is then re-written as

$$\gamma_\alpha = Q_{1-\alpha}\sigma_{L0} + \mu_{L0}. \tag{2.40}$$

Finally, using the pre-determined false alarm α as a function of threshold $\gamma^{'}$, the probability of detection P_D is obtained as

$$\begin{aligned} P_D &= 1-\Phi(\tfrac{\gamma_\alpha-\mu_{L1}}{\sigma_{L1}}) \\ &= 1-\Phi\left(Q_{1-\alpha} - \tfrac{\alpha}{N\sigma_{L1}}\sum_{i=1}^{N} W_i^2\right). \end{aligned} \tag{2.41}$$

Therefore, to maximize the power of the test, we minimize the second term in (2.41). To use the linear correlation detector (2.26), we have to know the mean of the original data μ_X and its variance σ^2_X which is used in the detector. The obvious choice however is

to estimate these statistical parameters, but due to wide range of vector of the received data (i.e. high magnitude), it is sometimes difficult to estimate these parameters. It also means that watermark is not robust against some simple attacks commonly encountered. In [1], it is shown that to get around this problems, computing the normalized correlation between the received data and the watermark of (2.26) is better and more robust detector than simple linear correlation detector. This is called *normalized correlation detector* given as

$$L(Y,W) = \sum_{i=1}^{N} \frac{Y_i}{|Y_i|}\frac{W_i}{|W_i|} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma^{'}. \tag{2.42}$$

2.4 Classical Density Estimation Techniques for Watermarked Features

As already seen in Section 2.3, modelling the distributions of the watermarked data is essential part of watermark detector. In the previous Section when we examined the spatial domain features, the ratio of two distributions is approximated and replaced with its sufficient statistics. This is common practice due to simplicity of its nature and ease of implementations, but it has limitations as well. The three general density estimation techniques that can be applied to watermark applications are parametric, nonparametric and semiparametric estimation techniques. Two most common errors obtained as a result of these estimation techniques are parameter estimation error and modelling error. The usage of each of these techniques depends on the prior knowledge of the model which the underlying density belongs to. If we believe that our model is correct, then parametric estimation is the best choice since it gives best parameter estimates with very high precision. On the other hand, if our model is incorrect we can estimate parameters precisely, but with high modelling error [23], so the convergence of parameter is not accurate as well. Therefore, the obvious choice for this case is to employ a nonparametric estimation technique where the modelling error is zero. Semiparametric technique gives a good compromise between parametric and nonparametric.We briefly summarize these techniques in the next Subsections.

2.4.1 Parametric Estimation Model

The idea behind linear correlation detectors is of parametric nature. The underlying density is assumed to come from a specified model. If the assumed model deviates

completely from the true model, we obtain very high modelling error. Let us denote the modelling error as e_m and parameter estimation error as e_p. The total error of parametric model becomes

$$e_T = e_p + e_m. \tag{2.43}$$

A typical watermark detection model involves using received signal plus noise test statistics given as

$$Y_i = f(X_i|\theta_o) + \varepsilon_i, \quad i = 1, ..., n. \tag{2.44}$$

$f(.)$ is a parametric function specified up to unknown true vector parameters $\theta_o \in \Theta$, ε_i are iid random noise errors with mean $\mu = 0$ and variance $\sigma_\varepsilon^2 > 0$ and $X_i \in \mathbb{R}^d$ denotes the underlying data sequence. The function $f(.)$ also belongs to model space $\mathcal{S}$, $i.e$ $f \in \mathcal{S}$ and the formal representation of parametric model space for this problem is written as

$$\mathcal{S} = \{f(X_i|\theta) : \ \theta \in \Theta\}. \tag{2.45}$$

The error in our model can be measured using the mean square error (MSE) as

$$E(\theta) = \mathbb{E}(Y_i - f(X_i|\theta))^2. \tag{2.46}$$

Furthermore, the function $f(.)$ needs to be approximated if it is unknown or it's parameter vector θ estimated if known. The classical least square

$$\theta_o = \underset{\theta \in \Theta}{\operatorname{argmin}} E(\theta) \tag{2.47}$$

can solve this problem. The empirical counterpart of MSE in (2.46) is given as

$$\hat{E}_n(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}(Y_i - f(X_i|\theta))^2, \tag{2.48}$$

where the empirical least square estimator is

$$\hat{\theta}_n = \underset{\theta \in \Theta}{\operatorname{argmin}} \hat{E}_n(\theta). \tag{2.49}$$

The estimated parameter $\hat{\theta}_n$ is consistent estimate of the true parameter θ_o [24][25], that is

$$\hat{\theta}_n \ \rightarrow \ \theta_o(P) \quad as \ \ n \rightarrow \infty, \tag{2.50}$$

where (P) denotes convergence in probability. The optimal rate of convergence then becomes $\hat{\theta}_n = \theta_o + \mathcal{O}_P(n^{-1/2})$. Therefore, the task becomes finding the optimal finite-dimensional parameters θ_o that characterizes the true model such that (2.50)is satisfied.[23] The two interpretations of θ_o are described as follow

1. Assume that the chosen data comes from true model state space $f \in \mathcal{S}$, we need to only estimate θ_o which completely specifies the system model. Thus, the correct model becomes

$$Y_i = f(X_i|\theta_o) + \varepsilon_i, \tag{2.51}$$

of which we obtain minimum possible error as

$$\begin{aligned} E(\theta_0) &= \mathbb{E}(Y_i - f(X_i|\theta_0))^2 \\ &= \mathbb{E}(f(X_i|\theta_o) + \varepsilon_i - f(X_i|\theta_0))^2 \\ &= \mathbb{E}\varepsilon_i^2. \end{aligned} \tag{2.52}$$

Therefore, we infer that modelling error does not exist and the only source of error comes from noise variance.

2. The second interpretation of θ_o is that if the data comes from an unknown function say $g(X_i)$ which is outside our assumed model class $f \notin \mathcal{S}$, such that the correct model is given as

$$Y_i = g(X_i) + \varepsilon_i. \tag{2.53}$$

Then we choose a known function $f(X_i|\theta)$ from the assumed class as an approximation of $g(X_i)$. The parameter θ_o is such a parameter called limiting value which minimizes the MSE of our model

$$\begin{aligned} E(\theta) &= \mathbb{E}(Y_i - f(X_i|\theta))^2 \\ &= \mathbb{E}(g(X_i) + \varepsilon_i - f(X_i|\theta))^2 \\ &= \mathbb{E}(g(X_i) - f(X_i|\theta))^2 + \mathbb{E}\varepsilon_i^2, \end{aligned} \tag{2.54}$$

where the minimum possible error (irreducible error) for this case is $\mathbb{E}(g(X_i) - f(X_i|\theta_0))^2$. Therefore, $f(X|\theta_0)$ is the closest model to the true function $g(X)$.

Consequently, we conclude that least square estimator is robust to misspecification problem. As we shall see, this approach is used extensively in parametric density ratio estimation of likelihood ratio test using logistic model. Therefore, the idea is to minimize the total error e_T experienced due to parametric assumption, and one way to do this is to use least square approximation method.

2.4.2 Nonparametric Estimation Model

When the a priori knowledge of the underlying function is unknown, it is appropriate to apply nonparametric method which perfectly can recover infinite-dimensional object

of such function. The nonparametric function estimation approach is completely free of the modelling assumption made in the parametric case. Any function can be captured by this approach, hence the modelling error $e_m = 0$. On the other hand, establishing the test statistics function using fully nonparametric method is not very useful because in most binary detection algorithm, it is assumed that both data under H_0 and H_1 are very similar with one only shifted or scaled version of the other. Applying nonparametric method in such situation, we will lose such similarity property. However, understanding nonparametric method is essential for the semiparametric approach used as our detection metric. The estimation error in nonparametric model can be measured also by the mean square error given as

$$E = \mathbb{E}(Y_i - f(X_i))^2. \tag{2.55}$$

Hence, we can obtain the minimum error for $f_o(X)$ being the regression given as the conditional expectation function [26]

$$f_o(X_i) = \mathbb{E}(Y_i|X_i). \tag{2.56}$$

If we let the true function be

$$Y_i = f_o(X_i) + \varepsilon. \tag{2.57}$$

Then the minimal error for this model becomes

$$E = \mathbb{E}\varepsilon^2, \tag{2.58}$$

since there is no modelling error and the error reduces to noise variance. Again the empirical MSE counterpart is

$$\hat{E}_n(\theta) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}(Y_i - f(X_i))^2, \tag{2.59}$$

The optimal rate of convergence for the estimate $\hat{f}(X_i)$ is given as $\mathcal{O}_P(n^{-\alpha})$ where $\alpha < 1/2$. Also, notice that there is no restriction on the form or model of $f(X_i)$. This is the main advantage of fully nonparametric model, for full details regarding nonparametric density estimation or regression models, see [26][17]

2.4.3 Semiparametric Estimation Model

In the previous Subsections, we introduced parametric and nonparametric models. The challenge with parametric method is mainly due to model misspecification especially

when we do not have prior knowledge of the underlying data at hand. But it has fast estimate convergence rate which is $\mathcal{O}_P(n^{-1/2})$. The nonparametric method does not have misspecification problem, but it is known to have slower convergence rate than $\mathcal{O}_P(n^{-1/2})$ [17] and we lose similarity property between data under H_0 and H_1. The semiparametric method is between parametric and nonparametric methods which allows model flexibility. In many watermark applications, an attacker may change the distribution of the original data. In this case, it is imperative to apply a technique that captures the model of the system accurately such that the power of the test is maximized. Let us denote a semiparametric model expressed as conditional expectation function as

$$\begin{aligned} \mathbb{E}(Y_i|X_i) &= f_o(X_i) \\ &= g(b(X_i|\theta)). \end{aligned} \tag{2.60}$$

This is idea of single index model (SIM) which summarizes input variables X_i as a one dimensional problem [26]. There are two step estimation processes of SIM model. First, we estimate the parameter $\hat{\theta}$, then use $\hat{\theta}$ in the second stage process which is estimating function $\hat{g}(.)$ nonparametrically (univariate). Therefore, both θ and $g(.)$ are to be estimated directly from the given data [26][17]. Similarly as in previous Subsections, we try to obtain the parameter θ and the function $g(.)$ that minimizes the error given as

$$E(\theta, g(.)) = \mathbb{E}[(Y_i - g(b(X_i|\theta))]^2, \tag{2.61}$$

where $Y_i = f_o(X_i) + \varepsilon$ and we obtain minimum possible error as

$$\begin{aligned} E(\theta_o, g_o(.)) &= \mathbb{E}[Y_i - g(b(X_i|\theta))]^2 \\ &= \mathbb{E}[f_o(X_i) + \varepsilon - g(b(X_i|\theta))]^2 \\ &= \mathbb{E}[f_o(X_i) - g_o(b(X_i|\theta_o))]^2 + \mathbb{E}\varepsilon^2. \end{aligned} \tag{2.62}$$

Equation (2.62) is the case when modeling error exists, but when we do not have modelling error (2.62) becomes $E(\theta_o, g_o(.)) = \mathbb{E}\varepsilon^2$.

Chapter 3

Optimal Watermark Detection in the Presence of Non-Gaussian Features

Figure 3.1 shows some of the distributions of most natural image pixels for different greyscale values and they are invariant to scaling see [21][22]. We can see clearly that in most cases, the densities of image pixels are skewed and not unimodal in nature, hence Gaussian PDF may not work very well during watermark decoding. Therefore, blindly estimating the statistical distribution of these watermarked images is appropriate.

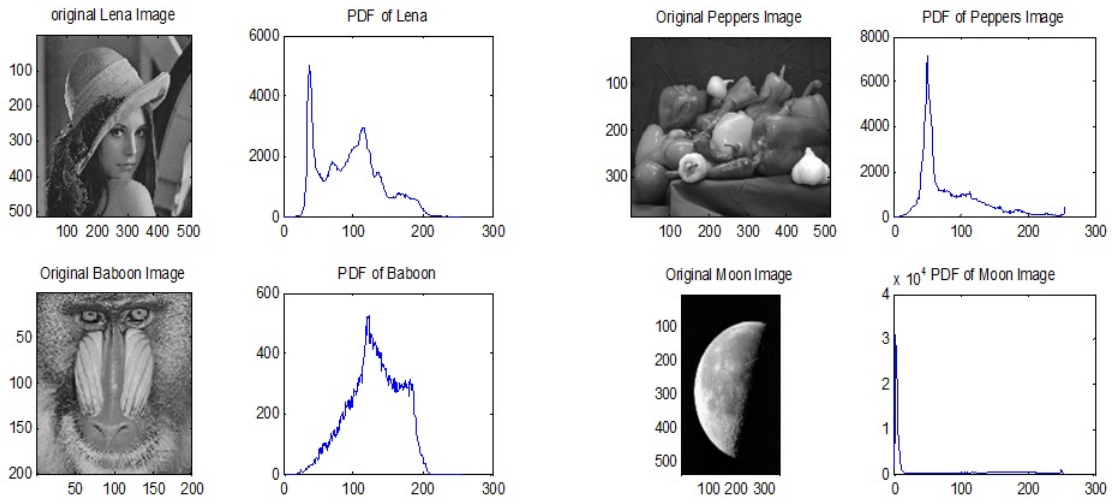


Figure 3.1: The histogram of most popular images.

3.1 Gaussian Mixture Model Distributed Features

The Gaussian mixture model (GMM) is a popular parametric probability density function that has been used extensively to model data which reveals skewness and multi-modality. A similar approach where the host features have been modelled as GMM can be found in [11]. The significant different between our model and the one in [11] is how the number of GMM components is obtained, also we approximate GMM with logistic regression and use it as likelihood ratio test. The advantages of using GMM to model the distribution of images are summarised as follow

- Most images are hardly unimodal Gaussians as observed in Figure 3.1, so the GMM can almost capture the underlying density perfectly if it is multimodal Gaussian.
- In the case of unimodal probability density distribution of a data, GMM can easily be reduced to a single component density, and comparable to optimal linear detector.
- GMM has the advantage of less computational load put on the detection system.
- In some cases, with GMM, the attack parameters such as Gaussian attack can be estimated and learned.

The obvious disadvantage of using GMM is that if data is close to other densities such as exponential, gamma or even mixture models, modelling error increases significantly.

To illustrate the proposed method, we assume that watermark is embedded in the presence of an additive white Gaussian noise (AWGN), with host features distributed according to Gaussian mixture model, the entire model for this system is demonstrated in the Figure 3.2 below

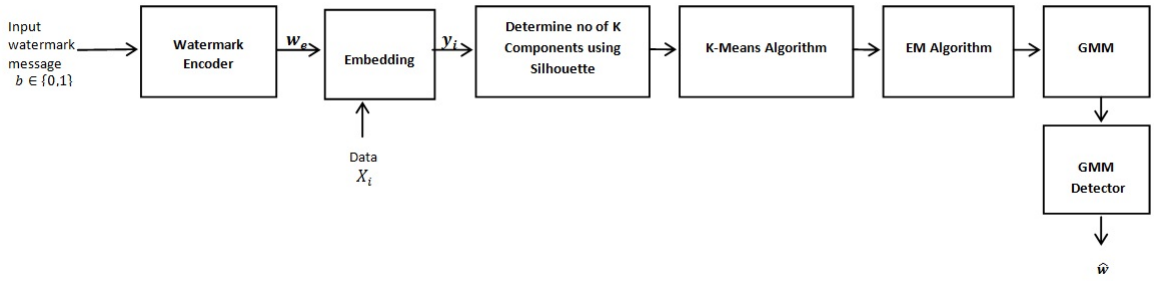


Figure 3.2: General block diagram of GMM setup.

There are many different researched methods of encoding watermark prior to embedding into the host data for security reasons. Since designing an optimal robust embedding strategy is not our goal in this thesis, we simplify the embedding process by encoding only one bit of message $b = 1$. It is proven [1] that representing watermark bits as identically and independent *iid* random variables provides robustness needed in watermark applications. This random variables is generally regarded as reference pattern. Embedding watermark $W_b = \mathcal{N}(\mu_w, \sigma_w^2)$ encoded this way has significant effect on the detection metric employed. After encoding, the watermark is inserted additively into host data vector X_i and can experience AWGN noise attack denoted as $n_i \backsim \mathcal{N}(\mu_n, \sigma_n^2)$ to yield

$$Y_i = X_i + \alpha W_b + n_i, \tag{3.1}$$

where α is the scaling parameter representing trade-off between watermark robustness and watermark strength.

The original data X_i is distributed according to Gaussian mixture model $X_i \backsim \sum_{j=1}^{K} \pi_j \mathcal{N}(\mu_{xj}, \sigma_{xj}^2)$ and finally

$$Y_i \simeq \sum_{j=1}^{K} \pi_j \mathcal{N}(\mu_{yj}, \sigma_{yj}^2), \tag{3.2}$$

where $\mu_{yj} = \mu_{xj} + \alpha\mu_w$ and $\sigma_{yj}^2 = \sigma_{xj}^2 + \alpha^2\sigma_w^2 + \sigma_n^2$.

Obtaining a close form equation for this model is extremely difficult unlike the simple linear correlation detector, hence during performance evaluation we resort to Monte Carlo simulation. In this embedding algorithm, there is a requirement to have the watermarked version to be perceptibly similar to the original data. Therefore, the image data X_i and

the watermarked counterpart will have similar distributions. This requirement ensures that addition of watermark to the original data does not modify the data significantly. Hence, we can exploit this property to estimate the watermarked data using GMM density. If we let the watermarked data or received image to be y_i, and if we also assume that this data is identically and independent random variables, then under hypotheses $H_i, i = \{0, 1\}$ the received data is given as

$$p(y_1, y_2, ..., y_N \mid H_1) = \prod_{i=1}^{N} p(y_i|H_1),$$
$$p(y_1, y_2, ..., y_N \mid H_0) = \prod_{i=1}^{N} p(y_i|H_0). \tag{3.3}$$

The mixture models for the underlying density function with K components under both hypotheses are

$$p(y_i|H_1) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mu_{j1}, \sigma_{j1}^2)$$
$$p(y_i|H_0) = \sum_{j=1}^{K} \pi_j \mathcal{N}(\mu_{j0}, \sigma_{j0}^2), \tag{3.4}$$

where $\sum_{j=1}^{K} \pi_j = 1$.

3.1.1 Determine the Number of Components for GMM Using Silhouette Validation Technique

We want to estimate the parameters of Gaussian mixture and the number of mixture components. There are many existing methods already developed to obtain the number of components in mixture models. Here, we adopt a simple technique that is used to validate clustered data called Silhouette validation technique [27]. Since we assume blind watermark detection where the density of the original data is not available during detection, determining the optimal number of mixture components of our model becomes a challenge. If the number of the components K is too large, it over estimates the true PDF and if it is too small it suboptimally models the true PDF. The Silhouette value tells us how closely a data point is associated with its assigned cluster and how dissimilar it is from the other clusters [27]. The algorithm of using Silhouette value to determine optimal cluster number is as follow

Silhouette Algorithm

1. Select an arbitrarily number range for clusters (components) say $k = 1$ to 10.
2. Calculate the *K-means* of these selected clusters.
3. Determine the silhouette value of each clustered data point and the mean value over all the points.
4. Plot the graphical representation of silhouette mean value to find the peak corresponding to optimal number of cluster.

The next step is to use the k-th cluster corresponding to the peak Silhouette mean value in *K-means* again to determine the initialization parameters for expectation maximization (EM) algorithm given as

$$\theta = (\pi_j, \mu_j, \sigma_j^2)_{j=1}^K. \tag{3.5}$$

These initialization parameters obtained using K-means algorithm for k component PDFs are derived as

$$\mu_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_i \mathbb{1}(x_i \in G_j),$$

$$\sigma_j^2 = \frac{1}{n_j} \sum_{i=1}^{n_j} (x_i \mathbb{1}(x_i \in G_j) - \mu_j)^2,$$

$$\pi_j = \frac{n_j}{N}, \tag{3.6}$$

where $\sum_{i=1}^{n_j} \pi_j = 1$, $n_j \in N$, $j = 1, ..., K$ and $\mathbb{1}(.)$ is an indicator function defined as

$$\mathbb{1} = \begin{cases} 1, & if \quad x_i \in G_j \\ 0, & if \quad x_i \notin G_j. \end{cases} \tag{3.7}$$

The μ_j and σ_j^2 are the sample mean and variance of each component of the density respectively, whereas π_j is the mixing coefficient of the density, G_j is the jth cluster group and N is the total number of data in the entire image or data. We are now ready to use this parameter set $\theta = (\pi_j, \mu_j, \sigma_j^2)_{j=1}^K$ from *K-Means* to initialize EM algorithm.

3.1.2 Iterative Expectation Maximization Algorithm for Estimating the Parameters

The EM algorithm is generally accepted algorithm used to estimate or refine the parameters of Gaussian mixture model [12]. The algorithm requires initialization of parameters which is obtained by *K-Means* as described above.

Iterative EM Procedure

1. Initialize using parameters obtained from *K-Means* algorithm.
2. For each data point x_i, calculate the probability that point x_i belongs to the jth term called posterior probability $\hat{\tau}_i$.
3. The parameters are updated iteratively using the estimates $\hat{\pi}_j, \hat{\mu}_j$ and $\hat{\sigma}_j^2$
4. Repeat step 2 to step 3 until estimate converges.

We note that it is not our intention to develop new parameter estimation algorithm, therefore we use the popular iterative EM algorithm since it has been proven to be efficient. The above EM procedure can also be found in [12]. The posterior probability of step 2 in the algorithm is given analytically as

$$\hat{\tau}_i = \frac{\hat{\pi}_j \mathcal{N}(y_i; \hat{\mu}_j, \hat{\sigma}_j^2)}{\sum_{j=1}^{K} \hat{\pi}_j \mathcal{N}(y_i; \hat{\mu}_j, \hat{\sigma}_j^2)}. \tag{3.8}$$

The updated parameters which we will use to substitute the parameters obtained by the *K-Means* algorithm are given as

$$\hat{\mu}_j = \frac{1}{n_j} \sum_{i=1}^{n_j} x_i \mathbb{1}(x_i \in G_j),$$

$$\hat{\sigma}_j^2 = \frac{1}{n_j} \sum_{i=1}^{n_j} (x_i \mathbb{1}(x_i \in G_j) - \mu_j)^2,$$

$$\hat{\pi}_j = \sum_{i=1}^{N} \hat{\tau}_i. \tag{3.9}$$

3.1.3 GMM Detector for Watermark Signal

Next, since watermarked data is estimated and ready, we can apply a detection criterion such as Bayes or Neyman Pearson criterion to detect the embedded watermark. The GMM detector is a family of Bayes criterion, but the significant contribution is that the PDF used in this detector is explicitly estimated using Gaussian mixture model. The likelihood ratio test (LRT) for this detector is therefore given as

$$L(Y) = \frac{\prod_{i=1}^{N} p(y_i|H_1)}{\prod_{i=1}^{N} p(y_i|H_0)} \underset{H_0}{\overset{H_1}{\gtrless}} \gamma, \tag{3.10}$$

where γ is the threshold. As stated earlier, it is difficult to obtain a closed form equation of Gaussian mixture model detector like in the case of the linear detector. This is therefore one of the setbacks of this model since the only way to check the performance measure for this detector is by resorting to Monte Carlo (MC) simulation. However, it is known that MC simulation approach is always very close to the true system if the number of experimental trial is chosen to be very high.

3.2 An Example of Gaussian Mixture Model

Here, we apply the GMM detection technique to watermarked greyscale images. The images selected for watermarking are some of the popular images used in image processing field which are shown in Figure 3.3. We assume that an additive embedding rule has been used such as the one explained in Section (3.1), then the task is to detection the embedded watermark in the host images at the receiver side. As already shown, the first step is to estimate the probability density function of the host images from the watermarked images.

The set of test images selected are 128 x 128 pixels in size, each has different characteristics of intensity and distributions. These test images are selected in such a way that no single image has the same data distributions with each other. For each image we employ the Silhouette validation algorithm discussed in Section (3.1.1) and the results of these data clusters validation is shown in Figure 3.4

As we can see in Figure 3.4, different number of clusters for each image is plotted against the mean Silhouette value. In each image, the cluster corresponding to the maximum Silhouette mean value is regarded as the actual number of clustered data in

Figure 3.3: Test Images (a)Baboon (b)Lena (c)Peppers (d)Elaine (e)Fishing Boat (f)Clock.

that image. We obtained three clusters in baboon image, five in Lena, four in pepper, five in Elaine, two in fishing boat and two in the clock.

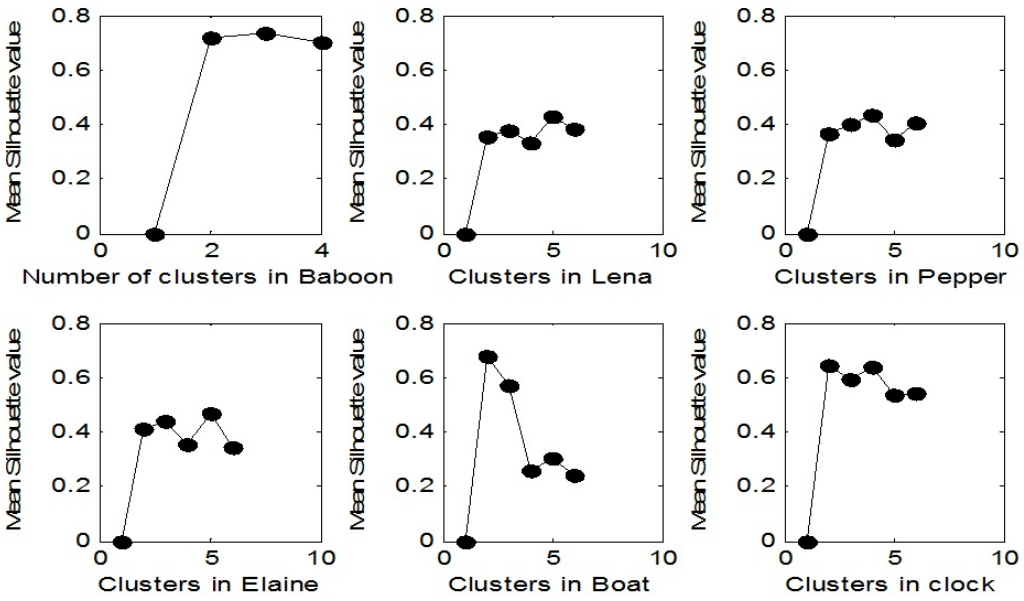


Figure 3.4: Number of clusters in each test image.

3.3 Experimental Results of Gaussian Mixture Model

We perform experiment on the watermarked images shown in Figure 3.3. For each experiment, the watermark signal is chosen to be *iid* random variables from Gaussian distribution with parameters mean μ_w and variance σ_w^2. The variance of the watermark σ_w^2 is equivalent to the scaling parameter α which controls the strength of the watermark. One may vary α as needed so far as it does not significantly corrupt the original image data. The idea is to compare the performance of GMM detector with the traditional linear detector in the presence of images with different distributions. We shall experiment this in simulation studies, for now lets concentrate on estimating the parameters of the unknown density using GMM.

3.3.1 Monte Carlo Simulation of GMM

Generally in simulations, we are often required to generate more random variables following an underlying image watermark data received which has been estimated by the GMM procedure [12]. To generate these random samples from GMM distribution, the following Monte Carlo algorithm is used

GMM Algorithm:

1. Use the estimated parameters obtained by the Iterative EM algorithm $\theta = (\hat{\pi}_j, \hat{\mu}_j, \hat{\sigma}_j^2)_{j=1}^K$ to generate GMM.
2. Generate $N\text{-}Uniform(0,1)$ random variables U_i
   - If $U_i < \hat{\pi}_1$, then Y_i belongs to group 1 of the density.
   - If $\hat{\pi}_1 \leq U_i < (\hat{\pi}_1 + \hat{\pi}_2)$, then Y_i belongs to group 2 of the density.
   - If $(\hat{\pi}_1 + \hat{\pi}_2) \leq U_i < (\hat{\pi}_1 + \hat{\pi}_2 + \hat{\pi}_3)$, Y_i belongs to group 3 of the density.
   - And so on ...
3. Generate $\mathcal{N}(\hat{\mu}_j, \hat{\sigma}_j^2), j = 1 : K(term)$ according to group members found in step 2.
4. Then, the GMM is generated as $p(y_i|H_k) = \sum_{j=1}^K \pi_j \mathcal{N}(\mu_{k_j}, \sigma_{k_j}^2)$ for $k = \{0, 1\}$ and $i = 1, ..., N$.

In Figure 3.5 to 3.7 , we plotted the original histograms of these watermarked images with their corresponding estimated data obtained using the Monte Carlo finite Gaussian

mixture model. We can see from these figures, that the estimated data using the proposed Gaussian mixture model is very close to the true underlying histograms of the watermark images which are similar in distribution to the original image data. Looking at these histograms, we can also see that applying a detector such as linear detector that assumes image data to be Gaussian distributed may not perform very well.

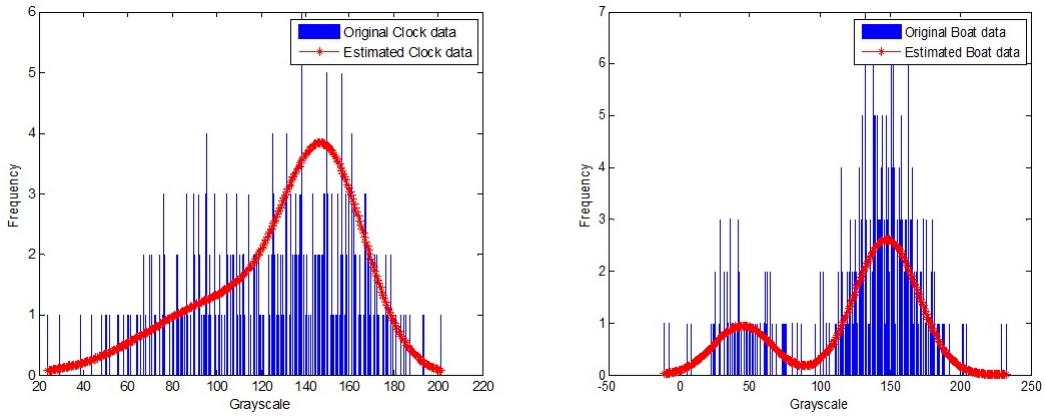


Figure 3.5: Comparing the histogram of original data to the estimated data using GMM (left clock and right fishing boat).

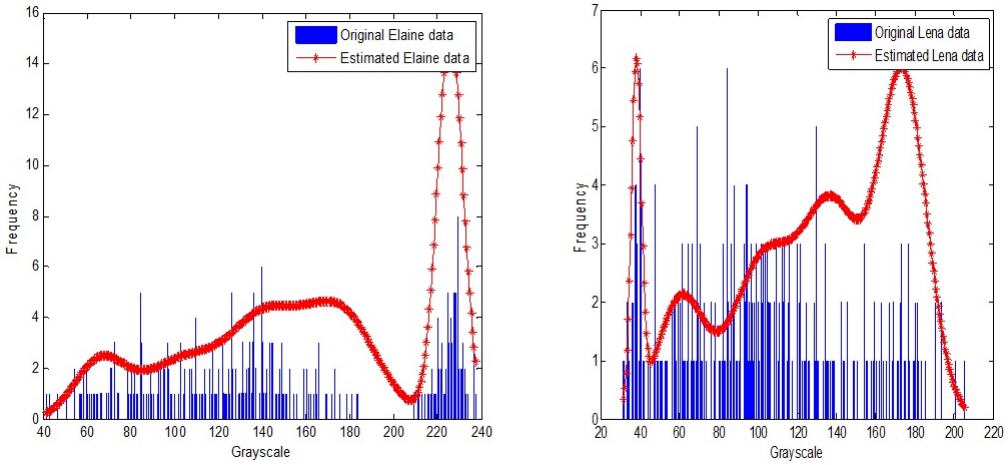


Figure 3.6: Comparing the histogram of original data to the estimated data using GMM (left Elaine and right Lena).

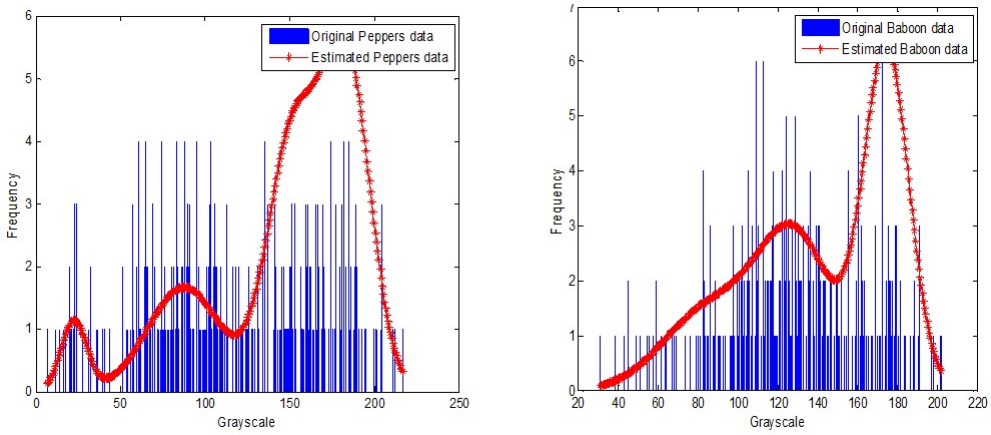


Figure 3.7: Comparing the histogram of original data to the estimated data using GMM (left Peppers and right Babbon).

3.4 Density Ratio Estimation of Likelihood Ratio Test

Density ratio estimation (DRE) is an estimation technique which focuses on estimating the ratio of two densities directly [13]. In other words, with DRE technique, we try to avoid estimating individual densities from two set of random variables and estimate their ratio directly. As already seen in Section 3.3, estimating mixture models (GMM) of individual densities coming from both hypotheses is very cumbersome. It requires accurately estimating the following parameters before being used for detection.

- The number of components of the density using Silhouette validation technique. Sometimes, we can encounter error using this technique to determine number of mixture components in a data.
- The initialization parameters of EM-algorithm must be obtained using *K-means* algorithm. *K-means* requires accurate initial guess of number of components to work well. Therefore if we unknowingly made an error while estimating mixture components, the entire GMM estimation becomes suboptimal and affects watermark detection probability.

We clearly see that the idea of estimating density ratios directly is beneficial and easier than estimating individual densities.

3.4.1 Density Ratio Estimation of Likelihood Ratio Test via Exponential Tilt Model

Exponential tilt model also known as density ratio model is equivalent to classical parametric logistic regression model. The logistic model is well known as probabilistic classification model [13][15][28] . In probabilistic classification related to watermark detection system, we try to classify the two hypotheses as binary outcomes. Let us assign output binary labels or statues to the watermark hypotheses using an indicator function given as

$$\begin{cases} Z = 0, \;\; (X_1, ..., X_m) \backsim g(x) \\ Z = 1, \;\; (Y_1, ..., Y_n) \backsim f(x), \end{cases} \tag{3.11}$$

where the output Z is regarded as a random variable and $n + m = N$ is the total sample size.
Specifically, if we consider the two hypotheses data as $(X_i, ..., X_m; Y_j, ..., Y_n)$, then their corresponding output binary response becomes $(Z_1, ..., Z_N) = (0, ..., 0; 1, ..., 1)$ where $i = 1, ..., m$ and $j = 1, ..., n$, $X_i \in \mathbb{R}^p$ and $Y_i \in \mathbb{R}^p$, p is the p-dimensional vector same size as the original data, it has to be estimated from the data. Let us denote a new variable $x \in \{H_0, H_1\}$, then following Baye's rules the density ratio is expressed as

$$r(x|\theta) = \frac{P(Z = 1|x)}{P(Z = 0|x)}, \tag{3.12}$$

or

$$r(x|\theta) = \frac{f(x)P(Z = 1)}{g(x)P(Z = 0)}, \tag{3.13}$$

where $n = m$, complete derivation and proof of this function can be found in [13].

If we apply the logistic model (sigmoid function) to the model's relationships between watermarked and non-watermark data (since they are probabilities),we have

$$P(Z = 1|x, \theta) = \frac{exp(\alpha + \beta^T b(x))}{1 + exp(\alpha + \beta^T b(x))} \tag{3.14}$$

$$P(Z = 0|x, \theta) = \frac{1}{1 + exp(\alpha + \beta^T b(x))}, \tag{3.15}$$

where $\theta = (\alpha, \beta), \theta \in \Theta \subset \mathbb{R}^d$ is the unknown parameter to be estimated from the data with $d = p + 1$ and $b(x) \in \mathbb{R}^b$ is a function which has the form

$$b_1(x) = x, b_2(x) = x^2, ..., b_p(x) = x^p. \tag{3.16}$$

Taking the likelihood ratio test of these probabilities in (3.14) and (3.15) yield the expression derived as

$$\begin{aligned} r(x|\theta) &= \frac{P(y=1|x,\theta)}{P(y=0|x,\theta)} \\ &= exp(\alpha + \beta^T b(x)). \end{aligned} \tag{3.17}$$

The equation (3.17) is called the link function between densities $g(x)$ and $f(x)$. The goal is to express the likelihood ratio test during watermark detection as a certain function called the link function $exp(\alpha + \beta^T b(x))$ and to get powerful estimator which is more efficient. Hence, using exponential family distribution as a link function is very useful because it covers large densities from which the real density ratios of two distributions are obtained. The link function depends on unknown parameter θ, estimating this parameter accurately will yield a link function between the density ratios of the two distributions. Therefore, we form a powerful and more efficient likelihood ratio test detector than the traditional linear correlation.

3.4.2 Maximum Likelihood Estimation of Parameters of Exponential Tilt Model

Figure 3.8 illustrates the idea of how to estimate the parameters of the exponential function. To estimate the parameters, the idea is to utilize all the data together coming from both hypotheses, then apply maximum likelihood estimation to obtain the estimate of the density ratio of two set of random variables.

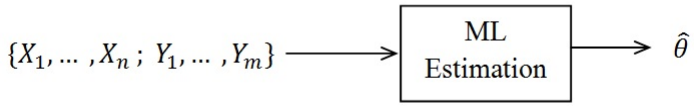


Figure 3.8: Block diagram illustrating ML estimation process of the parametric density ratio.

Let us assume that we have respectively independent data of watermarked and non-watermark data given as

$$x_N = \begin{cases} (X_1, ..., X_m) \backsim g(x) \\ (Y_1, ..., Y_n) \backsim f(x), \end{cases} \tag{3.18}$$

The following derivation of maximum likelihood parameter estimation can also be found in [15][26]. We can express the likelihood ratio test as a link function between the two continuous densities as

$$r(x;\theta) = \frac{f(x)}{g(x)}. \tag{3.19}$$

Re-arranging we obtain

$$r(x;\theta)g(x) - f(x) = 0,$$

if only $\theta = \theta^*, \theta \in \Theta$. Let $\eta(x;\theta) : \mathcal{H}$ x $\Theta \rightarrow \mathbb{R}^d$ be a vector function, then the moment matching estimator for density ratio model which is derived in [15] is given as

$$\mathbb{E}(r(x;\theta)\eta(x;\theta)) - \mathbb{E}(\eta(x;\theta)) = 0. \tag{3.20}$$

The corresponding estimation function for density ratio is therefore given as

$$\hat{\varphi}_\eta(\theta) = \frac{1}{m}\sum_{i=1}^{m} r(x_i;\theta)\eta(x_i;\theta) - \frac{1}{n}\sum_{i=1}^{n}\eta(x_i;\theta). \tag{3.21}$$

The estimate of θ^* requires using Newton algorithm since our problem involves system of nonlinear equations, and the roots of the following equation

$$\hat{\varphi}_\eta(\hat{\theta}) = 0 \tag{3.22}$$

gives our estimate $\hat{\theta}$. The mathematical expression for Newton iterations algorithm to estimation θ is given as

$$\hat{\theta}^{new} = \hat{\theta}^{old} - \left(\mathcal{J}(\theta^{old})\right)^{-1}\hat{\varphi}_\eta(\theta^{old}), \tag{3.23}$$

where $\mathcal{J}(.)$ is the Jacobian matrix.

The iterations is stopped when we obtain the value of $\hat{\theta}$ such that $\hat{\varphi}_\eta(\hat{\theta}) = 0$. In experiment, it is difficult to obtain the exact solution that yields $\hat{\varphi}_\eta(\hat{\theta}) = 0$, so the exact solution is obtained by checking when the estimated error is less or equal to a predetermined small value $\epsilon > 0$. The bound of this estimated error is expressed as

$$\|\frac{\hat{\theta}^{new} - \hat{\theta}^{old}}{\hat{\theta}^{old}}\| \leq \epsilon. \tag{3.24}$$

The detailed computational Newton algorithm conducted in MATLAB software can be found in [29]

The optimal close form equation for the equation $\eta(x;\theta)$ is derived in [15] and is given as

$$\eta_{opt}(x;\theta) = \frac{1}{1+\rho r(x;\theta)}\frac{\partial}{\partial}\log r(x;\theta), \tag{3.25}$$

where $r(x;\theta) = exp(\alpha + \beta^T b(x))$ and $\rho = \frac{n}{m}$, note that in most watermark applications $\rho = 1$, since $m = n$.

3.4.3 An Example of Likelihood Ratio Test as Link Function

In this Section, we apply the technique of logistic regression model discussed in the previous Section to illustrate how the true parameter values are obtained. Then we compare with the estimated counterparts to understand how the estimate approaches the true value. It is worth noting that simple Gaussian distribution is used as the data features and watermark added in the data is also Gaussian distributed. In watermarking, data from H_0 and H_1 only differ either in their means or variances. If they differ in their mean values, that is one is the shifted version of the other, their density ratio yields linear function. On the other hand, if their mean values are the same and differ only in their variances, they yield quadratic density ratio.

- **Quadratic Density Ratio Function:** Here, we illustrate with an example where the ratio of two densities such as Gaussian gives a quadratic model. The simple case is scaled density functions given as

  $$f(x) = \frac{1}{\sigma}g(\frac{x}{\sigma}), \tag{3.26}$$

  where $\mathbb{E}(X) = \mathbb{E}(Y) = \mu$, given that $X \backsim g(x)$ and $Y \backsim f(x)$, σ controls the shape of the densities.

  More complicated case is when $\mathbb{E}(X) \neq \mathbb{E}(Y)$ with also a scaling factor σ. Such a relationship between $f(x)$ and $g(x)$ is not common in watermark applications and it is given as

  $$f(x) = \frac{1}{\sigma}g(\frac{x-\mu}{\sigma}). \tag{3.27}$$

  An example of such quadratic model is expressed as

  $$\begin{cases} (X_1, ..., X_m) \backsim \mathcal{N}(\mu, \tau^2) = g(x) \\ (Y_1, ..., Y_n) \backsim \mathcal{N}(\mu, \sigma^2) = f(x). \end{cases} \tag{3.28}$$

The likelihood ratio test for these variables gives

$$r(x|\theta) = \frac{\mathcal{N}(\mu, \sigma^2)}{\mathcal{N}(\mu, \tau^2)} = \frac{\frac{1}{\sqrt{2\pi}}\frac{1}{\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi}}\frac{1}{\tau}\exp\left(-\frac{(x-\mu)^2}{2\tau^2}\right)}. \tag{3.29}$$

Further simplification gives the quadratic link function which is given as

$$r(x|\theta) = \frac{\mathcal{N}(\mu, \sigma^2)}{\mathcal{N}(\mu, \tau^2)} = \exp\left(\alpha + \beta_1 x + \beta_2 x^2\right), \tag{3.30}$$

where $\alpha = (\log(\frac{\tau}{\sigma}) + \frac{\mu^2}{2\tau^2} - \frac{\mu^2}{2\sigma^2})$, $\beta_1 = (\frac{\mu}{\sigma^2} - \frac{\mu}{\tau^2})$, $\beta_2 = (\frac{1}{2\tau^2} - \frac{1}{2\sigma^2})$ and $\theta = (\alpha, \beta_1, \beta_2)$.

Clearly, we see that equating the density ratio of distributions like Gaussian distribution with exponential family link function is justified. Furthermore, we observed that the optimal dimensionality for the link function for the Gaussian case is $d = 2$. Letting $\tau = 2$, $\sigma = 4$ and $\mu = 20$ for this example, the true parameter becomes

$$\theta_o = (37.5, -3.75, 0.09375). \tag{3.31}$$

- **Linear Density Ratio Model:** Another case of model where two density functions differ only in their mean values is expressed as

$$f(x) = g(x - \mu), \tag{3.32}$$

where $\mathbb{E}(X) \neq \mathbb{E}(Y)$, $\mathbb{V}ar(X) = \mathbb{V}ar(Y) = \sigma^2$ and $\theta = (\alpha, \beta_1)$.
The next example is a typical example of watermark applications where the watermark signal is in form of deterministic signal.

$$\begin{cases} (X_1, ..., X_m) \backsim \mathcal{N}(0, \sigma^2) = g(x) \\ (Y_1, ..., Y_n) \backsim \mathcal{N}(\mu, \sigma^2) = f(x) \end{cases} \tag{3.33}$$

The likelihood ratio test for this model is

$$r(x|\theta) = \frac{\mathcal{N}(\mu, \sigma^2)}{\mathcal{N}(0, \sigma^2)} = \frac{\frac{1}{\sqrt{2\pi}}\frac{1}{\sigma}\exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}{\frac{1}{\sqrt{2\pi}}\frac{1}{\sigma}\exp\left(-\frac{x^2}{2\sigma^2}\right)}. \tag{3.34}$$

Again, simplifying

$$r(x|\theta) = \frac{\mathcal{N}(\mu, \sigma^2)}{\mathcal{N}(0, \sigma^2)} = \exp\left(\alpha + \beta_1 x\right) \tag{3.35}$$

where $\alpha = (\frac{-\mu^2}{2\sigma^2})$, $\beta_1 = \frac{\mu}{\sigma^2}$, $\beta_2 = 0$ and $\theta = (\alpha, \beta_1)$.

If we let $\sigma = 4$ and $\mu = 20$, we get the true parameter as

$$\theta_o = (-12.5, 1.25). \tag{3.36}$$

Therefore, the link function totally depends on parameter vector $\theta = (\alpha, \beta_i), i = 1, ..., p$ and parameters of the densities μ, σ. In the simulation studies, we apply the maximum likelihood and the Newton method to estimate these parameters and check how fast they approach the true values obtained analytically.

3.5 Least Square Approximation of Link Function to Mixture Model

In the previous Section where we made assumption of unimodal Gaussian distribution in the example of density ratio, we obtained a well define analytical true parameter vector $\theta = (\alpha, \beta_1, ..., \beta_p)$ of the link function. As already shown, the distributions of noise such as original images data are mostly skewed and multimodal in nature. Modelling it with distribution such as Gaussian mixture model (GMM) for parametric case gives very powerful estimator and therefore can improve the detection measure. Also, we noticed in Section 3.4 that we can avoid estimating individual densities of the hypotheses and play with density ratio directly. This technique offers some advantages such as ease of implementation of test statistics and adapts fairly for smaller sample size N. That is, it does not require central limit theory like linear correlation case to give good estimate of test statistics. The challenge with the case of mixture model distribution is to obtain a good analytical true parameter vector θ, such that estimated parameter $\hat{\theta}$ is comparable to θ. The ratio of Gaussian mixture models is very difficult to solve analytically and we want to avoid individual density estimation. We apply classical least square approximation technique to fit the exponential link function to the mixture model. This approach helps to reduce modelling error due to misspecification. Least square approximation to an unknown function has been studied extensively in [25].

Let $L(x|\eta)$ represents the density ratio of the multimodal data such as Gaussian mixture models with parameter set η and let $r(x|\theta)$ be the link function of which the optimal dimension for unimodal Gaussian case is $d = 2$. Let $\hat{\theta}_N$ be an estimate of

true value parameter θ^*. Then the measured expected error is obtained by applying an empirical mean sqaure error

$$E_{rr}(\theta, d) = \frac{1}{N}\sum_{i=1}^{N}(L(x_i|\eta) - r_d(x_i|\theta))^2, \tag{3.37}$$

where $\hat{\theta}_N$ is a consistent estimator of θ^*, that is

$$\hat{\theta}_N \to \theta^*(P) \ \ as \ \ N \to \infty. \tag{3.38}$$

We apply the classical least square estimator to approximate link function to mixture model as

$$E_{rr}(\hat{\theta}_N, \hat{d}) = \underset{\theta\in\Theta, d\in\mathcal{D}}{\operatorname{argmin}} E_{rr}(\theta, d). \tag{3.39}$$

We find parameter set such that

$$\lim_{N\to\infty} E_{rr}(\hat{\theta}_N, \hat{d}) = E_{rr}(\theta^*, d^*). \tag{3.40}$$

The dimension $d \in \mathcal{D}$ of $r_d(x_i|\theta)$ is varied from $d = 2, 3, ...$ until we obtain an optimal d^* that minimizes the error $E_{rr}(\theta, d)$ and also produces the optimal parameter set θ^*. This is observed in the simulation studies. Hence, we can apply $\hat{r}_{d^*}(x|\theta^*)$ as a link function to the density ratio of Gaussian mixture models, thereby avoiding density estimation of complicated GMM.

3.6 Monte Carlo Evaluation of Link Function as Test Statistics via Bootstrap Method

The basic methodology for checking the presence or absence of watermark in a received signal is through null and alternative hypotheses testing. The test statistics in our case is the link function already developed in the previous Section. As noted earlier, there is no close form formula of test statistics of exponential tilt model which is an approximation of mixture models. Therefore, we resort to Monte Carlo simulation via bootstrapping which is generally very close to the true model as the sample size tends to infinity i.e $N \to \infty$ .We follow Neyman Pearson hypothesis criterion as it is popular testing model in watermark applications. Two types of errors commonly encountered during detection of watermark using NP test are type 1 error commonly known as probability of false

alarm P_{fa} and type 2 error called probability of missed detection P_m. These errors are given respectively as

$$P_{fa} = P\left(r(X|\hat{\theta}) > \gamma|H_0\right),$$
$$P_m = P\left(r(X|\hat{\theta}) < \gamma|H_1\right), \tag{3.41}$$

where γ is a pre-determined threshold.
The null hypothesis is rejected when our test statistics exceed a pre-determined threshold γ_α, that is $r(x|\hat{\theta}) > \gamma_\alpha$, where γ_α is also called the control limit or the critical value for the test statistics [12]. The critical value is the value of our test statistics under H_0 that demarcate the regions where null hypothesis will be accepted or rejected. It is determined by constraining the null hypothesis (probability of false alarm) to a significance level α which is amount of error we are willing to accept. The rejection probability is generally very small say $\alpha = 0.05$ and γ_α is found as

$$\gamma_\alpha = min\left\{\gamma : P\left(r(X|\hat{\theta}) > \gamma|H_0\right) \leq \alpha\right\}, \tag{3.42}$$

Before proceeding to estimation of type 1 error and γ_α, we need to estimate the parameter $\hat{\theta}$ used in the link function $r(x|\hat{\theta})$.

3.6.1 Bootstrap Estimation of Link Function Parameters

Bootstrap is a method of Monte Carlo simulation that assumes no knowledge of parent distribution or parameters of a distribution. Instead, sample of an estimate of parameters of the population is used. The general procedure of bootstrap is to resample with replacement from the original sample [12]. Let us illustrate a typical example in bootstrap approach, if we assume that random sample is given as $x = (4, 7, 2, 1)$, then the first sample with replacement could be $x^{*1} = (x_4, x_2, x_2, x_1) = (1, 7, 7, 4)$ and the second sample could be $x^{*2} = (x_4, x_2, x_3, x_3) = (1, 7, 2, 2)$ and so on. Hence, we get new samples as

$$Original\ \ Sample\ \ x = (x_1, ..., x_N),$$
$$New\ \ Sample\ \ x = (x_1^*, ..., x_N^*), \tag{3.43}$$

To estimate the parameter θ, we apply the bootstrap procedure by combining the entire data coming from watermarked and those not watermarked. We denote $(X_1, ..., X_m)$ as data not watermarked H_0 and $(Y_1, ..., Y_n)$ as watermarked data H_1 where $m + n = N$.

$$(X_1, ..., X_m; Y_1, ..., Y_n)$$

$$
\begin{aligned}
B_1 &: (X_1^{*b_1}, ..., X_m^{*b_1}; Y_1^{*b_1}, ..., Y_n^{*b_1}) \Rightarrow \hat{\theta}^{b_1}, \\
B_2 &: (X_1^{*b_2}, ..., X_m^{*b_2}; Y_1^{*b_2}, ..., Y_n^{*b_2}) \Rightarrow \hat{\theta}^{b_2}, \\
&\vdots \qquad \vdots \\
B_M &: (X_1^{*b_M}, ..., X_m^{*b_M}; Y_1^{*b_M}, ..., Y_n^{*b_M}) \Rightarrow \hat{\theta}^{b_M}.
\end{aligned} \tag{3.44}
$$

In each bootstrap replicate $B_i, i = 1, ..., M$, we estimate the parameter set of the combined data $\hat{\theta}^{b_i}$, and obtain the final estimate of link function parameters as

$$\hat{\theta}_M = \frac{1}{M} \sum_{i=1}^{M} \hat{\theta}^{b_i}. \tag{3.45}$$

Since Monte Carlo bootstrap is based on law of large numbers, we expect the estimate to be consistent and that the following expression holds for very small value ϵ

$$\lim_{N \to \infty} P[\theta - \epsilon < \hat{\theta}_N < \theta + \epsilon] \to 1. \tag{3.46}$$

3.6.2 Bootstrap Hypothesis Testing of Errors

To estimate type 1 error using bootstrap method, we first put all the data together, estimate the parameter $\hat{\theta}_M$ as described above, and then use only those data not watermarked to test type 1 error. We note that non-watermark data used in the classifier link function are those not selected during the bootstrap resampling. This procedure is similar to dividing our samples into training and testing data set as illustrated in the Figure 3.9 below.

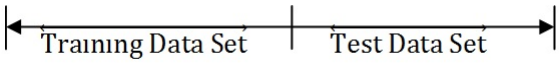


Figure 3.9: Data partitioning for evaluating the test statistics.

We use the training data to build the test statistics (i.e. estimate the parameters of link function) and then use the independent test data to evaluate its accuracy. The bootstrap re-sampling of type 1 error for each replicate B_i

$$\hat{P}_{fa}^{b_i} = \frac{1}{K} \sum_{j=1}^{K} \mathbb{1}(\hat{r}(x_j | \hat{\theta}^{b_i}) > \gamma_\alpha), \tag{3.47}$$

where K are those data not selected, and final bootstrap of type 1 error estimate becomes

$$\hat{P}_{fa}^{Boot} = \frac{1}{M}\sum_{i=1}^{M}\hat{P}_{fa}^{b_i}. \tag{3.48}$$

Similarly, we estimate type II error using test data from the alternate hypothesis, then yielding

$$\hat{P}_{m}^{b_i} = \frac{1}{K}\sum_{j=1}^{K}\mathbb{1}(\hat{r}(x_j|\hat{\theta}^{b_i}) < \gamma_\alpha). \tag{3.49}$$

K again are those data not selected during bootstrap re-sampling of alternate hypothesis, and final bootstrap type 1 error estimate becomes

$$\hat{P}_{m}^{Boot} = \frac{1}{M}\sum_{i=1}^{M}\hat{P}_{m}^{b_i}. \tag{3.50}$$

Finally, the probability of detection of watermark is obtained as

$$P_d = 1 - \hat{P}_{m}^{Boot}. \tag{3.51}$$

It is worth noting that since the data of hypothesis H_1 is generally larger than hypothesis H_0, dividing the corresponding densities or directly estimating the density ratio will always result in very large values. To obtain a reasonable density ratio model for type 1 and II errors, we invert the density ratios in each case, and also invert the critical threshold γ_α. The probabilities of false alarm and probability of detection in (3.48) and (3.50) can be rewritten respectively as

$$\frac{1}{\hat{P}_{fa}^{b_i}} < \frac{1}{\gamma_\alpha}, \tag{3.52}$$

$$\frac{1}{\hat{P}_{m}^{b_i}} > \frac{1}{\gamma_\alpha}. \tag{3.53}$$

3.7 Simulation Studies

In this Section, we conduct simulation studies of the proposed model starting with the estimation of parameters of the density ratio (link function) which is used as our likelihood ratio test in watermark detection. As already stated, accurate estimation of parameter vector $\hat{\theta}_N$ is crucial in determining a very good test statistics that is used for classification of our watermark model. The first step of simulation is to use the Monte Carlo

simulation method precisely bootstrap approach described in Section 3.6.1 to obtained the parameter estimates. Furthermore, a model misspecification simulation analysis is carried out. This simulation shows the core results and reason behind this research. We have already seen similar results in Section 3.3 when we use Gaussian mixture model to show that true distributions of greyscale image data is not really Gaussian. We also perform some simulations to study the least square approximation of exponential tilt model to Gaussian mixture model discussed in Section 3.5. The idea is to prove that approximated exponential tilt model can be used as test statistics which is easy and improved detection probability than using mixture model or linear correlation. Finally, we used the receiver operation characteristics (ROC) to check the performance measure of our watermark detector built using the model described in Section 3.1 to 3.6.

3.7.1 Maximum Likelihood Estimation of Density Ratio Model vs Sample Size

The analytical ML estimate model for density ratio estimation is described in Section 3.4.2. Since we know that our estimate $\hat{\theta}_N$ is a consistent estimator as $\lim_{N\to\infty}$, it is easy to see that we will perform our simulation as a function of N called the sample size. In this simulation, we use artificial univariate Gaussian random variables examples given in Section 3.4.3. The focus is to perform experiments both on quadratic density ratio model and linear density ratio model.
Quadratic density ratio model where $x_1, ..., x_m \backsim \mathcal{N}(20, 2^2)$ and $x_1, ..., x_n \backsim \mathcal{N}(20, 4^2)$ has the form $\exp(\alpha + \beta_1 x + \beta_2 x^2)$. Recall that the true parameter $\theta_o = (37.5, -3.75, 0.09375)$

Applying the parameter estimation technique in bootstrap method, we obtain the estimates of parameters with increasing number of data from $n = 50$ to $n = 2000$. In each run of simulation, we repeat the experiment $M = 1000$ and then compute the $\hat{\theta}$ as the estimate. Table 3.1 below shows these estimates as data n increases, and we can see that the estimates starts converging stochastically to the true parameter as n increases which satisfies the consistence property.

	n=50	n=500	n=1000	n=2000
$\hat{\alpha}$	48.8140	42.2694	41.2731	36.2225
$\hat{\beta_1}$	-4.8694	-4.2577	41.2731	-3.7010
$\hat{\beta_2}$	0.1193	0.1054	0.1044	0.0929

Table 3.1: Parameter estimation of quadratic density ratio model

Similarly, we performed experiment on the linear density ratio model of our example in Section 3.4.3. Linear density ratio model where $x_1, ..., x_m \backsim \mathcal{N}(0, 4^2)$ and $x_1, ..., x_n \backsim \mathcal{N}(20, 4^2)$ has the form $\exp(\alpha + \beta_1 x)$, where the true parameter is $\theta_o \in (-12.5, 1.25)$. The estimation for this linear model is summarized in the Table 3.2

	n=50	n=500	n=1000	n=2000
$\hat{\alpha}$	-232.6166	-92.3623	-13.5714	-11.9917
$\hat{\beta_1}$	22.065	9.8152	1.4363	1.2223

Table 3.2: Parameter estimation of Linear density ratio model

Therefore, we clearly see that our estimates converges and can be used in the link function as the test statistics in the situation where $f(x)$ and $g(x)$ are distributed according to Gaussian distributions. This function can be comparable to the linear correlation test statistics due to the fact that both require sample data to be large for the detector to perform well. Also noticing that linear correlation is based on sufficient statistics, we see that both linear correlation and link function model do not require individual estimation of density functions of both hypotheses to perform well. The obvious improvement using link function is that we can approximate it to the case where the underlying data is not really Gaussian distributed.

3.7.2 Least Square Approximation Error vs Dimension of Link Function

In this Section, we perform simulation study to compute the least square approximation error versus dimension of our logistic model. We take data to be $f(x) \backsim \frac{1}{2}\mathcal{N}(2, 1) + \frac{1}{2}\mathcal{N}(4, 1)$ and $g(x) \backsim \frac{1}{2}\mathcal{N}(2, 0.5) + \frac{1}{2}\mathcal{N}(4, 0.5)$. As shown in Section 3.5, the idea is to approximate the density ratio of this model by link function $r(x|\theta)$. This is done by varying the dimension of $r(x|\theta)$ from $d = 2, ...$ until we obtain the dimension

d^* which gives minimum mean square error. To do this, we estimate type I and II errors of our models, find the d^* in each case that gives the best dimension appropriate for test statistics $r(x|\theta)$. The number of observations used for this experiment is fixed at $N = 1000$. Figure 3.10 shows the plot of mean square error of the original density ratio versus the input dimension of the link function for probability of false alarm. One can see that dimensionality of link function needed to approximate the logistic model to the simulated Gaussian distribution is at $d^* = 8$. This gives the minimum MSE of the least square model and can be used to obtain the critical threshold γ_α needed for watermark detection.

In Figure 3.11, we observe the plot of density ratio models estimated by the logistic regression. We see that at $d = 2$, the model gives large error which is 0.3125. The shape of density model at this point is Gaussian type shape. This is in agreement with our discussion in Section 3.5 where we stated that the optimal dimension for univriate Gaussian density is at $d = 2$. Hence, we cannot model the Gaussian mixture data with link function when $d = 2$. The third plot in Figure 3.11 shows density ratio plot of logistic regression model when $d^* = 8$. This is found to have minimum mean square error of 0.0292, we see that is it comparable to the true density ratio model in (a). Hence we conclude that this link function is the best we can use as test statistics for the Gaussian mixture model at hand. For the figure in (c), we see clearly that the shape starts deviating slightly away from the true model and the MSE starts getting bigger as well.

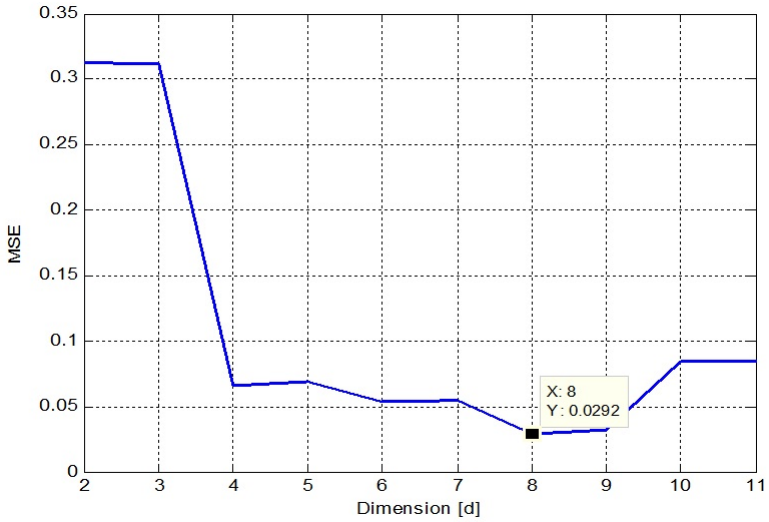


Figure 3.10: Least square approximation error vs dimensionality of link function for P_{fa} .

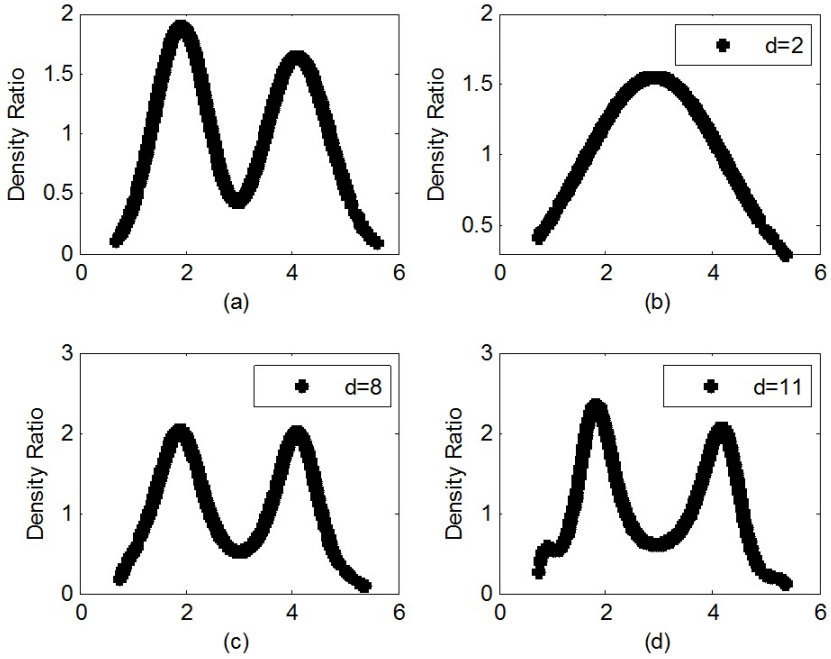


Figure 3.11: The plot of density ratio estimated by Logistic regression model. (a) True density ratio model. (b) Exponential density ratio at $d = 2$ (c) $d^* = 8$ and (d) d=11 .

The corresponding parameter estimated at this dimension $d^* = 8$ is shown in the

Table 3.3

$\hat{\theta}_{N=1000}$	$\hat{\alpha}$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$
$Values$	-35.9	-136.1	-224.9	203.5	-107.5	33.8	-6.2	0.6	-0.025

Table 3.3: Parameter estimation of exponential density ratio model when $d^* = 8$

Similarly, for type II error which is probability of miss detection, we observe that the dimension required in link function that gives the minimum error is at $d^* = 9$ (see Figure 3.12). Therefore, we can apply an exponential link function with $d^* = 9$, where the corresponding estimated parameters for this model is given in Table 3.4 as well.

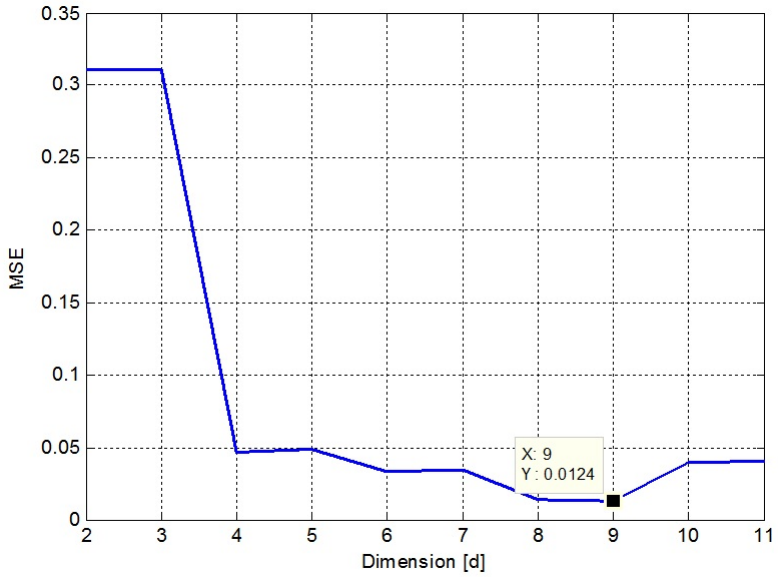


Figure 3.12: Least square approximation error vs dimensionality of link function for P_m .

$\hat{\theta}_{N=1000}$	$\hat{\alpha}$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$	$\hat{\beta}_7$	$\hat{\beta}_8$	$\hat{\beta}_9$
$Values$	-51.78	209.4266	-364.3	346.9	-196.4	68.448	-14.7	1.90	-0.13	0.004

Table 3.4: Parameter estimation of exponential density ratio model when $d^* = 9$

Furthermore, we run simulations for different artificial data under hypothesis H_0 used to obtain the probability of false alarm. The aim is to apply least square approximation method to different Gaussian mixture model data set and find the optimal dimension in each case. Table 3.5 shows that for different Gaussian mixture models randomly

generated for simulation purpose only, that the optimal dimension required in exponential link function is $d^* = 8$.

GMM	$Minimum\ \ MSE$	d^*
$f(x\|H_1) \backsim 0.5\mathcal{N}(4,2) + 0.5\mathcal{N}(6,2)$		
$g(x\|H_0) \backsim 0.5\mathcal{N}(4,0.5) + 0.5\mathcal{N}(6,0.5)$	0.0281	8
$f(x\|H_1) \backsim 0.5\mathcal{N}(20,9) + 0.5\mathcal{N}(35,9)$		
$g(x\|H_0) \backsim 0.5\mathcal{N}(20,6) + 0.5\mathcal{N}(35,6)$	0.0022	8

Table 3.5: Mean square error vs dimension of logistic regression model for randomly picked Gaussian mixture models under H_0

To understand the behaviour of our least square approximation method under hypothesis H_1, we conduct similar experiments focusing on watermarked data. The results summarized in Table 3.6 is obtained for this case. Again we see that optimal dimension is around $d^* = 8$, sometimes we can obtained optimal $d^* = 9$. We conclude that the dimension of our exponential model test statistics required for Gaussian mixture model distributed features is around $d^* = [8, 9]$. Obviously this is an approximation technique, getting exact dimension without error is very difficult, and hence we conjecture that approximating Gaussian mixture model with $d^* = 8$ or 9 will be a good model for test statistics.

GMM	$Minimum\ \ MSE$	d^*
$f(x\|H_1) \backsim 0.5\mathcal{N}(4,2) + 0.5\mathcal{N}(6,2)$		
$g(x\|H_0) \backsim 0.5\mathcal{N}(4,0.5) + 0.5\mathcal{N}(6,0.5)$	0.0159	8
$f(x\|H_1) \backsim 0.5\mathcal{N}(20,9) + 0.5\mathcal{N}(35,9)$		
$g(x\|H_0) \backsim 0.5\mathcal{N}(20,6) + 0.5\mathcal{N}(35,6)$	0.0122	8

Table 3.6: Mean square error vs dimension of logistic regression model for randomly picked Gaussian mixture models under H_1

3.7.3 Density Ratio Model Probability of Detection vs Sample Size

To understand the effect of the estimated density ratio model on probability of detection P_d, we simulate the P_d of density ratio model (logistic model) and compare it with P_d of the density ratio when individual densities are estimated independently (linear

correlation detector). We vary the sample size of data $x_1, ..., x_m|H_0 \backsim \mathcal{N}(20, 2^2)$ and $x_1, ..., x_n|H1 \backsim \mathcal{N}(20, 4^2)$ from $N = 10$ to $N = 5000$. The Monte Carlo bootstrap method described in Section 3.6.2 is used to obtain the critical threshold. Figure 3.13 (a) shows the plot of type 1 error and the critical threshold for this problem is approximately $\gamma_\alpha = 1.2$ for significance level $\alpha = 0.05$. Finally, we use the obtained critical threshold (control limit) γ_α to test type 2 error which is given as $P_r(r(x|\hat{\theta}) < \gamma_\alpha|H_1)$. To estimate likelihood ratio test $r(x|\hat{\theta})$ for type 2 error, we use the parameter $\hat{\theta}$ estimated from all the data put together and apply only watermarked data estimate to get type 2 error. In Figure 3.13 (b), we plot corresponding power of the test versus sample size N. The probability of detection increases with increasing sample size and we see clearly that the link function based on exponential model perfectly fits the true model estimated.

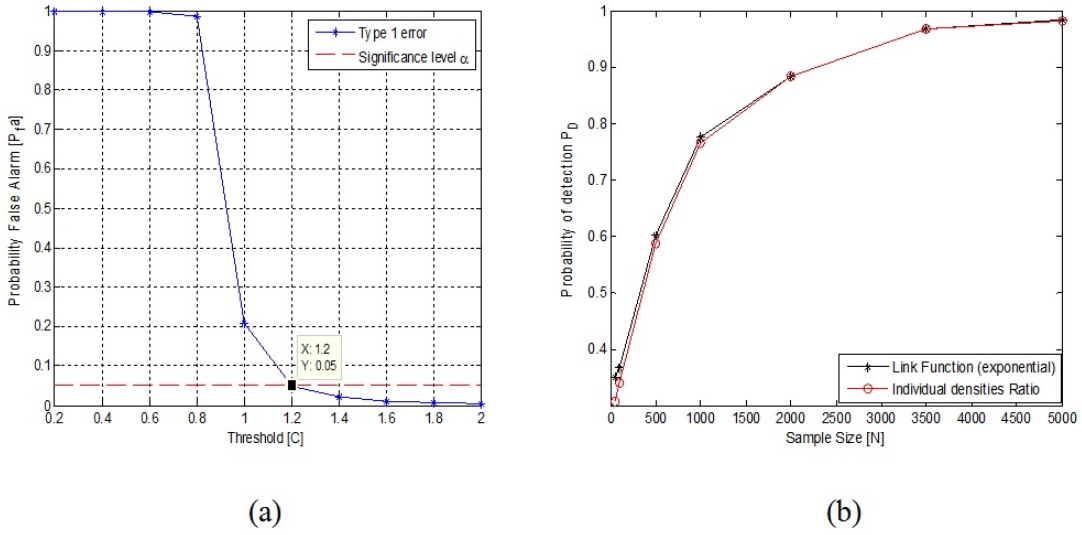


Figure 3.13: Probability of detection versus sample size. (a) Critical threshold with siginifican level $\alpha = 0.05$. (b) Corresponding power of the test.

3.7.4 Model Misspecification vs Sample Size

The model misspecification as applied to the context of detection theory in this research is choosing wrong density estimation model for the underlying distribution of the data. We have already stated that correctly estimating the distributions of the underlying density of data can improve the watermark detection. For comparison purpose, we

apply the standard nonparametric estimation technique called the kernel density function (KDE) which is given as $\hat{f}_{Ker}(x) = \frac{1}{n}\sum_{i=1}^{n} K_b(x - X_i)$ for $f(x)$ and $\hat{g}_{Ker}(x) = \frac{1}{m}\sum_{i=1}^{m} K_b(x - X_i)$ for $g(x)$. Parameter b is the bandwidth according to Silvermans rule of thumb given as $b = 1.06\hat{\sigma}n^{-1/5}$ and $K(.)$ used here is the classical Kernel Gaussian. We know that KDE can capture perfectly the shape of the distribution of data when n is large enough and perform sub-optimally when n is small. To check the performance of our parametric detector based on the link function, we perform experiments by applying it to different set of artificial data. Firstly, we generate data according to Gaussian distributions and then another set of data according to Gaussian mixture model. We applied least square approximation to get the optimal dimension of Gaussian mixture model at $d^* = 8$

The probability of detection computed by the bootstrap method in Section 3.6.2 is used as performance measure. We perform experiment for $N = 10$ to $N = 1000$ and compare KDE with link function probability of detection. In this experiment, if we assume that the actual individual densities are Gaussians. We use a simple Gaussian case where $f(x) \backsim \mathcal{N}(1, 1)$ and $g(x) \backsim \mathcal{N}(1, 0.5)$. In the case where we assume that actual densities are Gaussian mixture model we take the functions as $f(x) \backsim 0.5\mathcal{N}(1, 2) + 0.5\mathcal{N}(4, 2)$ and $g(x) \backsim 0.5\mathcal{N}(1, 0.5) + 0.5\mathcal{N}(4, 0.5)$. A Monte Carlo realization of $M = 1000$ is applied to each sample size $N = 10$ to $N = 1000$ and averaged to obtain detection probability of watermark. Shown in Figure 3.14 is the probability of detection versus sample size when the actual distribution of data is Gaussian distributed in (a) and when it is Gaussian mixture model (b). From Figure 3.14, we made the following observations.

- In the Gaussian distributed features, as N increases to 1000, both detectors built based on KDE and logistic function tend to 1 which is expected as increasing number of samples, increases both probability of false alarm as well as probability of detection.

- In the Gaussian distributed individual density case in (a), the logistic model detector with dimension $d = 2$ outperforms the kernel based detector. This is also expected as well, because logistic based detector is optimal for Gaussian case model. On the other hand, when the distributions are distributed according to complicated Gaussian mixture model, the KDE detector outperforms the logistic detector with $d = 2$. This is attributed to the fact that KDE can capture the shape of the underlying distribution better than parametric logistic model, but logistic detector with

$d = 8$ outperforms KDE especially for smaller sample size. Therefore, we see the need to design a detector that takes into account the distribution of data so that the detection probability is improved.

- Finally, Figure 3.14 shows as well that for small sample size, logistic based detector with both $d = 2$ and $d = 8$ performs better than KDE. It is due to the fact that KDE is not suitable for small sample size, even in the case where the density model is Gaussian mixture model, logistic detector performs reasonably well comparing to KDE.

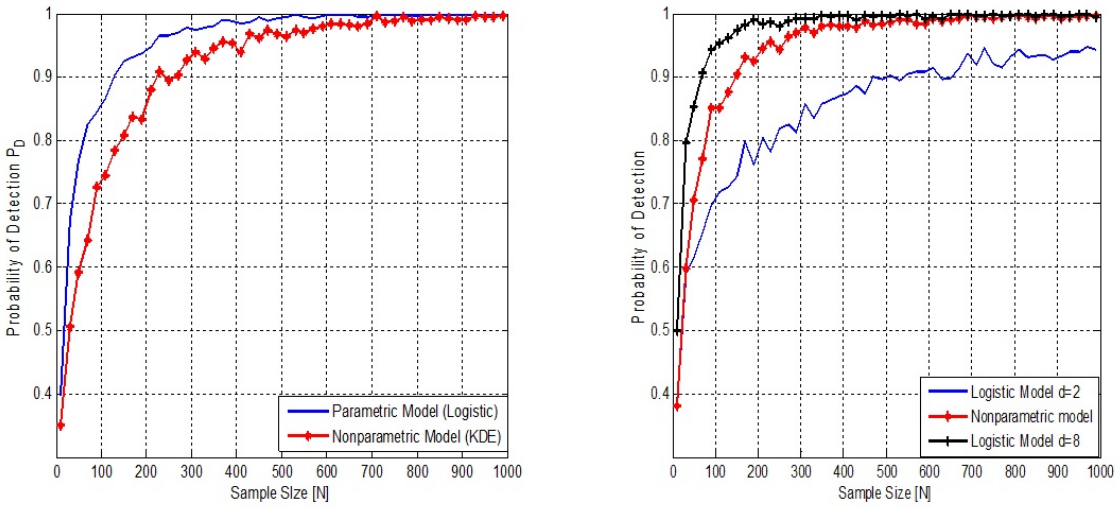


Figure 3.14: Model misspecification: probability of detection of watermark versus sample size. (a) Gaussian distributed features. (b) Gaussian mixture model distributed features .

3.7.5 Receiver Operating Characteristic of Approximated Link functions

The receiver operating characteristics normally known as ROC curve is a graphical representation of performance evaluation of classifier commonly used in signal detection analysis. The ROC curve shows the trade off between the probability of detection and probability of false alarm. To apply this curve to check the quality of our test statistics,

we vary the threshold to obtain the false alarm probability and for each case plot the detection probability versus P_{fa}. The values of the P_{fa} used to obtain threshold set are in the range of $P_{fa} = 0.01$ to 0.99 with the step size equal to 0.01. We use the same artificial data applied in Section 3.7.3 where under null hypothesis the data is given as $g(x) \backsim 0.5\mathcal{N}(2, 0.5) + 0.5\mathcal{N}(4, 0.5)$ and under the alternate hypothesis the data is $f(x) \backsim 0.5\mathcal{N}(2, 1) + 0.5\mathcal{N}(4, 1)$. The sample size is fixed at $N = 1000$. We know that from Section 3.7.3 that under H_0 and H_1 the dimensionality of exponential density ratio for this model that yields the minimum least square errors are $d^* = 8$ and $d^* = 9$ respectively. Also for the univariate single mode Gaussian case $d^* = 2$. Hence, we compare the performance of detectors built using exponential link function with $d^* = 8$,9 for bimodal Gaussian mixture and $d^* = 2$ for single mode Gaussian data using ROC curve. Note also that we applied univariate kernel density estimation (KDE) for correspondences. The kernel bandwidth applied is an arbitrary value close to Silvermans rule with Gaussian kernel. There are several methods to simulate the ROC performance curve. One such method is to apply the bootstrap Monte Carlo simulation described in Section 3.6.1. Another method is to apply cross validation where the data is partitioned into two sets $N = N_{Train} + N_{Test}$, the N_{Train} is used to build the density ratio and N_{Test} used to test it.

The observations from Figure 3.15 shows that when appropriate dimensions of the link function is used which corresponds to the case where the distributions of the original data is used. Watermark signals can be detected better than the case when density ratio is wrongly modelled as a single mode Gaussian. Also notice that kernel based detector is very close to the parametric density ratio model with $d^* = 8, 9$ which confirms that that modelling density ratio with an optimal d^* is good assumption.

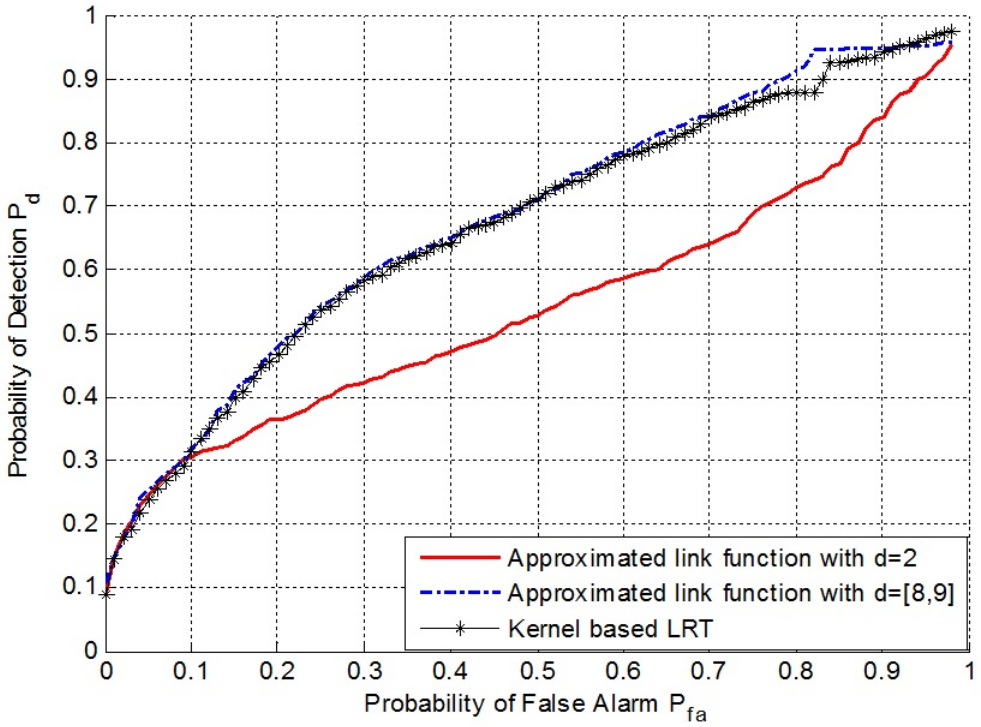


Figure 3.15: The ROC curve of detectors of watermark based on exponential density ratio models with different dimensionalities and kernel based likelihood ratio test.

3.8 Conclusion and Future Works

In this Section, we conclude the approximated parametric inference used as likelihood ratio test. According to conducted simulations, most data such as image data watermarked are not distributed according to unimodal Gaussian density. Therefore, applying linear correlation detector works sub-optimal. Designing a detector which incorporates the close enough characteristics distribution of the original data helped to improve watermark detection. The Gaussian mixture model (GMM) is a non-Gaussian parametric model applied as true characteristics. The advantage of GMM is that it can adapt to different models such as unimodal and multimodal Gaussians.

In Section 3.4, we have seen that an individual multimodal density can be totally avoided by estimating the density ratio directly. The density ratio which is the link function between watermark and non-watermark data is replaced with exponential fam-

ily model. The simulation in Section 3.7.1 shows that parameters of this model can be estimated accurately. The irreducible error of this parametric inference is simulated in Section 3.7.3 and we clearly see that this model with its irreducible error can perform better than linear correlation detector when the underlying data is not Gaussian distributed. This is observed in ROC curve simulations conducted in Section 3.7.4.

Future Work: Some of the future work for approximated parametric inference described in this Section are summarised below

- The optimal dimension d^* of logistic regression model should be estimated directly without going through sub-optimal approach of least square approximation as in our case.
- A generalized model which includes non-exponential family member should be developed. In this regard, estimating nonparametrically the link function should be considered. The advantage of such approach is that for models which fall outside the exponential family, this nonparametric generalized model would capture it perfectly. Hence, this will eliminate the modelling error completely.

Chapter 4

Semiparametric Based Watermark Detector

In Chapter 2, we briefly introduced the semiparametric method for estimating the density of watermarked data. We learnt that in many applications of watermark that no knowledge of the distribution of the underlying data is available during detection. Hence, applying a semiparametric estimation technique to obtain the distribution of the data seems appropriate. This is an ideal candidate for blind watermark detection, where we lack the knowledge of the underling distribution of data. It is also noted that semiparametric model is a good compromise between parametric and nonparametric models. This is because it takes care of model misspecifications commonly experience in parametric model and slow convergence rate that is predominant in nonparametric systems. In Section 3.4, we introduced a parametric density ratio model via exponential family model. This parametric model focuses on estimating density ratio known in detection paradigm as likelihood ratio test directly and avoids individual densities estimation. The density ratio is set as a known parametric function called the link function given as $r(x;\theta) = exp(\alpha + \beta^T b(x))$ where $\theta = (\alpha, \beta)$, so we simply try to estimate θ correctly and use least square to approximate the misspecified model.

4.1 Modified Weighted Kernel Estimator

The modified weighted kernel estimation is two step process algorithms. First, we estimate the parametric part of the process using density ratio logistics model estimation $r(y;\theta)$, and secondly kernel density estimation which depends on the parametric ratio as

the second process. If we denote $f(x)$ and $g(x)$ as probability densities under alternate and null hypothesis respectively, then we can combine data coming from each hypothesis to estimate our semiparametric model term modified weighted kernel estimator. This is illustrated in the Figure 4.1

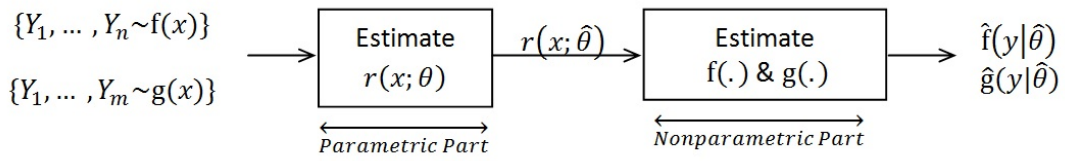


Figure 4.1: Block diagram illustrating semiparametric based watermark detector.

In semiparametric inference, we assume that the density ratio can be modelled by a certain function $r(x;\hat{\theta})$ with exponential family which is link function between $f(.)$ and $g(.)$ .This link function depends on two unknown densities, so we have set of unknowns $\{\alpha, \beta, f(.), g(.)\}$ all to be estimated. This is why the semiparametric approach is more difficult than parametric counterpart. Nevertheless, if they are estimated accurately, we will have an advantage of getting the link function $r(x;\theta)$ as well as individual densities $\hat{f}(x|\hat{\theta})$ and $\hat{g}(x|\hat{\theta})$. Therefore, except getting the density ratio semiparametric method gives the shape of each class under H_0 and H_1. The main idea behind this estimator is that since both data coming from H_0 and H_1 are linked together (i.e. there is information of source $g(.)$ in source $f(.)$, we can take advantage of that combine data together to estimate our functions.
Let the density ratio (likelihood ratio test) of our model be given as

$$\frac{f(x)}{g(x)} \triangleq r(x|\theta^*), \tag{4.1}$$

where θ^* denotes true parameter $r(x|\theta^*) = \exp(\alpha^* + \beta^{*T} b(x)), b(x)$ is a known function (see Section 3.4.1) and $\theta^* \in (\alpha^*, \beta^*), \theta^* = \Theta \subset \mathbb{R}^d$.
The function under hypothesis H_1 can then be

$$f(x) \triangleq g(x) \exp(\alpha^* + \beta^{*T} b(x)), \tag{4.2}$$

The equal by definition sign $\triangleq$ indicates that for equality to hold, function $f(x)$ depends on correct estimation of the term in right-hand-side which involve the link functions $exp(.)$. Before proceeding to modified weighted kernel estimator, let us first consider a conventional approach of estimating semiparametric models. Two steps naive process of estimating function $f(x)$ are

1. Estimate $g(x)$ by classical kernel density estimation (KDE) ignoring the fact that $g(x)$ and $f(x)$ are linked. Hence we get

$$\hat{g}(x) = \frac{1}{m}\sum_{i=1}^{m} K_b(x - X_i), \tag{4.3}$$

where $K_b(.) = K(./b)/b$ and $K(.) \geq 0$ is a kernel function that satisfies $\int_{-\infty}^{+\infty} K(v)dv = 1$.

2. Secondly, estimate the parameters of the parametric part $\exp(\hat{\alpha} + \hat{\beta}^T b(x))$. Hence, $\hat{f(x)}$ is obtained by plugging in $\hat{g(x)}$ into (4.2) to yield

$$\hat{f}(x) = \hat{g}(x)\exp(\hat{\alpha} + \hat{\beta}^T b(x)). \tag{4.4}$$

This naive process of estimating densities by KDE method is suboptimal because in most watermark detection process, individual data are always related either through their means or variances. Even though similarity property is taken into account in the parametric part, we lost lots of information of their dependence in the nonparametric part because data are treated completely independent. The second efficient method called modified weighted kernel estimator takes into account the density ratio and incorporates it into the nonparametric part.

4.1.1 Parametric Inference

We start by estimating the parametric part of the semiparametric problem. This estimator has been built in Section 3.8 and fitted using least square method. Let the entire data under both hypotheses be represented as

$$(X_1, ..., X_m, Y_1, ..., Y_n) = (x_1, ..., x_N), \tag{4.5}$$

where $n + m = N$.
As usual, we apply least square estimator to obtain an optimal dimension d^* as well as parameter θ^* that minimizes sum of square errors

$$E_{rr}(\hat{\theta}, \hat{d}) = \underset{\theta\in\Theta, d\in(D)}{\operatorname{argmin}} \frac{1}{N}\sum_{i=1}^{N}(L(x_i|\eta) - r_d(x_i|\theta))^2, \tag{4.6}$$

where $L(x_i|\eta)$ is the true density ratio model and $r_d(T_i|\theta)$ model used for fitting. Finally we obtain the parametric part of the semiparametric inference as

$$\frac{f(x_i)}{g(x_i)} \triangleq r(x_i|\theta^*), \tag{4.7}$$

This function is very important in the nonparametric part because it is going to appear in each individual density as a weight function. The parameter θ is estimated by maximum likelihood procedure explained in Section 3.4.2 which is equivalent to logistic model see [15] for details.

4.1.2 Semiparametric Inference

The idea behind this model is that since the functions $f(.)$ and $g(.)$ are linked, we put data together as in parametric part and assign weight function $\mathcal{U}_i(x_i), i = 1, , N$ in the kernel density formula such that $\mathcal{U}_i(.) \geq 0$, so that equation (4.3) becomes

$$\hat{g}_b(x) = \sum_{i=1}^{N} \mathcal{U}_i(x_i) K_b(x - x_i), \tag{4.8}$$

So weights depend on the link model, and for classical KDE the weight reduces to

$$\mathcal{U}_i(x_i) = \begin{cases} \frac{1}{m}, & i = 1, ..., m \\ 0, & otherwise. \end{cases} \tag{4.9}$$

Therefore, we can infer immediately that classical kernel density estimation is the special case of modified weighted kernel estimator (4.8) when we neglect data coming from source $f(.)$

Statistical Properties: The weight function $\mathcal{U}_i(x_i)$ completely depends on the statistical properties of estimator (4.8) such as mean and variance. The task is to obtain an optimal weight function $\mathcal{U}_i^*(.)$ by using unbiased estimator of (4.8) that minimizes the variance. We can either express the semiparametric properties either as mean square error or individually mean and variance. The MSE of this estimator is

$$MSE(\hat{g}_b(x)) = \mathbb{V}ar(\hat{g}_b(x)) + [\mathbb{B}ias(\hat{g}_b(x))]^2. \tag{4.10}$$

The bias is given as

$$\begin{aligned} \mathbb{B}ias(\hat{g}_b(x)) &= \mathbb{E}(\hat{g}_b(x)) - g(x) \\ &= \textstyle\sum_{i=1}^{N} \mathbb{E}(\mathcal{U}_i(x_i) K_b(x - x_i)) - g(x). \end{aligned} \tag{4.11}$$

Consequently, the first term in 4.11 becomes the expectation of $\hat{g}_b(x)$ given as

$$\mathbb{E}(\hat{g}_b(x)) = \sum_{i=1}^{N} \mathbb{E}(\mathcal{U}_i(x_i) K_b(x - x_i)). \tag{4.12}$$

Since the data are combined to get the estimator in (4.8), the individual term in (4.12) via Taylor expansion becomes

$$\begin{cases} \mathcal{U}_i(x)g(x) + \mathcal{O}(b), & 1 \leq i \leq m \\ \mathcal{U}_i(x)g(x)r(x|\theta) + \mathcal{O}(b), & m+1 \leq i \leq N. \end{cases} \tag{4.13}$$

Then, the overall combined expectation is written as

$$\mathbb{E}(\hat{g}_b(x)) = g(x)\left(\sum_{i=1}^{m}\mathcal{U}_i(x) + \sum_{i=m+1}^{N}\mathcal{U}_i(x)r(x|\theta)\right) + \mathcal{O}(b). \tag{4.14}$$

Similarly, the variance of the single term in (4.8) coming from each density source is given as

$$\begin{cases} \mathcal{U}_i^2(x)g(x)(1+\mathcal{O}(b)), & 1 \leq i \leq m \\ \mathcal{U}_i^2(x)\frac{\gamma}{b}g(x)r(x|\theta)(1+\mathcal{O}(b)), & m+1 \leq i \leq N, \end{cases} \tag{4.15}$$

where $\gamma = \int K^2(v)dv$.
Then, the empirical combined variance is

$$\mathbb{V}ar(\hat{g}_b(x)) = g(x)\frac{\gamma}{b}\left(\sum_{i=1}^{m}\mathcal{U}_i^2(x) + \sum_{i=m+1}^{N}\mathcal{U}_i^2(x)r(x|\theta)\right) + \mathcal{O}(b). \tag{4.16}$$

Optimal weight function $\mathcal{U}_i^*(.)$: To obtain the optimal weight function used in our estimate, we perform an optimization that minimizes the variance. The optimization problem statement is as follow
Define an objective function as

$$L = \sum_{i=1}^{m}\mathcal{U}_i^2(x) + \sum_{i=m+1}^{N}\mathcal{U}_i^2(x)r(x|\theta)). \tag{4.17}$$

The task is to minimize L

$$\mathcal{U}_i^*(.) = \underset{\mathcal{U}\in\Psi}{\operatorname{argmin}}\, L, \tag{4.18}$$

subject to the constraint

$$\sum_{i=1}^{m}\mathcal{U}_i(x) + \sum_{i=m+1}^{m}\mathcal{U}_i(x)r(x|\theta) = 1. \tag{4.19}$$

The constraint is an asymptotically unbiased estimator . In this case the optimal weight function is found to be

$$\mathcal{U}_i^*(.) = \frac{1}{m + nr(x|\hat{\theta})}. \tag{4.20}$$

Intuitively, we say that an optimal weight $\mathcal{U}_i^*(.)$ is the weight from the set Ψ that gives the smallest variance that satisfied the constraint. We clearly see that optimal weight depends on the parametric link model between densities under $H_0 \backsim g(x)$ and under $H_1 \backsim f(x)$. Therefore, the most important part of our semiparametric estimator is the optimal weight. If the estimated density ratio $r(x|\hat{\theta})$ is close enough to the true ratio $r(x|\theta^*)$, we can get an optimal weight $\mathcal{U}_i^*(x)$. In this case, the estimator of $\hat{g}(x)$ is more efficient than classical kernel estimator. But if the estimated density ratio is far away from the true ratio, we end up with classical kernel density estimation. Finally, the estimate of these densities under H_0 and H_1 are given respectively as

$$\hat{g}(x) = \sum_{i=1}^{N} \frac{1}{m + nr(x_i|\hat{\theta})} K_b(x - x_i) \tag{4.21}$$

and

$$\hat{f}(x) = \sum_{i=1}^{N} \frac{r(x|\hat{\theta})}{m + nr(x_i|\hat{\theta})} K_b(x - x_i). \tag{4.22}$$

Notice that density under H_1 need not to be estimated since it depends on $\hat{g}(x)$ and $r(x|\hat{\theta})$ (i.e. constant). This is an additional advantage of this estimator over KDE. We also notice that the parametric part $r(x|\hat{\theta})$ is present in both $\hat{g}(x)$ and $\hat{f}(x)$. Hence, we conclude that prior knowledge of relationship between data in H_0 and H_1 greatly help to get better estimate of $r(x|\hat{\theta})$ and $\hat{g}(x)$.

4.1.3 Bandwidth Parameter Selection

As already stated earlier, the bandwidth parameter b is very important parameter for smoothness of the semiparametric estimator developed in this thesis. Selecting b optimally will yield an estimator which is neither over-smooth nor under-smooth which in turn affects detection probability. Furthermore, another advantage of this estimator is that since $\hat{f}(x)$ is constant, we need to only select optimally the bandwidth when estimating function $\hat{g}(x)$, then use it in $\hat{f}(x)$.

One of the most popular methods for bandwidth selection is Silvermans reference rule of thumb [12]. The idea is to choose values of b such that asymptotic mean integrated square error (MISE) is minimized. That way we can maintain a good variance-bias trade off. It is shown that when the Gaussian kernel $K(.)$ assumption is made, that the optimal bandwidth is given as

$$b = \hat{C} N^{-1/5}, \tag{4.23}$$

where $\hat{C} = 1.06\hat{\sigma}$, N is the sample size of data combined together and

$$\hat{\sigma} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2}. \tag{4.24}$$

Noting that

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{4.25}$$

is the empirical sample mean, hence we can employ a Monte Carlo approach to estimate these variables.

To verify the selected b and obtain the error made due to this estimator, we compute the MISE between $\hat{g}(x)$ and the true model $g(x)$ as follow

$$ISE(\hat{g}_b) = \frac{1}{N}\sum_{i=1}^{N}(\hat{g}_b(x_i) - g(x_i))^2. \tag{4.26}$$

Then MISE is obtained by computing

$$MISE(\hat{g}) = \frac{1}{N}\sum_{i=1}^{N} ISE(\hat{g}). \tag{4.27}$$

We vary bandwidth say $b = 0.03$ to 1 and obtain the MISE in each case. It is worth noting that the optimal bandwidth b^* used in our experiment is the one with minimum MISE. This bandwidth also gives the maximum possible probability of detection of watermark in our experiment.

Semiparametric Estimation Algorithm

The following algorithm is the procedure of estimating our semiparametric model called the modified weighted kernel estimator.

1. Generate domain set x under $H_i, i \in \{0, 1\}$, put data together to get $x_i, i = 1, ..., N$, choose a kernel function say Gaussian and finally the bandwidth b.

2. Estimate the parametric part as $r(x_i|\hat{\theta})$ and weight function $\mathcal{U}_i^*(.)$

3. For each x_i, evaluate the kernels at x

$$K_i = K\left(\frac{x - x_i}{b}\right), i = 1, ..., N \tag{4.28}$$

4. Normalize each kernel by $\frac{1}{b}$

5. Finally for each x, obtain the average of modified weighted kernel using the weight function $\mathcal{U}_i^*(.)$

4.1.4 Likelihood Ratio Test for Semiparametric Model

Having estimated correctly the shape of distributions under each hypothesis as defined in (4.21) and (4.22), the next step is to obtain the likelihood ratio test (LRT) or test statistics denoted as D_N and use it to determine the presence or absence of watermark. We regard D_N also as semiparametric based detector written as

$$D_N = \frac{\hat{f}(x)}{\hat{g}(x)}. \tag{4.29}$$

To detect watermark, we simply apply any of the detection criteria such as Bayes or Neyman-Pearson (NP) as described in chapter 2. When NP criterion is applied, the threshold is again obtained by

$$\gamma_\alpha = min\left\{\gamma : P\left(D_N > \gamma | H_0\right) \leq \alpha\right\}. \tag{4.30}$$

Therefore, we detect presence of watermark when $D_N > \gamma_\alpha$ and absence of watermark is obtained when $D_N < \gamma_\alpha$

Cross Validation Algorithm for Semiparametric LRT

The proper evaluation of test statistics D_N for this estimator is very crucial for detector based on semiparametric model. A good D_N requires using different set of data to build D_N and another set to test it. Cross-validation is very effective method to partition data into two sets called k training data and $N-k$ testing data [12]. The training set is denoted N_{Train} and testing set denoted as N_{Test}, such that $N_{Train} + N_{Test} = N$. Two test statistics are required to evaluate the efficiency of the proposed detector, one for non-watermark data denoted as $(D_N|H_0)$ used to obtain the threshold γ_α given α and another for watermarked data given as $(D_N|H_1)$. The computational algorithm for LRT is given below

Algorithm

1. Given observations $x_i \backsim H_0, i = 1, .., N$ select a value for k

2. Leave out k points from $x_i \backsim H_0$ to obtain the test data set N_Test

3. Use the remaining $N-k$ data to get the semiparametric estimators $\hat{g}(x)$ and $\hat{f}(x)$

4. For each of k point test data, evaluate the test statistics as

$$D_N|H_0 = \frac{\hat{f}(x)}{\hat{g}(x)}. \tag{4.31}$$

5. Repeat step 2 through 4 focusing on data under H_0

6. Similarly, by now focusing on data under $x_i \backsim H_1$, steps 2 through 5 will be repeated so that we get test statistics as

$$D_N|H_1 = \frac{\hat{f}(x)}{\hat{g}(x)}. \tag{4.32}$$

Before proceeding to simulation results, we shall state briefly the method of application of semiparamteric based watermark detector in commonly watermark embedding approaches already existed. Two most common watermark embedding approaches are one-bit watermarking via two pseudonoise sequences (PNS) and one-bit watermarking via one PNS [4]. For host features where a single bit watermark $m \in \{0,1\}$ is embedded using two PNS's, we generate the watermark message as follow

$$W_m = \begin{cases} W_0, & if \;\; m = 0 \\ W_1, & if \;\; m = 1, \end{cases} \tag{4.33}$$

where $W_0 = W_0[1], ..., W_0[K]$, and $W_1 = W_1[1], ..., W_1[K]$, for $W_j[i] \backsim (N)(\mu_w, \sigma_w^2), j = 0, 1; i = 1, ..., K$
The semiparametric detector in this case is applied as usual and the presence of watermark is checked by considering

$$\hat{m} = \begin{cases} 0, & if \;\; D_N(W_0) > \gamma_\alpha \\ 1, & if \;\; D_N(W_1) > \gamma_\alpha \\ None, & if \;\; max\left\{D_N(W_0), D_N(W_1)\right\} < \gamma_\alpha \end{cases} \tag{4.34}$$

is satisfied.

On the other hand for host features where only one PNS is generated to represent the watermark, we have the embedding process as

$$W_m = \begin{cases} -W_0, & if \ \ m = 0 \\ W_0, & if \ \ m = 1, \end{cases} \tag{4.35}$$

where $W_0 = W_0[1], ..., W_0[K]$. Then the watermark detection process is

$$\hat{m} = \begin{cases} 0, & if \ \ D_N(W_0) < -\gamma_\alpha \\ 1, & if \ \ D_N(W_0) > \gamma_\alpha \\ None, & if \ \ |D_N(W_0)| \leq \gamma_\alpha. \end{cases} \tag{4.36}$$

4.2 Simulation Studies

To understand the effect of the proposed semiparametric detector on the watermark applications, we investigate the advantages of this detector via simulations. We first use semiparametric estimation algorithm described in Section 4.3 to obtain individual densities coming from both sources under H_0 and H_1. The effect of bandwidth selection on the estimator is investigated in Section 4.2.1 and in Section 4.2.2, we explore the performance detection measure for this detector by obtaining the probability of watermark detection while varying the sample size of data.

4.2.1 Estimation Error vs Kernel Bandwidth

The artificial data used for simulation are $f(x) \backsim 0.5\mathcal{N}(2,1) + 0.5\mathcal{N}(4,1)$ and $g(x) \backsim 0.5\mathcal{N}N(2,0.5) + 0.5\mathcal{N}(4,0.5)$. We recall that estimating the received data accurately requires choosing a near optimal value of bandwidth parameter b. This bandwidth in our case is the one that maximizes the probability of detection as well as minimizing the MISE of the estimated data. To do this, we vary the bandwidth arbitrary from $b = 0$ to $b = 1$ and choose the bandwidth which gives the minimum MISE or maximum power of the test. We run simulations for MISE by repeating the number of trials $N = 1000$ times and then taking the average of the error as described in Section 4.2. To compare our proposed semiparametric estimator fairly with a nonparametric estimator, we run simulations on

same data in both cases. We know that applying a fully nonparametric estimator on these data ignores the fact that both data $f(x)$ and $g(x)$ are linked. This can actually affect the detection performance as we shall see. Figure 4.2 shows the estimation error of density ratios (LRT) for both semiparametric and nonparametric estimators. We clearly see that nonparametric estimator always yield higher error than semiparametric estimator for any bandwidth chosen. This is simply due to prior knowledge of relationships between watermarked and non-watermark data which is clearly observed in the semiparametric estimation. Second observation is that the bandwidth which maximizes our power of test is at $b^* = 0.21$ with minimum error 0.0037. This bandwidth b^* is close to the Silvermans rule bandwidth which for this case is $b = 0.3$ with error 0.0086. The conclusion is that the range of bandwidth necessary for obtaining high power of the test for these data set is from $b = 0.03$ to $b = 0.3$.

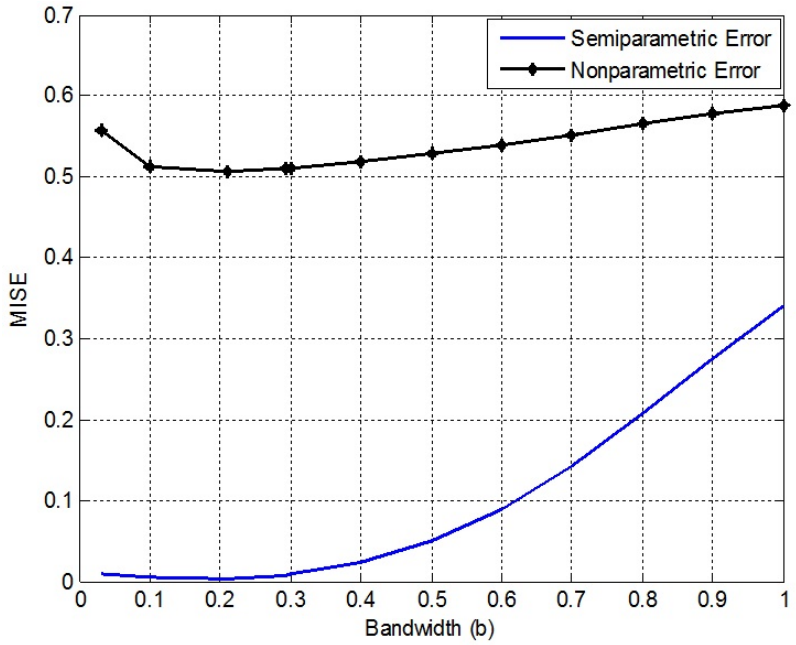
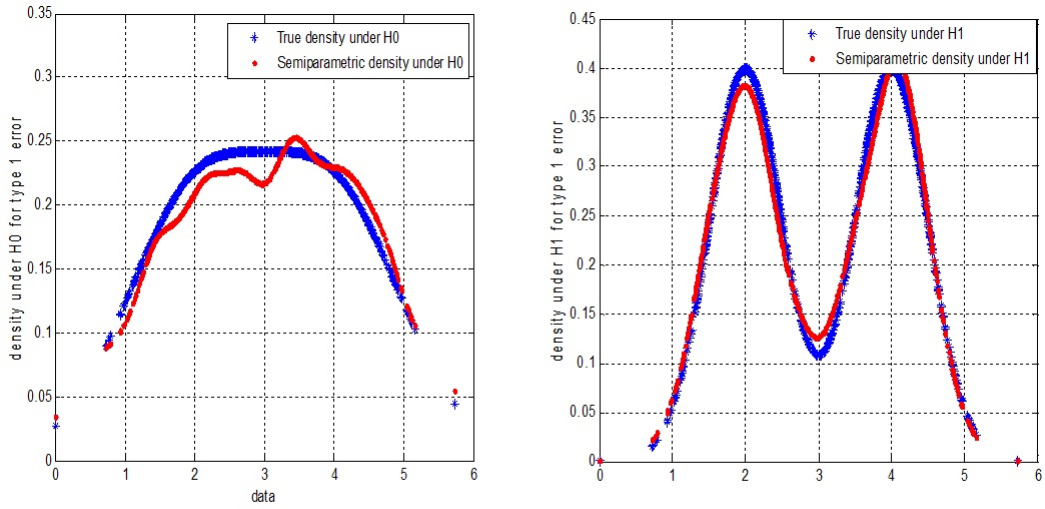


Figure 4.2: Estimation error in semiparametric and nonparametric estimators with varying kernel bandwidth.

We recall that semiparametric detector has the advantage of getting the density ratio directly as well as individual densities under the hypotheses H_0 and H_1. We display these estimated densities for the two most common errors encountered in watermark detection metric called type 1 error (false alarm probability) and type 2 error (miss detection probability). In Figure 4.3 , we can see that for type 1 error, the individual densities

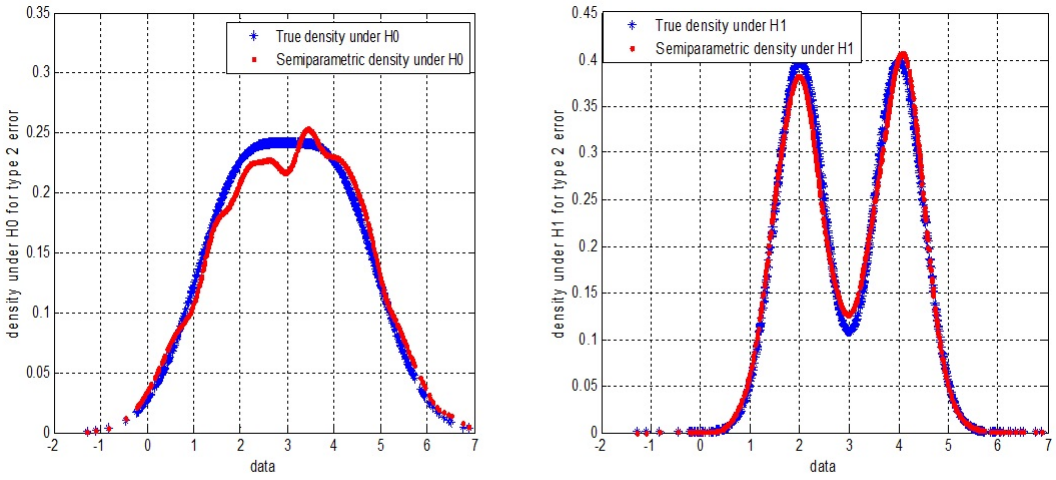
under H_0 in (a) and H_1 in (b) can be captured with high accuracy by the semiparametric model. Similarly, in Figure 4.4 below, the densities for type 2 error under H_0 in (a) and H_1 in (b) are displayed, and we clearly see that our semiparametric model almost captures the characteristics shape perfectly.



(a) (b)

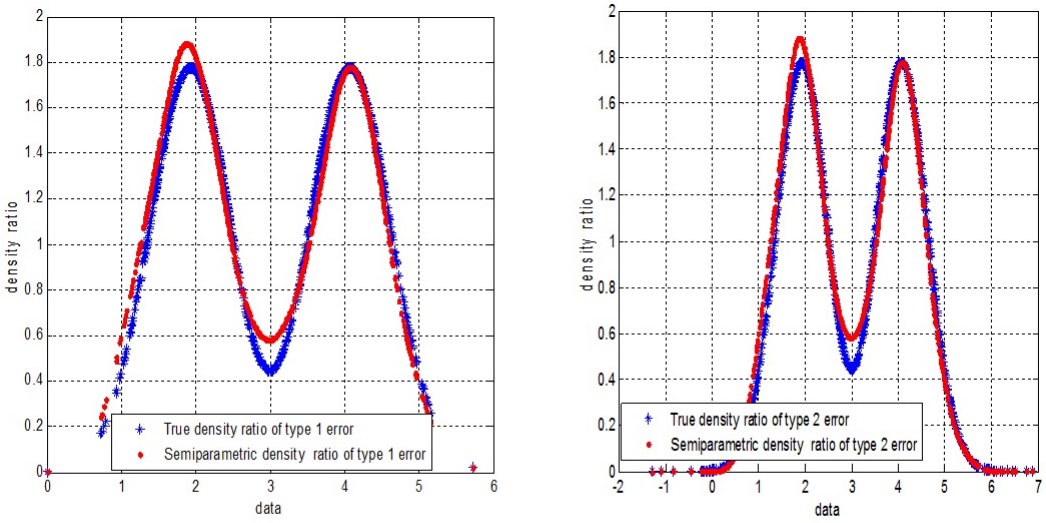
Figure 4.3: Individual densities of true model and semiparametric estimated model for type 1 error. (a)Densities under H_0. (b)Densities under H_1.

Next, we display the corresponding density ratios used to obtain the estimation errors described previously. In this regard, since it is important to obtain the density ratios both when watermark is present and when it is absent, we display these density ratios as seen in figure 4.5. Clearly, our estimated ratio is very close to the true model and hence can be used as likelihood ratio test during watermark detection.



(a) (b)

Figure 4.4: Individual densities of true model and semiparametric estimated model for type 2 error. (a) Densities under H_0. (b) Densities under H_1.



(a) (b)

Figure 4.5: Density ratios of true model and semiparametric estimated model . (a)Density ratio for type 1 error. (b)Density ratio for type 2 error.

4.2.2 Model Misspecification vs Sample Size

Having obtained individual densities and density ratios for semiparametric detector, the next task is to detect the presence or absence of watermark in a received data. Different detection measures such as probability of detection versus sample size and probability of detection versus probability of false alarm (ROC) can be applied. In model misspecification case, we try to fix the threshold by setting significance level $\alpha = 0.05$ and then use Monte Carlo simulation approach to obtain the critical threshold $\gamma_\alpha = 1.95$. this result is plotted in figure 4.6.

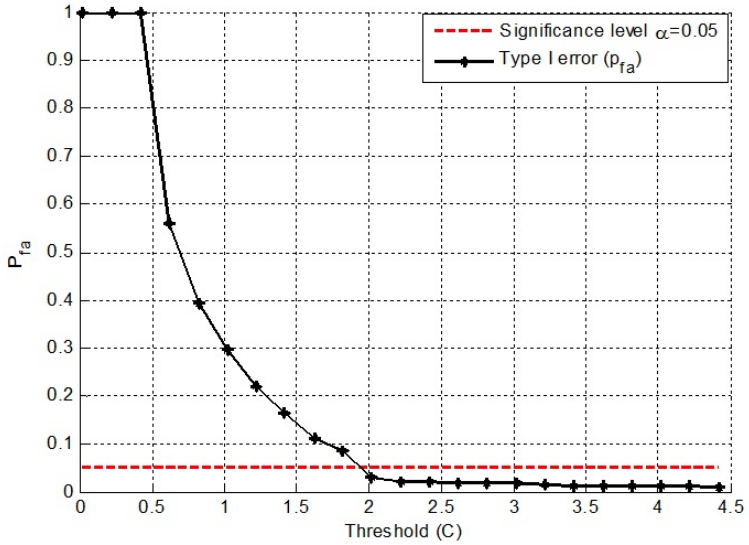


Figure 4.6: Critical threshold selection significant level $\alpha = 0.05$.

To understand how semiparametric model works as sample size increases, we conduct two sets of experiments. First, we assume perfect knowledge of the original data and assume that actual individual densities are unimodal Gaussian given as $f(x) \backsim \mathcal{N}(1, 1)$ and $g(x) \backsim \mathcal{N}(1, 0.5)$. For this case we applied parametric logistic model with dimension $d = 2$, semiparametric model and classical kernel density function (KDE). Secondly, we assume that individual densities are multimodal densities such as Gaussian mixture models given as $f(x) \backsim 0.5\mathcal{N}(1, 2) + 0.5\mathcal{N}(4, 2)$ and $g(x) \backsim 0.5\mathcal{N}(1, 0.5) + 0.5\mathcal{N}(4, 0.5)$. Also we applied parametric logistic model with dimension $d = 2$, semiparametric model and classical KDE. A Monte Carlo realization of $M = 1000$ is applied to each sample size $N = 20$ to 1000 with step size 20 and averaged to obtain detection probability of watermark. We plot in Figure 4.7 the probability of detection versus sample size when the

actual distribution of data is Gaussian distributed in (a) and when it is Gaussian mixture model (b).We know that the optimal detection model for the case in (a) is linear correlation or logistic model with optimal dimension $d = 2$, we see that linear correlation has the highest detection probability and our semiparametric based detector is very close to the linear correlation, but obviously kernel based detector perform poorly as expected. The semiparametric detector performs reasonably well because it incorporates the parametric inference properties as well as nonparametric inference especially for smaller sample sizes where fully nonparametric detector suffers. The second part of experiment plotted in Figure 4.7 (b) shows that semiparametric based detector outperforms both parametric and nonparametric detectors. As expected the linear correlation detector performs poorly since the underlying data is not Gaussian distributed. The interesting observation is that semiparametric detector outperforms the fully nonparametric detector for smaller sample size. Again this is the due to the compromise semiparametric model brings, and also it exploit the fact that both data under H_0 and H_1 are combined during estimation which is not the case in nonparametric inference.

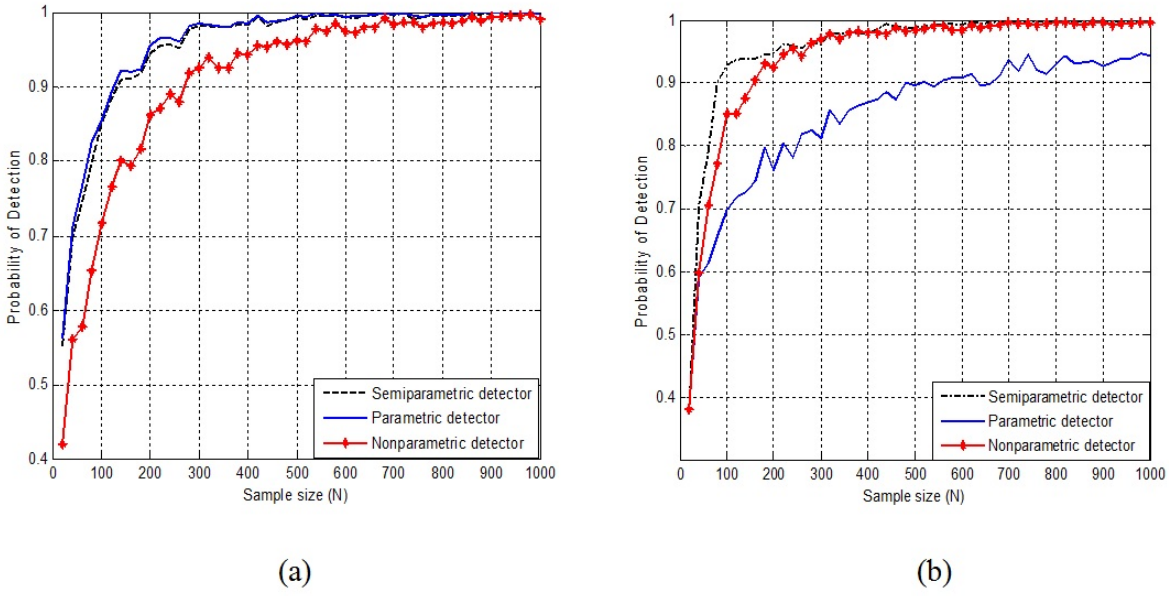


Figure 4.7: Model misspecification: probability of detection versus sample size .(a)Gaussian distributed features. (b)Gaussian mixture model distributed features.

4.3 Conclusion

In this Chapter, we have examined the semiparametric based watermark detection. One of the advantages of this detector is that it gives the density ratio directly as well as individual densities coming from both sources under H_0 and H_1. The hybrid nature of this semiparametric model is very important for watermark applications where the prior knowledge relationship between data under H_0 and H_1 are not known. The efficiency of this model totally depends on optimal weight function. If the prior knowledge of data is known, semiparametric detector works optimally, but if prior knowledge is not known, semiparametric model reduces to worst case nonparametric based detector. In Section 4.2.1, we have seen that the bandwidth kernel for this model is very close to normal reference rule. We have also seen that applying classical kernel density estimation when data under H_0 and H_1 are connected yields higher MISE error in nonparametric inference than semiparametric model. Both individual densitites and density ratios estimated using this model are very close to the true density of the original data.

A very important conclusion is that semiparametric based detector can work as well as linear correlation detector when the actual density of the received data is Gaussian distributed. When the true density is far from Gaussian distribution, semiparametric based detector outperforms linear correlation detector as seen in simulation studies. Another important advantage of this model over parametric inference described in Chapter 3 is its adaptive nature to fully nonparametric model when data are not connected. For instance, if an attack modified the distribution of watermarked data completely from nonwatermark data, semiparametric inference will still detect optimally which is difficult for parametric inference. This is robustness property of semiparametric based detector.

Appendix A

Appendices

A.1 Derivation of (2.32)

Let X and W be independent random variables, then the variance of product of two random variables is derived as

$$\begin{aligned}
\mathbb{V}ar(XW) &= \mathbb{E}(X^2W^2) - \mathbb{E}^2(XW) \\
&= \mathbb{E}(X^2)\mathbb{E}(W^2) - \mathbb{E}^2(X)\mathbb{E}^2(W) \\
&= (\underbrace{\mathbb{E}(X^2) - \mathbb{E}^2(X)}_{=\mathbb{V}ar(X)} + \mathbb{E}^2(X))(\underbrace{\mathbb{E}(W^2) - \mathbb{E}^2(W)}_{=\mathbb{V}ar(W)} + \mathbb{E}^2(W)) - \mathbb{E}^2(X)\mathbb{E}^2(W) \ . \\
&= \mathbb{V}ar(X)\mathbb{V}ar(W) + \mathbb{V}ar(X)\mathbb{E}^2(W) + \mathbb{V}ar(W)\mathbb{E}^2(X)
\end{aligned} \tag{A.1}$$

If $\mathbb{E}(X) = \mathbb{E}(W) = 0$, then

$$\mathbb{V}ar(XW) = \mathbb{V}ar(X)\mathbb{V}ar(W) \tag{A.2}$$

A.2 Derivation of (2.35)

The assumption is that U and W are linearly correlated by $U = \gamma W$, and both signal parameters are independent on X. Again following the method of calculating the variance of product of two random variables we have

$$\begin{aligned}
\mathbb{V}ar(YU) &= \mathbb{V}ar(XU + \alpha UW) \\
&= \mathbb{V}ar(XU) + \alpha^2\mathbb{V}ar(UW) + 2\alpha Cov(XU, UW). \ .
\end{aligned} \tag{A.3}$$

Starting with the last term on the right hand side in (A.3), the covariance can be derived as

$$\begin{aligned} Cov(XU, UW) &= \mathbb{E}(XWU^2) - \mathbb{E}(XU)\mathbb{E}(WU) \\ &= \mathbb{E}(X)\mathbb{E}(WU^2) - \mathbb{E}(X)\mathbb{E}(U)\mathbb{E}(UW) \\ &= \mathbb{E}(X)(\underbrace{\mathbb{E}(WU^2) - \mathbb{E}(U)\mathbb{E}(U)}_{=Cov(UW,U)}) \\ &= \mathbb{E}(X)Cov(UW, U). \end{aligned} \quad . \tag{A.4}$$

Using the fact that $U = \gamma W$, we can derive the covariance in (A.4) as follow

$$\begin{aligned} Cov(U, UW) &= Cov(\gamma W, \gamma W^2) \\ &= \gamma^2 Cov(W, W^2) \\ &= \gamma^2(\mathbb{E}(W^3) - \mathbb{E}(W)\mathbb{E}(W^2)). \end{aligned} \quad . \tag{A.5}$$

Finally, the last derivation is the second term of (A.3) which is the variance of the correlated random variables U and W,

$$\begin{aligned} \mathbb{V}ar(UW) &= \mathbb{V}ar(\gamma W^2) \\ &= \gamma^2 \mathbb{V}ar(W^2) \\ &= \gamma^2 \mathbb{E}(W^4) - \mathbb{E}^2(W^2). \end{aligned} \quad . \tag{A.6}$$

Substituting (A.5) into (A.4), then (A.4) and (A.6) into (A.3) yields the variance of (2.35) as

$$\begin{aligned} \sigma^2_{L1} &= \mathbb{V}ar(X)\mathbb{V}ar(U) \\ &+ \mathbb{E}^2(U)\mathbb{V}ar(X) + \mathbb{E}^2(X)\mathbb{V}ar(U) \\ &+ \alpha^2\mathbb{E}(W^4) - \mathbb{E}^2(W^2) \\ &+ 2\alpha\mathbb{E}(X)\alpha^2(\mathbb{E}(W^3) - \mathbb{E}(W)\mathbb{E}(W)^2) \end{aligned} \tag{A.7}$$

Bibliography

[1] I. Cox, M. Miller, J. Bloom, and M. Miller, *Digital Watermarking*. Morgan Kaufmann, 2001.

[2] W. Zhang, *Secure Data Aggregation in Wireless Sensor Networks*. University of Texas, Alington: Computer Science & Engineering, 2008.

[3] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia," *Image Processing, IEEE Transactions on*, vol. 6, no. 12, pp. 1673–1687, 1997.

[4] Y. Xin, *Robust Digital Watermarking in Images*. University of Manitoba, Canada: Engineering & Computer Engineering, 2006.

[5] Y. Xin, S. Liao, and M. Pawlak, "Circularly Orthogonal Moments for Geometrically Robust Image Watermarking," *Pattern Recognition*, vol. 40, no. 12, pp. 3740–3752, 2007.

[6] M. Barni and F. Bartolini, *Watermarking Systems Engineering: Enabling Digital Assets Security and Other Applications*. CRC Press, 2004.

[7] E. Y. Lam and J. W. Goodman, "A Mathematical Analysis of the DCT Coefficient Distributions for Images," *Image Processing, IEEE Transactions on*, vol. 9, no. 10, pp. 1661–1666, 2000.

[8] Q. Cheng and T. S. Huang, "An Additive Approach to Transform-Domain Information Hiding and Optimum Detection Structure," *Multimedia, IEEE Transactions on*, vol. 3, no. 3, pp. 273–284, 2001.

[9] "Asymptotically Optimal Detection for Additive Watermarking in the DCT and DWT Domains,"

[10] J. R. Hernandez, M. Amado, and F. Perez-Gonzalez, "Dct-Domain Watermarking Techniques for Still Images: Detector Performance Analysis and a New Structure," *Image Processing, IEEE Transactions on*, vol. 9, no. 1, pp. 55–68, 2000.

[11] T. P.-c. Chen and T. Chen, "A Framework for Optimal Blind Watermark Detection," in *Proceedings of the 2001 workshop on Multimedia and security: new challenges*, pp. 11–14, ACM, 2001.

[12] W. L. Martinez and A. R. Martinez, *Computational Statistics Handbook with MATLAB*. CRC press, 2001.

[13] M. Sugiyama, T. Suzuki, and T. Kanamori, "Density Ratio Estimation: A Comprehensive Review," *RIMS Kokyuroku*, pp. 10–31, 2010.

[14] M. Sugiyama, T. Suzuki, and T. Kanamori, *Density Ratio Estimation in Machine Learning*. Cambridge University Press, 2012.

[15] J. Qin, "Inferences for Case-Control and Semiparametric Two-Sample Density Ratio Models," *Biometrika*, vol. 85, no. 3, pp. 619–630, 1998.

[16] J. L. M. Jerry D. Gibson, *Introduction to Nonparametric Detection with Applications*. Academic Press Inc, 1976.

[17] W. Greblicki and M. Pawlak, *Nonparametric System Identification*. Cambridge University Press Cambridge, 2008.

[18] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. Wiley. com, 2004.

[19] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume 2: Detection Theory*. Prentice Hall PTR, 1998.

[20] B. C. Levy, *Principles of Signal Detection and Parameter Estimation*. Springer, 2008.

[21] E. P. Simoncelli and B. A. Olshausen, "Natural Image Statistics and Neural Representation," *Annual review of neuroscience*, vol. 24, no. 1, pp. 1193–1216, 2001.

[22] A. Srivastava, A. B. Lee, E. P. Simoncelli, and S.-C. Zhu, "On Advances in Statistical Modeling of Natural Images," *Journal of mathematical imaging and vision*, vol. 18, no. 1, pp. 17–33, 2003.

[23] J. Lv, *Machine Learning Techniques for Large-Scale System Modeling*. University of Manitoba, Canada: Engineering & Computer Engineering, 2011.

[24] R. I. Jennrich, "Asymptotic Properties of Non-Linear Least Squares Estimators," *The Annals of Mathematical Statistics*, vol. 40, no. 2, pp. 633–643, 1969.

[25] H. White, "Consequences and Detection of Misspecified Nonlinear Regression Models," *Journal of the American Statistical Association*, vol. 76, no. 374, pp. 419–433, 1981.

[26] W. Hardle, *Nonparametric and Semiparametric Models*. Springer Verlag, 2004.

[27] P. J. Rousseeuw, "Silhouettes: A Graphical Aid to the Interpretation and Validation of Cluster Analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[28] T. A.-W. Hammel, *Semiparametric Estimation for Finite Mixture Models Using an Exponential Tilt*. PhD thesis, The Pennsylvania State University, 2010.

[29] A. Gilat and V. Subreamaniam, *Numerical Methods for Engineers and Scientists: An Introduction with Applications Using MATLAB*. Wiley Hoboken, NJ, 2007.