Reduced Precision Bayesian Network Classifiers

Author: Carlos Eduardo Cancino Chacón, 1131710
Date: Graz, May 30, 2013
Rev.: Beta 1.0

ABSTRACT

Probabilistic graphical models represent an important approach to machine learning problems since they merge graphical models and probabilistic inference. Bayesian network classifiers are probabilistic classifiers which achieve good classification rates in several applications. They consist of a directed acyclic graph and a set of conditional probability distributions. In discrete valued domains these classifiers can be represented by conditional probability tables. In this paper we study the effects of quantizing the entries of the conditional probability tables by deriving classification performance bounds. A deterministic analysis, based on interval arithmetic, and a statistical analysis, based on statistical decision theory, are performed. Our results suggest that only small bit-widths are required to obtain a good classification performance.

Contents

1 Introduction

Probabilistic Graphical Models (PGMs) are the method of choice for modeling uncertainty in several areas including computer vision, speech processing, cognitive science and artificial intelligence [1]. They represent an important approach to machine learning problems since they merge graphical models and probabilistic inference, and because qualitative patterns of commonsense reasoning *are naturally embedded within the syntax of probability calculus* [2]. PGMs provide techniques to deal with uncertainty and complex problems. Examples of statistical models that can be represented as PGMs are hidden Markov models, Kalman filters and Boltzmann machines.

1.1 Motivation

In discrete valued domains, Bayesian Network Classifiers (BNCs) can be represented by compact models. These models often require less parameters than Support Vector Machines (SVMs), while achieving comparable classification performance [3]. For BNCs, classification resorts to the computation of a sum of log probabilities followed by the evaluation of a maximum operator. In contrast, for SVMs using Gaussian kernels, Euclidean distances must to be calculated and exponential functions evaluated. This suggest that BNCs are suitable for implementation on low-complexity platforms, like embedded systems.

1.2 Objectives

The objective of this project was to study the effects of quantizing the entries of the conditional probability tables of BNCs by deriving classification performance bounds. A deterministic analysis, based on interval arithmetic, and a statistical analysis were performed.

This report is organized as follows: In the rest of Chapter 1, the concepts of BNs, probabilistic classification and BNCs are introduced, alongside with the notation. In Chapter 2, performance bounds for BNCs are derived. These bounds are illustrated in experiments in Chapter 3. Finally, in Chapter 4 conclusions and future work are provided.

1.3 Bayesian Networks (BNs)

Definition *Bayesian Network*

A Bayesian Network $\mathcal{B} = (\mathcal{G}, \mathcal{P}_{\mathcal{G}})$ consists of the following [4]:

1. A set of random variables $\mathbf{X} = \{X_0, \ldots, X_L\}$ (also referred to as nodes) and a set of directed edges $\mathbf{E}$ between the variables. The variables together with the directed edges form a directed acyclic graph (DAG) $\mathcal{G} = (\mathbf{X}, \mathbf{E})$.
2. To each variable X_i with parents $\mathbf{Pa}(X_i)$ a Conditional Probability Distribution (CPD) $P_{X_i}(X_i|\mathbf{Pa}(X_i))$ is attached, so $\mathcal{P}_{\mathcal{G}} = \{P_{X_0}(X_0|\mathbf{Pa}(X_0)), \ldots, P_{X_L}(X_L|\mathbf{Pa}(X_L))\}$ will be the set of local CPDs.

A BN over $\mathbf{X}$ specifies a uniquely defined joint probability distribution $\mathrm{P}^{\mathcal{B}}(\mathbf{X})$, given as

$$\mathrm{P}^{\mathcal{B}}(X_0 \ldots, X_L) = \prod_{i=0}^{L} P_{X_i}(X_i|\mathbf{Pa}(X_i)). \tag{1.1}$$

If each X_i can assume only one of a finite set of mutually exclusive states, the CPDs can be specified by Conditional Probability Tables (CPTs) (This is for example the case if all nodes assume discrete values only).

For the rest of this report, if not stated otherwise, the conditional probability $P(X_i = j|\mathbf{Pa}(X_i) = \mathbf{h})$ will be abbreviated as $\theta^i_{j|\mathbf{h}}$, and its respective logarithmic probability as $w^i_{j|\mathbf{h}} = \log(\theta^i_{j|\mathbf{h}})$.

An important instance of BNs assume *naive Bayes* structures. In particular for such a structure, Equation (1.1) becomes

$$\mathrm{P}^{\mathcal{B}}(X_0 \ldots, X_L) = P(X_0) \prod_{i=1}^{L} P(X_i|X_0). \tag{1.2}$$

The corresponding structure $\mathcal{G}$ is shown in Figure 1.1.

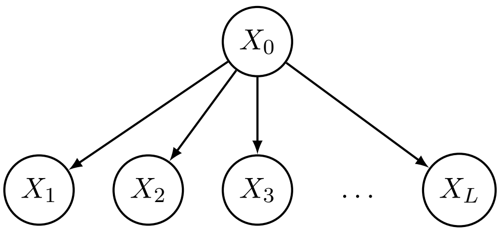


Figure 1.1: Naive Bayes Network

1.4 Probabilistic Classification

Let $\mathbf{X} = \{X_1, \ldots, X_L\}$ be a set of random variables called *features* or *attributes* and C a *class variable*. These RVs are modeled by the joint probability distribution $P^*(C, \mathbf{X})$.

Definition *Classifier*

A classifier h is a function

$$\begin{aligned} h\colon \mathrm{sp}(\mathbf{X}) &\rightarrow \mathrm{sp}(\mathrm{C}) \\ \mathbf{x} &\mapsto h(\mathbf{x}), \end{aligned} \tag{1.3}$$

where $\mathrm{sp}(\mathbf{X})$ is the set of all possible assignments of $\mathbf{X} = \{X_1, \ldots, X_L\}$ to $\mathrm{sp}(C)$, i.e. the set of all *classes* C[1].

Any probability distribution $P(C, \mathbf{X})$ induces a classifier $h_{P(C,\mathbf{X})}$ according to

$$\begin{aligned} h_{P(C,\mathbf{X})}\colon \mathrm{sp}(\mathbf{X}) &\rightarrow \mathrm{sp}(\mathrm{C}) \\ \mathbf{x} &\mapsto \arg\max_{c \in C} P(C = c | \mathbf{X} = \mathbf{x}). \end{aligned} \tag{1.4}$$

That is, each instantiation $\mathbf{x}$ of $\mathbf{X}$ is classified as the maximum a-posteriori (MAP) estimate of C given $\mathbf{x}$ under $P(C, \mathbf{X})$.

Definition *Classification Error*

The *Classification Error* of classifier h is

$$\mathrm{Err}(h) = \mathbb{E}_{P^*(C,\mathbf{X})}[\mathbf{1}\{C \neq h(\mathbf{X})\}], \tag{1.5}$$

where $\mathbf{1}\{A\}$ is the indicator function defined as

$$\mathbf{1}\{A\} = \begin{cases} 1 & \text{if } A \text{ is true,} \\ 0 & \text{otherwise.} \end{cases} \tag{1.6}$$

Usually, the generalization error can not be evaluated but is estimated using cross-validation.

1.5 Learning Bayesian Network Classifiers

Since BNs represent a joint probability distribution, they naturally induce classifiers. From the definitions of BNs and probabilistic classifiers, without loss of generality we can identify X_0 as the class variable C. Here it is assumed that C has no parents in $\mathcal{G}$, i.e. $\mathbf{Pa}(C) = \emptyset$. BNs $\mathcal{B} = (\mathcal{G}, \mathcal{P}_\mathcal{G})$ for classification can be optimized in two ways: selecting the graph structure $\mathcal{G}$ (*structure learning*), or by learning the conditional probabilities $\mathcal{P}_\mathcal{G}$ (*parameter learning*). In this paper, only classifiers with naive Bayes structure will be considered, i.e. each feature has the class as its only parent [4]. For parameter learning two paradigms exist, namely *generative* and *discriminative parameter learning.* We review both paradigms shortly in the following.

[1] Since typically $P^*(C, \mathbf{X})$ is unknown, the classifier h can be constructed using a *training set* $\mathcal{D} = \{(c^{(n)}, \mathbf{x}^{(n)} | n = 1, \ldots, N\}$ with N samples drawn i.i.d. from $P^*(C, \mathbf{X})$, where $c^{(n)}$ and $\mathbf{x}^{(n)}$ denotes the instantiation of C and $\mathbf{X}$ in the n^{th} sample.

Generative Parameter Learning

In *generative parameter learning*, the goal is the identification of the parameters representing the generative process that results in the data of the training set. An example of this approach is maximum likelihood (ML) learning. Its objective is the maximization of the likelihood of the training data given the parameters. This means, they are learned as

$$\mathcal{P}_{\mathcal{G}}^{ML} = \arg\max_{\mathcal{P}_{\mathcal{G}}} \prod_{n=1}^{N} \mathrm{P}^{\mathcal{B}}(c^{(n)}, \mathbf{x}^{(n)}), \tag{1.7}$$

where $\mathrm{P}^{\mathcal{B}}(c^{(n)}, \mathbf{x}^{(n)})$ is the joint distribution induced by the BN $\mathcal{B}(\mathcal{G}, \mathcal{P}_{\mathcal{G}})$.

Discriminative Parameter Learning

In *discriminative parameter learning*, the objective is to identify parameters that lead to a good classification performance on new samples drawn from $\mathrm{P}^*(C, \mathbf{X})$. Several approaches have been described in literature, like maximum conditional likelihood (MCL) and maximum margin (MM). Here, we present the MM objective, which compares the likelihood of the n^{th} sample belonging to the correct class $c^{(n)}$ to belonging to the strongest competing class. That is,

$$\mathcal{P}_{\mathcal{G}}^{MM} = \arg\max_{\mathcal{P}_{\mathcal{G}}} \prod_{n=1}^{N} \min\left(\gamma, d^{(n)}\right), \tag{1.8}$$

where $\min\left(\gamma, d^{(n)}\right)$ denotes the hinge loss and $d^{(n)}$ is the margin of the n^{th} sample given as

$$d^{(n)} = \frac{\mathrm{P}^{\mathcal{B}}(c^{(n)}, \mathbf{x}^{(n)})}{\max_{c \neq c^{(n)}} \mathrm{P}^{\mathcal{B}}(c, \mathbf{x}^{(n)})}$$

and $\gamma > 1$ is a parameter controlling the margin. The n^{th} sample is classified correctly if $d^{(n)} > 1$.

1.6 Quantization in Classification

A BNC classifies an unlabeled sample $\mathbf{x}$ as belonging to the class with highest posterior probability according to Equation (1.3). This classification can be equivalently performed in the logarithmic domain as

$$c^* = \arg\max_{c} \log \mathrm{P}^{\mathcal{B}}(c, \mathbf{x}). \tag{1.9}$$

Using Equation (1.2), i.e. the factorization properties of the BNC, this can be restated as

$$
\begin{aligned}
c^* &= \arg\max_c \left[\log \mathrm{P}(c) + \sum_{i=1}^{L} \log \mathrm{P}(\mathbf{x}_i|\mathbf{Pa}(\mathbf{x}_i))\right] \\
&= \arg\max_c \left[w_c + \sum_{i=1}^{L} w^i_{j|\mathbf{h}}\right],
\end{aligned}
\tag{1.10}
$$

where $\mathbf{Pa}(\mathbf{x}_i)$ denotes the instantiation of the parents of X_i according to $\mathbf{x}_i$. For the present analysis it is assumed that the quantization of the logarithmic probabilities, i.e. w_c and $w^i_{j|\mathbf{h}}$, is performed.

In this paper, the case of quantization using fixed-point numbers is considered. The quantization is given as

$$
\hat{w}^i_{j|\mathbf{h}} = Q^{b_i}_{b_f}(w^i_{j|\mathbf{h}}), \tag{1.11}
$$

where $b_i \in \mathbb{N}_0$ is the number of the integer bits, $b_f \in \mathbb{N}_0$ the number of fractional bits and $Q^{b_i}_{b_f}$ the quantization operator. This quantizer $Q^{b_i}_{b_f}(\cdot) : \mathbb{R}^- \to \mathbb{B}^{b_i}_{b_f}$ performs quantization by rounding, where $\mathbb{B}^{b_i}_{b_f} = \{-\sum_{k=-b_f}^{b_i} b_k 2^k : b_k \in \{0,1\}\}$ is the set of all negative fixed-point numbers with b_i integer bits and b_f fractional bits. It is important to note that the quantized log parameters are in general not properly normalized, which means that $\sum_j \exp(\hat{w}^i_{j|\mathbf{h}}) \neq 1$. Nevertheless, for the sake of simplicity of analysis we ignored this fact.

2 Classification Rate Performance Bounds

In this section, $\mathcal{B} = (\mathcal{G}, \mathcal{P}_{\mathcal{G}})$ represent a regular BNC with log parameters $w^i_{j|\mathbf{h}}$, and $\mathcal{B}_Q$ denote the quantized version of the BNC, i.e. with log parameters $\hat{w}^i_{j|\mathbf{h}}$.

2.1 Deterministic Case

In a deterministic scenario the objective is to bound the classification error of $\mathcal{B}_Q$ in terms of the classification error of $\mathcal{B}$. In order to do so, it can be seen that the largest possible error due to quantization of the parameters is $\Delta := 2^{-b_f-1}$ [5], i.e.

$$|Q^{b_i}_{b_f}(\alpha) - \alpha| \leq \Delta \tag{2.1}$$

for every $\alpha \in \mathbb{R}$[2].

2.1.1 Worst-Case Analysis

The joint log probability of a BNC with quantized parameters according to Equation (1.2) is given by

$$\log \mathrm{P}^{\mathcal{B}_Q}(c, \mathbf{x}) = \hat{w}_c + \sum_{i=1}^{L} \hat{w}^i_{j|\mathbf{h}}. \tag{2.2}$$

Using Equation (2.1), it is possible to find upper and lower bounds on this expression as

$$w_c - \Delta + \sum_{i=1}^{L}(w^i_{j|\mathbf{h}} - \Delta) \leq \hat{w}_c + \sum_{i=1}^{L} \hat{w}^i_{j|\mathbf{h}} \leq w_c + \Delta + \sum_{i=1}^{L}(w^i_{j|\mathbf{h}} + \Delta), \tag{2.3}$$

or equivalently as

$$\underbrace{w_c + \sum_{i=1}^{L} w^i_{j|\mathbf{h}}}_{\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x})} -(L+1)\Delta \leq \hat{w}_c + \sum_{i=1}^{L} \hat{w}^i_{j|\mathbf{h}} \leq \underbrace{w_c + \sum_{i=1}^{L} w^i_{j|\mathbf{h}}}_{\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x})} +(L+1)\Delta. \tag{2.4}$$

[2] For simplicity of analysis the cases in which α is larger than the largest possible value representable by the chosen number format is not considered.

Hence,

$$\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) - (L+1)\Delta \leq \log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) \leq \log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) + (L+1)\Delta. \tag{2.5}$$

Using Equation (1.5) the classification error can be expressed as

$$\begin{aligned}\mathrm{Err}(h_{P^{\mathcal{B}_Q}(C,\mathbf{X})}) &= \mathbb{E}_{\mathrm{P}^*(C,\mathbf{X})}\left[\mathbf{1}\{C \neq h_{P^{\mathcal{B}_Q}(C,\mathbf{X})}(\mathbf{X})\}\right] \\ &= \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) > \max_{c'\neq c} \log \mathrm{P}^{\mathcal{B}_Q}(c',\mathbf{x})\}.\end{aligned} \tag{2.6}$$

Using Inequality (2.5) it is possible to upper bound the classification error as

$$\begin{aligned}&\sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) > \max_{c'\neq c} \log \mathrm{P}^{\mathcal{B}_Q}(c',\mathbf{x})\} \\ &\leq \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) > \max_{c'\neq c} \left(\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + 2(L+1)\Delta\right)\}.\end{aligned} \tag{2.7}$$

This results in the *worst case bound*,

$$\mathrm{Err}(h_{P^{\mathcal{B}_Q}(C,\mathbf{X})}) \leq \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) > \max_{c'\neq c} \left(\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + 2(L+1)\Delta\right)\}. \tag{2.8}$$

2.1.2 Best-Case Analysis

In a similar way, a *best case bound* can be determined. Again, using Inequality (2.5), a lower bound for the generalization error can be derived as

$$\mathrm{Err}(h_{P^{\mathcal{B}_Q}(C,\mathbf{X})}) \geq \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) > \max_{c'\neq c} \left(\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + 2(L+1)\Delta\right)\}. \tag{2.9}$$

The meaning of this is that no classifier must perform better than this bound after parameter quantization.

As shown in the previous analysis, it is interesting to note that the evaluation of this analytic bounds does not actually require the quantization of the parameters, as they are given in terms of the non quantized probability $\mathrm{P}^{\mathcal{B}}(C,\mathrm{X})$.

2.2 Stochastic Case

The objective of a stochastic analysis is to derive tighter bounds on the classification error holding with a user specified probability. In order to do so, the quantized log probability is modeled as

$$\hat{w}^i_{j|\mathbf{h}} = w^i_{j|\mathbf{h}} + e^i_{j|\mathbf{h}}, \tag{2.10}$$

where the quantization error $e^i_{j|\mathbf{h}}$ is a random variable. For simplicity of analysis, we assume that the estimate of the parameters of $\mathcal{B}$ is uniformly distributed in the quantization interval[3] $[-\Delta, \Delta]$. The probability density function of the quantization error is

$$f_{e^i_{j|\mathbf{h}}}(x) = \begin{cases} \frac{1}{2\Delta} & -\Delta \leq x \leq \Delta, \\ 0 & \text{otherwise.} \end{cases} \tag{2.11}$$

The mean value is

$$\mu_{e^i_{j|\mathbf{h}}} = \int_{-\infty}^{\infty} x f_{e^i_{j|\mathbf{h}}}(x)dx = \frac{1}{2\Delta}\int_{-\Delta}^{\Delta} xdx = 0, \tag{2.12}$$

and its variance

$$\sigma^2_{e^i_{j|\mathbf{h}}} = \int_{-\infty}^{\infty} x^2 f_{e^i_{j|\mathbf{h}}}(x)dx - \mu^2_{e^i_{j|\mathbf{h}}} = \frac{1}{2\Delta}\int_{-\Delta}^{\Delta} x^2 dx = \frac{\Delta^2}{3} = \frac{2^{-2b_f}}{12}. \tag{2.13}$$

Using Equation (2.10), the joint log probability $\log \mathrm{P}^{\mathcal{B}_Q}(c, \mathbf{x})$ can be written as

$$\begin{aligned} \log \mathrm{P}^{\mathcal{B}_Q}(c, \mathbf{x}) &= \hat{w}_c + \sum_{i=1}^{L} \hat{w}^i_{j|\mathbf{h}} \qquad (2.14) \\ &= w_c + e_c + \sum_{i=1}^{L}(w^i_{j|\mathbf{h}} + e^i_{j|\mathbf{h}}) \\ &= \log \mathrm{P}^{\mathcal{B}}(c, \mathbf{x}) + E, \end{aligned}$$

where $E := e_c + \sum_{i=1}^{L} e^i_{j|\mathbf{h}}$ is the sum of $(L+1)$ uniformly distributed RVs. Hence E is distributed according to a shifted Irwin Hall distribution [6]. Using methods derived from statistical decision theory it is possible to provide an estimate of E.

Under the assumption previously made, i.e. that the probability distribution of E is known, computing the classification error is equivalent to *hypothesis testing* using the *Minimum Probability of Error* criterion. This resembles a special case of the more general *Bayes Risk* detector which is particularly suited for hypothesis testing in pattern recognition systems [7]. This particular setup consist of two different hypothesis and a decision threshold.

[3] While this assumption is common in quantization analysis, it does not necessarily hold for real BNs. For example, if the parameters are calculated using ML, the computed parameters will be distributed around the optimal ML parameters. However, if plenty data for training is available the parameter estimates will not span the quantization interval.

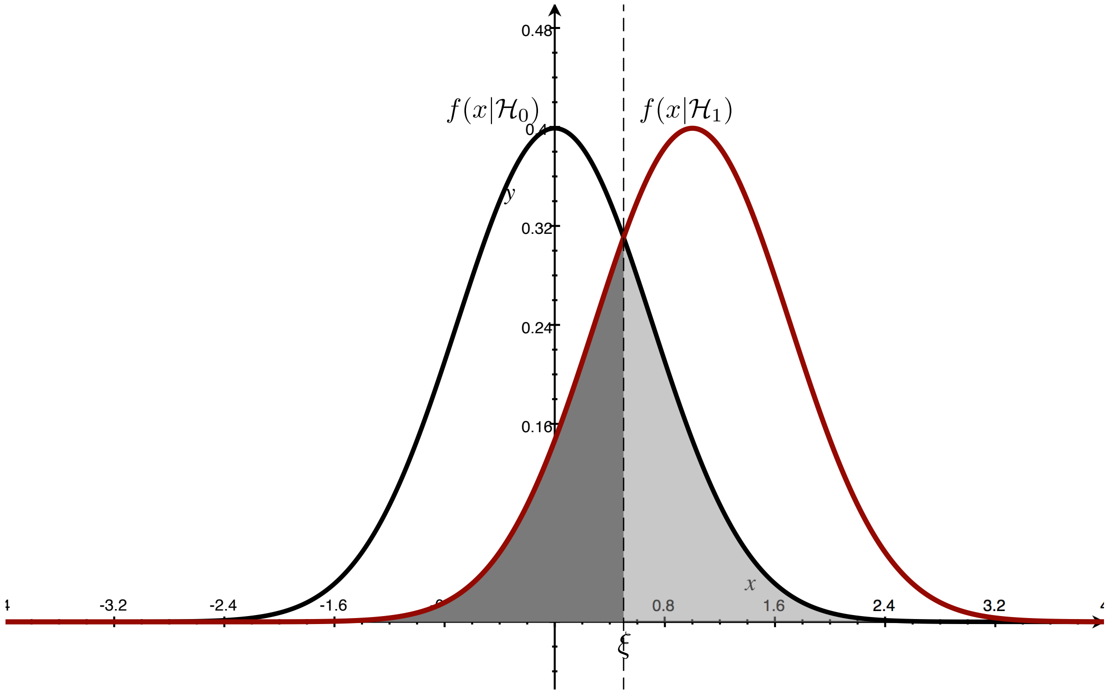


Figure 2.1: Hypothesis testing errors and their probabilities

The hypothesis $\mathcal{H}_0$ (null hypothesis) and $\mathcal{H}_1$ (alternative hypothesis) correspond to correct classification using the non quantized and the quantized parameters respectively. With these hypothesis there are four decision cases, namely

1. decide for $\mathcal{H}_0$ when $\mathcal{H}_0$ is true, with probability $\Pr\{\text{decide for } \mathcal{H}_0, \mathcal{H}_0 \text{ is true}\}$,
2. decide for $\mathcal{H}_0$ when $\mathcal{H}_1$ is true, with probability $\Pr\{\text{decide for } \mathcal{H}_0, \mathcal{H}_1 \text{ is true}\}$,
3. decide for $\mathcal{H}_1$ when $\mathcal{H}_0$ is true, with probability $\Pr\{\text{decide for } \mathcal{H}_1, \mathcal{H}_0 \text{ is true}\}$ and
4. decide for $\mathcal{H}_1$ when $\mathcal{H}_1$ is true, with probability $\Pr\{\text{decide for } \mathcal{H}_1, \mathcal{H}_1 \text{ is true}\}$.

Under the assumption that all the decision cases are disjoint, the whole probability of this decision setup is given as the sum of the probabilities of all four cases, i.e.

$$\begin{aligned} P_{decision} =& \Pr\{\text{decide for } \mathcal{H}_0, \mathcal{H}_0 \text{ is true}\} + \Pr\{\text{decide for } \mathcal{H}_0, \mathcal{H}_1 \text{ is true}\} \\ &+ \Pr\{\text{decide for } \mathcal{H}_1, \mathcal{H}_0 \text{ is true}\} + \Pr\{\text{decide for } \mathcal{H}_1, \mathcal{H}_1 \text{ is true}\} \\ =& 1. \end{aligned} \tag{2.15}$$

This can be rewritten as

$$P_{decision} = P_{correct} + P_{error} = 1, \tag{2.16}$$

where $P_{correct}$ is the probability of correct decision and P_{error} is the probability of error. From Equation (2.15), P_{error} is recognized as the sum of the cases where the an incorrect decision was made, i.e.

$$P_{error} = \Pr\{\text{decide for } \mathcal{H}_0, \mathcal{H}_1 \text{ is true}\} + \Pr\{\text{decide for } \mathcal{H}_1, \mathcal{H}_0 \text{ is true}\}. \tag{2.17}$$

Using Bayes theorem and the chain rule, the above equation can be rewritten as

$$P_{error} = P(\mathcal{H}_0|\mathcal{H}_1)P(\mathcal{H}_1) + P(\mathcal{H}_1|\mathcal{H}_0)P(\mathcal{H}_0), \tag{2.18}$$

where $P(\mathcal{H}_i|\mathcal{H}_j)$ is the conditional probability of deciding $\mathcal{H}_i$ given $\mathcal{H}_j$ is true and $P(\mathcal{H}_i)$ is the prior probability of $\mathcal{H}_i$.

In order to minimize the probability of error, we should decide for $\mathcal{H}_1$ if

$$\frac{f(x|\mathcal{H}_1)}{f(x|\mathcal{H}_0)} > \frac{P(\mathcal{H}_0)}{P(\mathcal{H}_1)}, \tag{2.19}$$

where $f(x|\mathcal{H}_1)$ are the pdfs of $P(\mathcal{H}_0|\mathcal{H}_0)$ and $P(\mathcal{H}_1|\mathcal{H}_1)$ respectively. Assuming that the prior probabilities are equal, i.e. $P(\mathcal{H}_0) = P(\mathcal{H}_1) = \frac{1}{2}$, we decide for $\mathcal{H}_1$ if

$$f(x|\mathcal{H}_1) > f(x|\mathcal{H}_0). \tag{2.20}$$

This decision setup represents a *maximum conditional likelihood* detector [7]. The decision threshold ξ is defined by

$$f(\xi|\mathcal{H}_1) = f(\xi|\mathcal{H}_0), \tag{2.21}$$

As illustrated in Figure 2.1, by plugging (2.19) and (2.21) into Equation (2.18), the minimum probability of error is

$$\begin{aligned} P_{me} &= \frac{1}{2}[P(\mathcal{H}_0|\mathcal{H}_1) + P(\mathcal{H}_1|\mathcal{H}_0)] \\ &= \frac{1}{2}\left[\int_{\xi}^{\infty} f(x|\mathcal{H}_0)dx + \int_{-\infty}^{\xi} f(x|\mathcal{H}_1)dx\right] \\ &= \frac{1}{2}\left[1 - F_{\mathcal{H}_0}(\xi) + F_{\mathcal{H}_1}(\xi)\right], \end{aligned} \tag{2.22}$$

where $F_{\mathcal{H}_i}(x)$ is the CDF of $f(\xi|\mathcal{H}_i)$.

At this point, it is important to introduce the concepts of sample error and true error regarding hypothesis testing [8]. *True error* of a hypothesis $\mathcal{H}_i$ is the probability of misclassifying according to the *real* distribution, whereas the *sample error* is an estimate of the true error computed from a test sample S drawn from the true distribution. The calculated error probability corresponds to the sample error, and statistical confidence bounds on the error can be calculated. A p

confidence interval is an interval that is expected with probability $p\%$ to contain the true error. Using the binomial distribution, it is possible to have an estimate of the true error $\hat{E}$ and its confidence interval as

$$\hat{E} = P_{me} \pm z_p \sqrt{\frac{P_{me}(1 - P_{me})}{n}}, \tag{2.23}$$

where n is the number of independent samples S and z_p the width of the smallest interval around the mean of the pdf of P_{me} that includes $p\%$ of the total probability. Here we can recognize the *confidence bound* B_p as

$$B_p = z_p \sqrt{\frac{P_{me}(1 - P_{me})}{n}}, \tag{2.24}$$

Using this confidence interval it is possible to write the probabilistic bounds in an analog form to the previous section as

$$\mathrm{Err}(h_{P^{\mathcal{B}_Q}}) \leq \sum_{c,\mathbf{x}} \mathrm{P}^*(c, \mathbf{x}) \mathbf{1}\{\log \mathrm{P}^{\mathcal{B}}(c, \mathbf{x}) > \max_{c' \neq c}(\log \mathrm{P}^{\mathcal{B}}(c', \mathbf{x}) + 2B_p)\}. \tag{2.25}$$

2.2.1 Irwin Hall Distribution

In the following, the pdf and CDF of a shifted Irwin Hall distribution are computed.

Probability density function

The RV X is defined as a sum of $(L + 1)$ uniformly distributed random variables U_k, taking values in $[a_k, b_k]$, i.e.

$$X = \sum_{k=0}^{L} U_k. \tag{2.26}$$

The probability density function of X is

$$f_X = f_{U_0} * f_{U_1} * \cdots * f_{U_L}, \tag{2.27}$$

where f_{U_i} is the pdf of U_i and $*$ denotes convolution. Using the Fourier transform $\mathcal{F}\{\cdot\}$ it is possible to transform the above equation

$$\mathcal{F}\{f_X\} = \prod_{k=0}^{L} \mathcal{F}\{f_{U_k}\}. \tag{2.28}$$

The pdf of the k^{th} RV U_k is

$$f_{U_k}(x) = \begin{cases} \frac{1}{b_k - a_k} & \text{if } x \in [a_k, b_k], \\ 0 & \text{otherwise.} \end{cases} \tag{2.29}$$

Calculating its Fourier Transform results in

$$\begin{aligned} F_{U_k}(p) = \mathcal{F}\{f_{U_k}(x)\} &= \int_{-\infty}^{\infty} f_{U_k}(x) \exp(-ipx) dx \\ &= \exp\left(-\frac{i(a_k + b_k)p}{2}\right) \operatorname{sinc}\left(\frac{p(b_k - a_k)}{2}\right). \end{aligned} \tag{2.30}$$

Substituting this result in Equation (2.28) yields

$$\begin{aligned} \mathcal{F}\{f_X(x)\} &= \prod_{k=0}^{L} \exp\left(-\frac{i(a_k + b_k)p}{2}\right) \operatorname{sinc}\left(\frac{p(b_k - a_k)}{2}\right) \\ &= \left(\prod_{k=0}^{L} \exp\left(-\frac{i(a_k + b_k)p}{2}\right)\right)\left(\prod_{k=0}^{L} \operatorname{sinc}\left(\frac{p(b_k - a_k)}{2}\right)\right) \\ &= \exp\left(-ip\sum_{k=0}^{L} \frac{(a_k + b_k)}{2}\right) \prod_{k=0}^{L} \operatorname{sinc}\left(\frac{p(b_k - a_k)}{2}\right). \end{aligned} \tag{2.31}$$

Then, in order to obtain the pdf again it is necessary to take the inverse Fourier transform, but this will result in an unholy integral, too difficult to calculate for a someone who isn't a demigod![4]

$$\begin{aligned} f_X(x) &= \mathcal{F}^{-1}\left\{\exp\left(-ip\sum_{k=0}^{L} \frac{(a_k + b_k)}{2}\right) \prod_{k=0}^{L} \operatorname{sinc}\left(\frac{p(b_k - a_k)}{2}\right)\right\} \\ &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \prod_{k=0}^{L} \operatorname{sinc}\left(\frac{p(b_k - a_k)}{2}\right) \exp\left(ip\left(x - \sum_{k=0}^{L} \frac{(a_k + b_k)}{2}\right)\right) dp. \end{aligned} \tag{2.32}$$

Another way of performing this integration is to consider the regular Irwin-Hall distribution. By substituting, $a_k = \mu_k - \Delta$ and $b_k = \mu_k + \Delta$, where $\mu_k = \frac{1}{2}$ and $\Delta = \frac{1}{2}$ for every k, the pdf becomes

[4] It is actually possible to compute this, assuming you have a lot of time, energy and patience, using a recursive integration by parts algorithm

$$\int uv = \sum_{i=0}^{n} (-1)^i u^{(i)} v_{i+1},$$

where $u^{(i)}$ denotes the i^{th} derivative with respect to the integration variable, and $u_i = \int \cdots \int v(dx)^{i+1}$.

$$
\begin{aligned}
f_{X_0}(x) &= \mathcal{F}^{-1}\left\{\exp\left(-ip\sum_{k=0}^{L}\frac{(a_k+b_k)}{2}\right)\prod_{k=0}^{L}\operatorname{sinc}\left(\frac{p(b_k-a_k)}{2}\right)\right\} \\
&= \mathcal{F}^{-1}\left\{\exp\left(-ip\Delta(L+1)\right)\operatorname{sinc}^{L+1}\left(\frac{p}{2}\right)\right\}.
\end{aligned} \tag{2.33}
$$

This is the pdf of the regular Irwin-Hall distribution [9], which is given by

$$
f_{X_0}(x) = \frac{1}{2L!}\sum_{k=0}^{L+1}(-1)^k\binom{L+1}{k}(x-k)^L\operatorname{sgn}(x-k). \tag{2.34}
$$

Subsequently, assuming that every random variable is defined in the interval $[\mu_k-\Delta, \mu_k+\Delta]$ it is possible to rewrite (2.31) as

$$
\begin{aligned}
f_X(x) &= \mathcal{F}^{-1}\left\{\exp\left(-ip\sum_{k=0}^{L}\mu_k\right)\prod_{k=0}^{L}\operatorname{sinc}(p\Delta)\right\} \qquad (2.35)\\
&= \mathcal{F}^{-1}\left\{\exp\left(-ip\sum_{k=0}^{L}\mu_k\right)\operatorname{sinc}^{L+1}\left(\frac{\frac{p}{2}}{\frac{1}{2\Delta}}\right)\right\} \\
&= \mathcal{F}^{-1}\left\{\underbrace{\exp\left(ip\Delta(L+1)-ip\Delta(L+1)\right)}_{=1}\exp\left(-ip\sum_{k=0}^{L}\mu_k\right)\operatorname{sinc}^{L+1}\left(\frac{\frac{p}{2}}{\frac{1}{2\Delta}}\right)\right\} \\
&= \mathcal{F}^{-1}\left\{\exp\left(ip\left(\Delta(L+1)-\sum_{k=0}^{L}\mu_k\right)\right)\exp\left(-ip\Delta(L+1)\right)\operatorname{sinc}^{L+1}\left(\frac{\frac{p}{2}}{\frac{1}{2\Delta}}\right)\right\} \\
&= \frac{1}{2\Delta}\mathcal{F}^{-1}\left\{\frac{1}{\frac{1}{2\Delta}}\exp\left(ip\left(\Delta(L+1)-\sum_{k=0}^{L}\mu_k\right)\right)\exp\left(-ip\Delta(L+1)\right)\operatorname{sinc}^{L+1}\left(\frac{\frac{p}{2}}{\frac{1}{2\Delta}}\right)\right\}.
\end{aligned}
$$

Using properties of the Fourier transform, it is possible to obtain the pdf of the shifted Irwin-Hall distribution as

$$
f_X(x) = \begin{cases} f_{ih}(x) & \sum_{k=0}^{L}\mu_k-\Delta(L+1)\le x\le\sum_{k=0}^{L}\mu_k+\Delta(L+1), \\ 0 & \text{otherwise}, \end{cases} \tag{2.36}
$$

where $f_{ih}(x)$ is

$$
\begin{aligned}
f_{ih}(x) =& \frac{1}{4\Delta L!}\sum_{k=0}^{L+1}(-1)^k\binom{L+1}{k}\left(\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k\right)^L \\
&\times\operatorname{sgn}\left(\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k\right).
\end{aligned} \tag{2.37}
$$

The function f_{ih} is a piecewise polynomial that is nonzero only in the interval $[\sum_{k=0}^{L}\mu_k - \Delta(L+1), \sum_{k=0}^{L}\mu_k + \Delta(L+1)]$.

Cumulative distribution function

The CDF of the shifted Irwin Hall Distribution is given by

$$F_X(x) = \begin{cases} 1 & x > \sum_{k=0}^{L}\mu_k + \Delta(L+1), \\ F_{ih}(x) & \sum_{k=0}^{L}\mu_k - \Delta(L+1) \leq x \leq \sum_{k=0}^{L}\mu_k + \Delta(L+1), \\ 0 & x < \sum_{k=0}^{L}\mu_k - \Delta(L+1), \end{cases} \tag{2.38}$$

where $F_{ih}(X)$ is

$$\begin{aligned} F_{ih}(x) &= \int_{\sum_{k=0}^{L}\mu_k - \Delta(L+1)}^{x} f_{ih}(\xi) d\xi \\ &= \frac{1}{4\Delta L!}\sum_{k=0}^{L+1}(-1)^k\binom{L+1}{k}\int_{\sum_{k=0}^{L}\mu_k - \Delta(L+1)}^{x}\left(\frac{\xi + \Delta(L+1) - \sum_{k=0}^{L}\mu_k}{2\Delta} - k\right)^L \\ &\operatorname{sgn}\left(\frac{\xi + \Delta(L+1) - \sum_{k=0}^{L}\mu_k}{2\Delta} - k\right) d\xi. \end{aligned} \tag{2.39}$$

Making a change of variable $\chi = \frac{\xi+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta} - k$ this can be simplified to

$$F_{ih}(x) = \frac{1}{2L!}\sum_{k=0}^{L+1}(-1)^k\binom{L+1}{k}\int_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k}\chi^L \operatorname{sgn}(\chi) d\chi. \tag{2.40}$$

Integrating by parts yields

$$\begin{aligned} \int_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k}\chi^L \operatorname{sgn}(\chi) d\chi = u(\chi)v(\chi)\Big|_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k} \\ - \int_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k} u(\chi)v'(\chi) d\chi, \end{aligned} \tag{2.41}$$

where $u(\chi)$ and $v(\chi)$ are

$$\begin{aligned} u(\chi) &= \frac{1}{L+1}\chi^{L+1} \\ v(\chi) &= \operatorname{sgn}(\chi) \end{aligned} \tag{2.42}$$

and

$$u'(\chi) = \chi^L \\ v'(\chi) = 2\delta(\chi), \tag{2.43}$$

respectively. Substituting in the above integral yields

$$\begin{aligned}
&\int_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k} \chi^L \operatorname{sgn}(\chi) d\chi = \qquad (2.44)\\
&= \frac{\chi^{L+1}}{L+1}\operatorname{sgn}(\chi)\Big|_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k} - \frac{2}{L+1}\int_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k} \chi^{L+1}\delta(\chi)d\chi \\
&= \frac{\chi^{L+1}}{L+1}\operatorname{sgn}(\chi)\Big|_{-k}^{\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k} - \frac{2}{L+1}\chi^{L+1}\Big|_{\chi=0} \\
&= \frac{1}{L+1}\left(\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k\right)^{L+1} \operatorname{sgn}\left(\frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta}-k\right) \\
&\quad - \frac{1}{L+1}(-k)^{L+1}\operatorname{sgn}(-k).
\end{aligned}$$

Finally substituting (2.44) in (2.39) yields

$$F_{ih}(x) = \frac{1}{2(L+1)!}\sum_{k=0}^{L+1}(-1)^k\binom{L+1}{k}\left(\chi^{L+1}\operatorname{sgn}(\chi) - (-k)^{L+1}\operatorname{sgn}(-k)\right), \tag{2.45}$$

where $\chi = \frac{x+\Delta(L+1)-\sum_{k=0}^{L}\mu_k}{2\Delta} - k$.

In order to compute the minimum probability of error using (2.22), the threshold ξ must first be calculated. From Equation (2.36), finding ξ is equivalent to find the roots of

$$\begin{aligned}
&\sum_{k=0}^{L+1}(-1)^k\binom{L+1}{k}\left[\left(\frac{\xi+\Delta(L+1)-\sum_{k=0}^{L}\mu_{0_k}}{2\Delta}-k\right)^L\right. \\
&\times \operatorname{sgn}\left(\frac{\xi+\Delta(L+1)-\sum_{k=0}^{L}\mu_{0_k}}{2\Delta}-k\right) - \left(\frac{\xi+\Delta(L+1)-\sum_{k=0}^{L}\mu_{1_k}}{2\Delta}-k\right)^L \\
&\left.\times \operatorname{sgn}\left(\frac{\xi+\Delta(L+1)-\sum_{k=0}^{L}\mu_{1_k}}{2\Delta}-k\right)\right] = 0, \qquad (2.46)
\end{aligned}$$

where μ_{i_k} is the mean value which corresponds to the hypothesis $\mathcal{H}_i$. Since this is a piecewise defined polynomial of order L, according to the Abel-Ruffini theorem, it is not possible to find a general algebraic solution for ξ for $L \geq 5$ [10]. Hence, numerical methods, like Newton-Raphson, are required to find the roots of (2.46).

In the appendix A, a MATLAB implementation of the pdf and the CDF of the Irwin Hall distribution is provided.

2.2.2 Truncated Normal (Gaussian) Distribution

By the *Central Limit Theorem*, for a large number of features L the Irwin Hall distribution can be accurately approximated by a truncated Gaussian distribution. This allows us to compute the classification bounds analytically.

The pdf of the sum of $L+1$ normally distributed RVs $U_k \sim \mathcal{N}(\mu_k, \sigma^2)$ is[5]

$$f_{X_g}(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}} \exp\left(-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right), \tag{2.47}$$

where the mean value is

$$\mu_X = \sum_{k=0}^{L} \mu_k, \tag{2.48}$$

and the variance

$$\sigma_X^2 = (L+1)\sigma^2. \tag{2.49}$$

A pdf of a truncated Gaussian distribution defined on the same interval as the pdf of the Irwin Hall distribution from Equation (2.36) is

$$f_X(x) = \begin{cases} f_{tg}(x) & \mu_X - \Delta(L+1) \leq x \leq \mu_X + \Delta(L+1), \\ 0 & \text{otherwise}, \end{cases} \tag{2.50}$$

where $f_{tg}(x)$ is

$$f_{tg}(x) = \frac{1}{\sqrt{2\pi\sigma_X^2}\operatorname{erf}\left(\frac{\Delta(L+1)}{\sqrt{2\sigma_X^2}}\right)} \exp\left(-\frac{(x-\mu_X)^2}{2\sigma_X^2}\right). \tag{2.51}$$

The function $f_{tg}(x) = Af_{X_g}(x)$ is a renormalized version of the pdf from Equation (2.47), where

[5] The proof of this pdf is found in the appendix C.

A is a normalization constant which satisfies

$$\int_{\mu_X-\Delta(L+1)}^{\mu_X+\Delta(L+1)} A f_{X_g}(x)dx = 1. \tag{2.52}$$

The CDF is given as

$$F_X(x) = \begin{cases} 0 & x < \mu_X - \Delta(L+1), \\ F_{tg}(x) & \mu_X - \Delta(L+1) \leq x \leq \mu_X + \Delta(L+1), \\ 1 & x \geq \mu_X + \Delta(L+1), \end{cases} \tag{2.53}$$

where $F_{tg}(x)$ is

$$\begin{aligned} F_{tg}(x) &= \int_{\mu_X-\Delta(L+1)}^{x} f_{tg}(t)dt \\ &= \frac{1}{\sqrt{2\pi\sigma_X^2}\operatorname{erf}\left(\frac{\Delta(L+1)}{\sqrt{2\sigma_X^2}}\right)} \int_{\mu_X-\Delta(L+1)}^{x} \exp\left(-\frac{(t-\mu_x)^2}{2\sigma_X^2}\right) dt \\ &= \frac{1}{2\operatorname{erf}\left(\frac{\Delta(L+1)}{\sqrt{2\sigma_X^2}}\right)} \left[\operatorname{erf}\left(\frac{x-\mu_X}{2\sigma_X^2}\right) + \operatorname{erf}\left(\frac{\Delta(L+1)}{2\sigma_X^2}\right)\right]. \end{aligned} \tag{2.54}$$

Bound Computation

From (2.21) and (2.50), the decision threshold ξ is found by solving

$$\exp\left(-\frac{(\xi-\mu_0)^2}{2\sigma_X^2}\right) = \exp\left(-\frac{(\xi-\mu_1)^2}{2\sigma_X^2}\right), \tag{2.55}$$

where $\mu_0 = \sum_{k=0}^{L} \mu_{k,0}$ and $\mu_1 = \sum_{k=0}^{L} \mu_{k,1}$ are the mean values of the pdfs describing $\mathcal{H}_0$ and $\mathcal{H}_1$ respectively. This results in

$$\xi = \frac{\mu_0 + \mu_1}{2}. \tag{2.56}$$

Plugging (2.53) and (2.56) in Equation (2.22), the minimum probability of error is

$$P_{me} = \frac{1}{2}\left[1 - \frac{\operatorname{erf}\left(\frac{\mu_1-\mu_0}{2\sqrt{2\sigma_X^2}}\right)}{\operatorname{erf}\left(\frac{\Delta(L+1)}{\sqrt{2\sigma_X^2}}\right)}\right]. \tag{2.57}$$

Substituting (2.13) and (2.49) in (2.57), the probability of error, given in terms of the number of fractional bits used for quantization is

$$P_{me}(b_f) = \frac{1}{2}\left[1 - \frac{\operatorname{erf}\left(\frac{\sqrt{3}(\mu_1-\mu_0)}{\sqrt{2^{1-2b_f}(L+1)}}\right)}{\operatorname{erf}\left(\sqrt{\frac{3(L+1)}{2}}\right)}\right]. \tag{2.58}$$

Using Equation (2.24), the bound of the interval with $p\%$ confidence is

$$B_p(b_f, p) = \frac{1}{2} z_p \sqrt{\frac{1 - \left(\frac{\operatorname{erf}\left(\frac{\sqrt{3}(\mu_1-\mu_0)}{\sqrt{2^{1-2b_f}(L+1)}}\right)}{\operatorname{erf}\left(\sqrt{\frac{3(L+1)}{2}}\right)}\right)^2}{n}}, \tag{2.59}$$

where z_p is given as[6]

$$\begin{aligned} z_p &= \sqrt{2}\operatorname{erf}^{-1}\left(\frac{p}{100}\operatorname{erf}\left(\frac{\Delta(L+1)}{\sqrt{2}}\right)\right) \\ &= \sqrt{2}\operatorname{erf}^{-1}\left(\frac{p}{100}\operatorname{erf}\left(\frac{2^{-b_f-1}(L+1)}{\sqrt{2}}\right)\right). \end{aligned} \tag{2.60}$$

Finally, a probabilistic bound can be computed substituting (2.59) in Equation (2.25) as

$$\begin{aligned} \operatorname{Err}(h_{P^{\mathcal{B}_Q}}) \leq & \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x}) \mathbf{1}\left\{\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) \right. \\ & \left. > \max_{c' \neq c} \left(\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + z_p \sqrt{\frac{1 - \left(\frac{\operatorname{erf}\left(\frac{\sqrt{3}(\mu_1-\mu_0)}{\sqrt{2^{1-2b_f}(L+1)}}\right)}{\operatorname{erf}\left(\sqrt{\frac{3(L+1)}{2}}\right)}\right)^2}{n}} \right) \right\}. \end{aligned} \tag{2.61}$$

[6] Full derivation of this factor in Appendix B.

3 Experiments

In this section, the results of classification experiments for MNIST and TIMIT data are shown, in order to evaluate the classification performance using BNCs with reduced precision parameters. The MNIST dataset deals with handwritten recognition, while the TIMIT corresponds to phonetic classification [3, 11, 12].

3.1 Experimental Setup

The experiments were performed as follows:

1. *Learning stage (Non quantized).* The parameters of the BNCs are initially learned using maximum likelihood and maximum margin paradigm, using the implementation described in [3]. Computations are performed using double precision floating point numbers. The structure of the classifiers is naive Bayes.

2. *Quantization of the parameters.* The determined log parameters are quantized using a fixed-point representation with varying precision. The precision is varied by incrementing the number of integer bits b_i up to 10 bits, and subsequently the number of fractional bits b_f up to 10 bits.

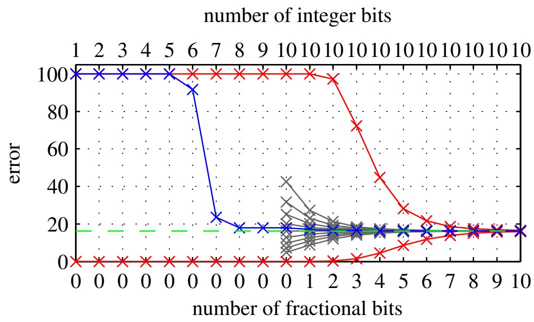
3. *Evaluation of the classification performance.* On a separate test the classification performance of the resulting BNC with reduced precision parameters is evaluated. All of the computations involved in the classification process are preformed using the same precision, i.e. the same number of quantization bits as the log parameters, and the sum of the log probabilities is computed with saturation arithmetic.

4. *Bound computation.* Using the results derived in Chapter 2, the error bounds are computed. The deterministic worst and best case bounds from Equations (2.8) and (2.9) are calculated first for $0 \leq b_i \leq 10$ integer bits and subsequently $1 \leq b_f \leq 10$ fractional bits. The probabilistic bounds using the truncated Gaussian distribution from Equation (2.61) are only computed for fractional bits, i.e. only for $b_i \geq 10$ integer bits.

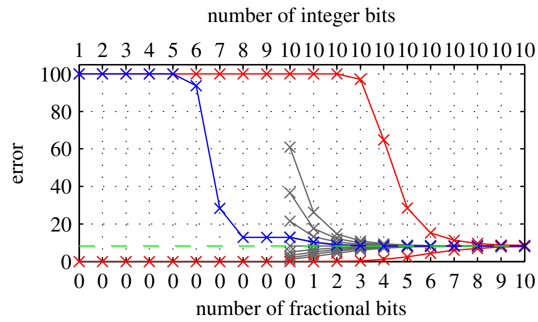
3.2 Results

In Figure 3.1, the results of classification error for MNIST and TIMIT data using BNCs are shown. The classification error in % is plotted as a function of the number of bits. The green dashed line represents the classification error with double precision parameters and the blue line represents the error with reduced precision parameters. The red lines represent the deterministic worst and best case bounds, whereas the grey lines represent the statistical bounds with 90%, . . . , 0% confidence.

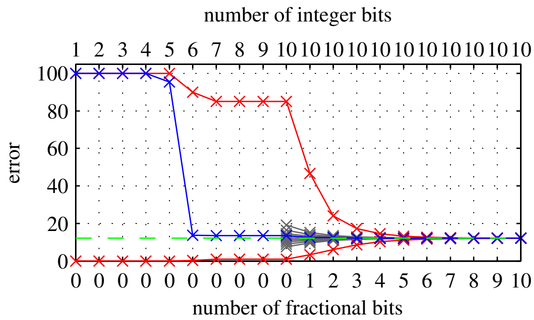
For MNIST data only 8 integer bits are necessary to achieve the classification performance of the BNC with non quantized parameters, while additional bits do not further increase performance. Similarly, for the TIMIT data only 7 integer bits are required. Using less than this amount



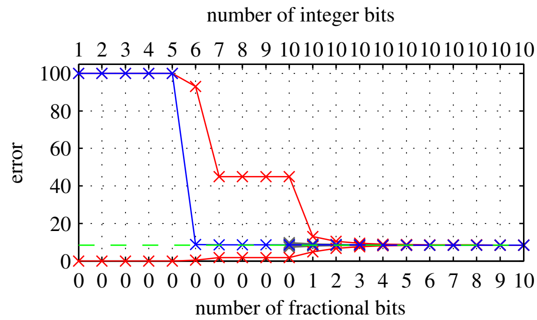
(a) MNIST data, ML parameters



(b) MNIST data, MM parameters



(c) TIMIT data, ML parameters



(d) TIMIT data, MM parameters

Figure 3.1: Classification error for MNIST and TIMIT data using BNCs with a NB structure and ML and MM parameters with deterministic and probabilistic bounds.

of bits, the classification error is large, mainly due to the non linear effects of the saturation arithmetic in the sum of the log probabilities.

Similar observations can be made from the bounds, and in particular it was noticed that they tighten with an increasing number of fractional bits. For the MNIST data they are almost tight for 3 or more fractional bits, while for the TIMIT data the bounds are almost tight with 2 or more fractional bits.

4 Conclusions and Future Work

In this paper, the quantization effects in Bayesian network classifiers with naive Bayes structure and reduced precision fixed point parameters were studied by computing bounds for the generalization error of the classifiers. A deterministic analysis resulted in worst case and best case bounds. A probabilistic analysis provided stochastic bounds assuming that the quantization error is uniformly distributed. Bounds using a shifted Irwin Hall distribution and a truncated Gaussian distribution were computed.

Future work includes deriving more accurate probabilistic bounds, considering a *dynamic range measure* and convergence criteria for loopy belief propagation proposed, as proposed in [13].

Bibliography

[1] F. Pernkopf, R. Peharz, and S. Tschiatschek, "Introduction to Probabilistic Graphical Models."

[2] J. Pearl, *Probabilistic Reasoning in Intelligent Systems.* Morgan Kaufmann Pub, 1988.

[3] F. Pernkopf, M. Wohlmayr, and S. Tschiatschek, "Maximum margin Bayesian network classifiers," *IEEE Transactions on Pattern Analysis and Machine Learning*, vol. 34, no. 3, pp. 521–532, 2012.

[4] T. D. Nielsen and F. V. JENSEN, *Bayesian Networks and Decision Graphs.* Springer, Jun. 2007.

[5] W. T. Padgett and D. V. Anderson, *Fixed-Point Signal Processing.* Morgan & Claypool Publishers, 2010.

[6] M. Batsyn and V. Kalyagin, "An Analytical Expression for the Distribution of the Sum of Random Variables with a Mixed Uniform Density and Mass Function."

[7] S. M. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory.* University of Rhode Island: Prentice Hall, Mar. 2009.

[8] T. M. Mitchell, *Machine Learning.* New York: McGraw Hill.

[9] F. Killmann and E. von Collani, "A note on the convolution of the uniform and related distributions and their use in quality control," *Economic Quality Control*, vol. 16, no. 1, pp. 17–41, 2001.

[10] R. L. Burden and J. D. Faires, *Análisis Numérico*, 7th ed. Youngstown State University: Thomson Learning, 2002.

[11] S. Tschiatschek, P. Reinprecht, M. Mücke, and F. Pernkopf, "Bayesian Network Classifiers with Reduced Precision Parameters," *Machine Learning and Knowledge Discovery in Databases*, pp. 74–89, 2012.

[12] R. Peharz and F. Pernkopf, "Exact Maximum Margin Structure Learning of Bayesian Networks," *arXiv*, vol. cs.LG, Jun. 2012.

[13] A. T. Ihler, J. W. Fisher, and A. S. Willsky, "Loopy belief propagation: Convergence and effects of message errors," *Journal of Machine Learning Research (JMLR)*, vol. 6, no. 1, p. 905, 2006.

[14] M. R. Spiegel and J. M. Liu, *Fórmulas y tablas de matemática aplicada.* McGraw-Hill Companies, 1999.

[15] H. P. Hsu, *Análisis de Fourier.* Department of Electrical Engineering Wayne State University, Michigan: Prentice Hall, 1998.

[16] J. E. Marsden and M. Hoffmann, *Análisis básico de variable compleja.* Editorial Trillas, 1996.

A MATLAB implementation of the Irwin Hall Distribution

The implementation of the Irwin Hall distribution as computed in Chapter 2 is provided.

A.1 Probability density function

```
function pdf = irwinhallpdf(x,mu,delta)
%IRWINHALLPDF  Probability density function of the Irwin Hall Distribution
%   irwinhallpdf(X,MU,DELTA) is the pdf of the Irwin Hall distribution
%   evaluated in X, with MU a vector of mean values of the L+1 uniformly
%   distributed RVs and DELTA the scalar length of the interval.

if nargin < 2
  % zero centered distribution
  mu = zeros(1,5);
end
if nargin < 3
  % Default interval length
  delta = 1;
end

%Variable initialization

Mu = sum(mu);
L = length(mu);
sumterm = zeros(1,L+2);

%Computation of the pdf
if x < Mu-delta*(L+1)
    pdf = 0;

elseif x <= Mu+delta*(L+1)
    fact = 1/(4*delta*factorial(L));

    for k = 0:L+1
        xi = (x+delta*(L+1)-Mu)/(2*delta)-k;
        sumterm(k+1) = (-1)^k*nchoosek(L+1,k)*(xi)^L*sign(xi);
    end

    pdf = fact*sum(sumterm);

elseif x > Mu+delta*(L+1)
    pdf = 0;
```

```
end
```

A.2 Cumulative distribution function

```
function cdf=irwinhallcdf(x,mu,delta)
%IRWINHALLCDF  Cumulative distribution function of the Irwin Hall Distribution
%   irwinhallcdf(X,MU,DELTA) is the CDF of the Irwin Hall distribution
%   evaluated in X, with MU a vector of mean values of the L+1 uniformly
%   distributed RVs and DELTA the scalar length of the interval.

if nargin < 2
  % zero centered distribution
  mu = zeros(1,5);
end
if nargin < 3
  % Default interval length
  delta = 1;
end

%Variable initialization

Mu = sum(mu);
L = length(mu);
sumterm = zeros(1,L+2);

if x < Mu-delta*(L+1)
    cdf = 0;

elseif x <= Mu+delta*(L+1)

    fact = 1/(2*factorial(L+1));
    for k = 0:L+1
        xi = (x+delta*(L+1)-Mu)/(2*delta)-k;
        sumterm(k+1) = (-1)^k*nchoosek(L+1,k)*((xi)^(L+1)*sign(xi)-(-k)^(L+1)*sign(-k));
    end
    cdf = fact*sum(sumterm);

elseif x > Mu+delta*(L+1)
    cdf = 1;

end
```

B Derivation of the confidence interval width z_p

As stated in Chapter 2, according to the Central Limit Theorem, for a large number of features $L+1$, the Irwin Hall distribution can be approximated accurately by a truncated normal distribution. The pdf of a truncated Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 1$ is

$$f(x) = \begin{cases} \frac{A}{\sqrt{2\pi}} \exp\left(-\frac{x^2}{2}\right) & a < x < b, \\ 0 & \text{otherwise,} \end{cases} \tag{B.1}$$

where A is a normalization constant, that can be computed as

$$A = \frac{2}{\operatorname{erf}\left(\frac{b}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right)}. \tag{B.2}$$

Using this pdf, the interval around the mean, that includes p % of the total pdf, z_p, is given as the solution of

$$\int_{-z_p}^{z_p} \frac{\sqrt{2}}{\sqrt{\pi}\left(\operatorname{erf}\left(\frac{b}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right)\right)} \exp\left(-\frac{x^2}{2}\right) dx = \frac{p}{100}, \tag{B.3}$$

for $z_p \in [a, b]$. Evaluating this integral yields

$$z_p = \sqrt{2}\operatorname{erf}^{-1}\left(\frac{p}{100} \frac{\operatorname{erf}\left(\frac{a}{\sqrt{2}}\right) - \operatorname{erf}\left(\frac{a}{\sqrt{2}}\right)}{2}\right), \tag{B.4}$$

where $\operatorname{erf}^{-1}(x)$ is the inverse error function. By evaluating the limits a and b of the truncated Gaussian pdf in the calculated limits of the Irwin Hall distribution, $-\Delta(L+1)$ and $\Delta(L+1)$ respectively, the above equation results in Equation (2.60).

C Derivation of the pdf of the sum of $L+1$ normally distributed RVs

Here it is proved by mathematical induction that the pdf of a sum of $L+1$ normally distributed variables, i.e. $U_k \sim \mathcal{N}(\mu_k, \sigma^2)$, is also normally distributed with mean value with mean value $\mu_X = \sum_{k=0}^{L} \mu_k$ and variance $\sigma_X^2 = (L+1)\sigma^2$.

Basis: For this case $L = 0$ is considered. This implies that $X = U_0$, and its pdf is simply

$$f_X(x) = f_{U_0}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_0)^2}{2\sigma^2}\right),$$

which is a Gaussian distribution with mean $\mu_X = \mu_0$ and variance $\sigma_X^2 = (0+1)\sigma^2 = \sigma^2$, and thus it has been shown that it holds for $L = 0$.

Inductive step: Supposing it holds for $L = N$ (inductive hypothesis), the pdf of $X = \sum_{k=0}^{N} U_k$ is

$$f_{X_N}(x) = \frac{1}{\sqrt{2\pi N\sigma^2}} \exp\left(-\frac{(x-\sum_{i=0}^{N}\mu_i)^2}{2N\sigma^2}\right).$$

Since $X_{L=N+1}$ is the sum of two RVs, namely $X_{L=N}$ and U_{N+1}, its pdf is given by

$$\begin{aligned} f_{X_{N+1}}(x) &= f_{X_N}(x) * f_{U_{N+1}}(x) \\ &= \int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi N\sigma^2}} \exp\left(-\frac{\left(x-y-\sum_{i=0}^{N}\mu_i\right)^2}{2N\sigma^2}\right) \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-\mu_{N+1})^2}{2\sigma^2}\right) dy \\ &= \frac{1}{\sqrt{2\pi(N+1)\sigma^2}} \exp\left(-\frac{(x-\sum_{i=0}^{N+1}\mu_i)^2}{2(N+1)\sigma^2}\right). \end{aligned}$$

This means that X_{N+1} is also normally distributed, which proves the inductive step.

Since both the basis and the inductive step have been proved, it has now been proved by mathematical induction that

$$f_{X_g}(x) = \frac{1}{\sqrt{2\pi(L+1)\sigma^2}} \exp\left(-\frac{(x-\sum_{k=0}^{L}\mu_k)^2}{2(L+1)\sigma^2}\right) \tag{C.1}$$

holds $\forall L \in \mathbb{N}$. Q.E.D.

The CDF in this case is given as

$$F_X(x) = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x - \sum_{k=0}^{L}\mu_k}{\sqrt{2(L+1)\sigma^2}}\right)\right]. \tag{C.2}$$