BOUNDS FOR BAYESIAN NETWORK CLASSIFIERS WITH REDUCED PRECISION PARAMETERS

S. Tschiatschek, C. E. Cancino Chacón, F. Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria

ABSTRACT

Bayesian network classifiers are probabilistic classifiers achieving good classification rates in various applications. These classifiers consist of a directed acyclic graph and a set of conditional probability densities, which in case of discrete-valued nodes can be represented by conditional probability tables. In this paper, we investigate the effect of quantizing these conditional probabilities. We derive worst-case and best-case bounds on the classification rate using interval arithmetic. Furthermore, we determine performance bounds that hold with a user specified confidence using quantization theory. Our results emphasize that only small bit-widths are necessary to achieve good classification rates.

Index Terms— Bayesian network classifiers, custom precision analysis, quantization effects, discriminative parameter learning

1. INTRODUCTION

Bayesian network classifiers (BNCs) are probabilistic classifiers, represented by a directed acyclic graph (DAG) and a set of conditional probability densities (CPDs). For determining these CPDs two paradigms exist, namely generative parameter learning and discriminative parameter learning. Generative parameter learning aims at identifying CPDs that model the data generation process. In contrast, discriminative parameter learning aims at identifying CPDs such that good classification rates are achieved. BNCs with discriminatively optimized CPDs achieve classification rates comparable to support vector machines (SVMs) in several applications [1]. While SVMs are purely discriminative models, BNCs often show good generative properties even when their parameters were trained discriminatively. Hence BNCs can naturally deal with scenarios such as handling missing features and semi-supervised learning. In contrast, SVMs require imputation techniques in this case.

In discrete-valued domains, the CPDs of BNCs can be specified by conditional probability tables (CPTs). This results in compact models, often requiring far less parameters than SVMs with comparable classification performance [1]. Assuming complete data, classification using BNCs requires evaluation of a product of conditional probabilities, or equivalently, a sum of log conditional probabilities, followed by the evaluation of a maximum operator. In contrast, in SVMs using Gaussian kernels, classification involves computing Euclidean distances and evaluating exponential functions. This suggests, that the classification process of BNCs is easier to implement in hardware than implementing the classification process of SVMs. Following this thread, we aim at investigating properties of BNCs relevant to implementation on low-complexity platforms, e.g. embedded systems.

This work was supported by the Austrian Science Fund (project number P22488-N23).

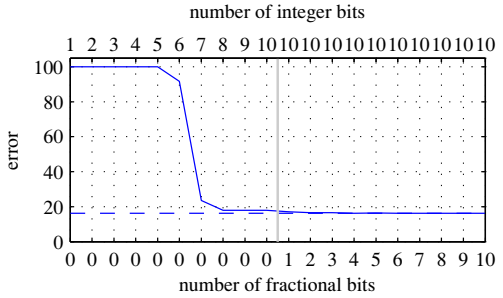


Fig. 1. Generalization error of a BNC for MNIST data using reduced precision parameters (solid line), and generalization error using double-precision floating point parameters (dashed line).

Various authors have already observed that the generalization error of BNCs is not sensitive with respect to the precision used for representing the entries of the conditional probability tables [2, 3]. For example, consider Fig. 1, which shows the generalization error of a BNC with naive Bayes (NB) structure and maximum likelihood parameters for MNIST data [4]. The number of bits used for representing the log parameters in a fixed-point format and performing the computations during classification is shown on the x-axis. Up to 10 bits are used for the integer part and up to 10 bits for the fractional part. When using only 9 bits in total, performance is close to optimal, i.e. the generalization error is as small as for double-precision floating point parameters.

In this paper, we study the effect of quantizing the entries of the CPTs of BNCs in more detail. Quantization is performed in the log domain using fixed-point numbers. This fixed-point representation is sufficient for accurate classification [5]. We are interested in worst-case estimates for the classification performance when quantizing the parameters. Additionally, we provide a statistical analysis of the generalization error observed after quantization.

This paper is organized as follows: In Section 2, we introduce our notation, Bayesian network classifiers and methods for parameter learning. In Section 3, we derive performance bounds for BNCs with reduced precision parameters and propose a method for estimating the expected classification rate. These results are illustrated by experiments in Section 4. We relate our work to prior results in Section 5. In Section 6, conclusions and an outlook are provided.

2. BACKGROUND

2.1. Probabilistic Classification

In probabilistic classification one assumes a random variable (RV) C denoting the class and RVs $X_1, \ldots, X_L$ representing the attributes of the classifier. These RVs are modeled by a joint probability distribution $\mathrm{P}^*(C, \mathbf{X})$, where $\mathbf{X} = [X_1, \ldots, X_L]$ is a random vector consisting of $X_1, \ldots, X_L$. Typically, $\mathrm{P}^*(C, \mathbf{X})$ is unknown. However, a training set $\mathcal{D}$ consisting of N samples drawn i.i.d. from $\mathrm{P}^*(C, \mathbf{X})$ is available, i.e. $\mathcal{D} = \{(c^{(n)}, \mathbf{x}^{(n)}) | n = 1, \ldots, N\}$, where $c^{(n)}$ denotes the instantiation of C and $\mathbf{x}^{(n)}$ the instantiation of $\mathbf{X}$ in the n^{th} training sample. The aim is to induce *good* classifiers provided a training set. Formally, a classifier $h\colon \mathrm{sp}(\mathbf{X}) \to \mathrm{sp}(C)$ is a mapping, where $\mathrm{sp}(\mathbf{X})$ denotes the set of all assignments of $\mathbf{X}$ and $\mathrm{sp}(C)$ is the set of classes. The generalization error of this classifier is

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

where $\mathbf{1}\{A\}$ denotes the indicator function and $\mathbb{E}_{\mathrm{P}^*(C,\mathbf{X})}[\cdot]$ is the expectation operator with respect to the distribution $\mathrm{P}^*(C, \mathbf{X})$. The indicator function $\mathbf{1}\{A\}$ equals one if statement A is true and zero otherwise. Typically, the generalization error can not be evaluated but is estimated using cross-validation [6].

Any probability distribution $\mathrm{P}(C, \mathbf{X})$ naturally induces a classifier $h_{\mathrm{P}(C,\mathbf{X})}$, given as

$$\begin{aligned} h_{\mathrm{P}(C,\mathbf{X})}\colon \quad & \mathrm{sp}(\mathbf{X}) \to \mathrm{sp}(C), \\ & \mathbf{x} \mapsto \arg\max_{c \in C} \mathrm{P}(C = c | \mathbf{X} = \mathbf{x}). \end{aligned} \tag{2}$$

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

2.2. Learning Bayesian Network Classifiers

Bayesian Networks (BNs) [7, 8] are used to represent joint probability distributions in a compact and intuitive way. A BN $\mathcal{B} = (\mathcal{G}, \mathcal{P}_\mathcal{G})$ consists of a directed acyclic graph $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V} = \{X_0, \ldots, X_L\}$ is the set of nodes and $\mathbf{E}$ the set of edges of the graph, and a set of local conditional probability distributions $\mathcal{P}_\mathcal{G} = \{\mathrm{P}(X_0|Pa(X_0)), \ldots, \mathrm{P}(X_L|Pa(X_L))\}$. The terms $Pa(X_0), \ldots, Pa(X_L)$ denote the set of parents of $X_0, \ldots, X_L$ in $\mathcal{G}$, respectively. We abbreviate the conditional probability $P(X_i = j|Pa(X_i) = \mathbf{h})$ as $\theta^i_{j|\mathbf{h}}$ and the corresponding logarithmic probability as $w^i_{j|\mathbf{h}} := \log(\theta^i_{j|\mathbf{h}})$. Each node of the graph corresponds to an RV and the edges of the graph determine dependencies between these RVs. Throughout this paper, we denote X_0 as C, i.e. X_0 represents the class, and assume that C has no parents in $\mathcal{G}$, i.e. $Pa(C) = \emptyset$. A BN induces a joint probability $\mathrm{P}^\mathcal{B}(C, X_1, \ldots, X_L)$ by multiplying the local conditional distributions together, i.e.

$$\mathrm{P}^\mathcal{B}(C, X_1, \ldots, X_L) = \mathrm{P}(C) \prod_{i=1}^{L} \mathrm{P}(X_i|Pa(X_i)). \tag{3}$$

BNs for classification [9] can be optimized in two ways: firstly, one can select the graph structure $\mathcal{G}$, and secondly, one can learn the conditional probabilities $\mathcal{P}_\mathcal{G}$. Selecting the graph structure is known as structure learning and selecting $\mathcal{P}_\mathcal{G}$ is known as parameter learning. Throughout this paper, we consider NB structures only.

For learning the parameters $\mathcal{P}_\mathcal{G}$ of a BN two paradigms exist, namely generative parameter learning and discriminative parameter learning [1]: In *generative parameter learning* one aims at identifying parameters representing the generative process that results in the data of the training set. An example of this paradigm is maximum likelihood (ML) learning. Its objective is maximization of the likelihood of the data given the parameters. Formally, ML parameters $\mathcal{P}^{\mathrm{ML}}_\mathcal{G}$ are learned as

$$\mathcal{P}^{\mathrm{ML}}_\mathcal{G} = \arg\max_{\mathcal{P}_\mathcal{G}} \prod_{n=1}^{N} \mathrm{P}^\mathcal{B}(c^{(n)}, \mathbf{x}^{(n)}), \tag{4}$$

where $\mathrm{P}^\mathcal{B}(C, \mathbf{X})$ is the joint distribution in (3) induced by the BN $(\mathcal{G}, \mathcal{P}_\mathcal{G})$.

In *discriminative learning* one aims at identifying parameters leading to good classification performance on new samples drawn from $\mathrm{P}^*(C, \mathbf{X})$. Several objectives for this purpose are known in the literature, e.g. the maximum conditional likelihood (MCL) [10] objective and the maximum margin (MM) [11, 1] objective. Throughout this paper, we consider the MM objective as a representative for discriminative parameter learning.

MM parameters $\mathcal{P}^{\mathrm{MM}}_\mathcal{G}$ are found as

$$\mathcal{P}^{\mathrm{MM}}_\mathcal{G} = \arg\max_{\mathcal{P}_\mathcal{G}} \prod_{n=1}^{N} \min\left(\gamma, d^{(n)}\right), \tag{5}$$

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

$$d^{(n)} = \frac{\mathrm{P}^\mathcal{B}(c^{(n)}, \mathbf{x}^{(n)})}{\max_{c \neq c^{(n)}} \mathrm{P}^\mathcal{B}(c, \mathbf{x}^{(n)})}, \tag{6}$$

and $\gamma > 1$ is a parameter controlling the margin. In this way, the margin *measures* the likelihood of the n^{th} sample belonging to the correct class $c^{(n)}$ in relation to the strongest competing class. The n^{th} sample is correctly classified if $d^{(n)} > 1$ and vice versa.

3. BOUNDS

In this section, we determine worst-case and best-case bounds on the classification rate achieved by BNCs with reduced precision parameters. In Section 3.1 we shortly review classification in BNCs and introduce the used quantization, in Section 3.2, we derive a worst-case bound using interval arithmetic, and in Section 3.3, we derive probabilistic bounds using quantization theory.

3.1. Classification Revisited and Quantization

In BNCs $\mathcal{B} = (\mathcal{G}, \mathcal{P}_\mathcal{G})$, an unlabeled sample $\mathbf{x}$ is classified according to (2), i.e. it is classified to the class with highest posterior probability given $\mathbf{x}$. This classification can equivalently be performed in the logarithmic domain, i.e. $\mathbf{x}$ is assigned to class c^* such that

$$\begin{aligned} c^* &= \arg\max_c \log \mathrm{P}^\mathcal{B}(c|\mathbf{x}) \\ &= \arg\max_c \log \mathrm{P}^\mathcal{B}(c, \mathbf{x}). \end{aligned} \tag{7}$$

Because of the factorization properties of BNs stated in (3), the equation above can be rewritten as

$$c^* = \arg\max_c \left[\log \mathrm{P}(c) + \sum_{i=1}^{L} \log \mathrm{P}(\mathbf{x}_i|\mathbf{x}_{Pa(X_i)})\right], \tag{8}$$

where $\mathbf{x}_{Pa(X_i)}$ denotes the instantiation of the parents of X_i according to $\mathbf{x}$. In the following, we consider quantization of the terms $\log \mathrm{P}(c)$ and $\mathrm{P}(\mathbf{x}_i|\mathbf{x}_{Pa(X_i)})$, i.e. the log probabilities

$w^i_{j|\mathbf{h}} = \log(\theta^i_{j|\mathbf{h}})$. Given a fixed number of integer bits $b_i \in \mathbb{N}_0$ and fractional bits $b_f \in \mathbb{N}_0$, we denote $\hat{w}^i_{j|\mathbf{h}} := Q(w^i_{j|\mathbf{h}})$, where $Q(\cdot) = Q^{b_i}_{b_f}(\cdot)$ is the quantization operator. The quantizer $Q^{b_i}_{b_f}(\cdot)\colon \mathbb{R}^- \to \mathbb{B}^{b_i}_{b_f}$ performs quantization by rounding [12], where $\mathbb{B}^{b_i}_{b_f} := \{-\sum_{k=-b_f}^{b_i-1} b_k 2^k \colon b_k \in \{0,1\}\}$ is the set of all negative fixed point numbers with b_i integer bits and b_f fractional bits.

3.2. Deterministic Case: Worst-Case and Best-Case Bounds

Consider the BNC $\mathcal{B} = (\mathcal{G}, \mathcal{P}_\mathcal{G})$ with log parameters $w^i_{j|\mathbf{h}}$. Denote by $\mathcal{B}_Q$ the BNC with parameters $\hat{w}^i_{j|\mathbf{h}} = Q(w^i_{j|\mathbf{h}})$. Note that these quantized log parameters are not properly normalized in general, i.e. $\sum_j \exp(\hat{w}^i_{j|\mathbf{h}}) \neq 1$. For ease of analysis we ignore this fact. We want to bound the generalization error of $\mathcal{B}_Q$ in terms of the generalization error of $\mathcal{B}$. Therefore, note that the largest error due to quantization is $\Delta := 2^{-b_f-1}$, i.e. $|Q(\alpha) - \alpha| \leq \Delta$ for all possible $\alpha \in \mathbb{R}^-$ (we ignore cases in which α is larger than the largest value representable by the chosen number format). The generalization error $\mathrm{Err}(h_{\mathrm{P}^{\mathcal{B}_Q}(C,\mathbf{X})})$ can be worst-case bounded as

$$\begin{aligned}
&\mathrm{Err}(h_{\mathrm{P}^{\mathcal{B}_Q}(C,\mathbf{X})}) = \qquad (9)\\
&= \mathbb{E}_{\mathrm{P}^*(C,\mathbf{X})}\left[\mathbf{1}\{C \neq h_{\mathrm{P}^{\mathcal{B}_Q}(C,\mathbf{X})}(\mathbf{X})\}\right],\\
&= \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) < \max_{c'\neq c} \log \mathrm{P}^{\mathcal{B}_Q}(c',\mathbf{x})\}\\
&\leq \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) < \max_{c'\neq c} (\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + 2(L+1)\Delta)\},
\end{aligned}$$

where the inequality follows because

$$\begin{aligned}
\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) &= \hat{w}_c + \sum_{i=1}^{L} \hat{w}^i_{j|\mathbf{h}} \qquad (10)\\
&\geq w_c - \Delta + \sum_{i=1}^{L} (w^i_{j|\mathbf{h}} - \Delta)\\
&= \log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) - (L+1)\Delta
\end{aligned}$$

and similarly $\log \mathrm{P}^{\mathcal{B}_Q}(c',\mathbf{x}) \leq \log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + (L+1)\Delta$. In general, this worst-case bound can not be achieved because the quantization error introduced on a specific log probability favors the correct classification of some samples while degrading that of others. Note that this bound depends on the margin (6) of the samples, i.e. in case of a large sample margin even coarse quantization may not change the classification of certain samples, while samples with a small margin are prone to misclassification under quantization errors.

Similarly, a best-case bound can be determined. It reads as

$$\begin{aligned}
&\mathrm{Err}(h_{\mathrm{P}^{\mathcal{B}_Q}(C,\mathbf{X})}) \qquad (11)\\
&\geq \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) < \max_{c'\neq c} (\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) - 2(L+1)\Delta)\}.
\end{aligned}$$

Any classifier after parameter quantization must not perform better than this bound.

Note that evaluating the bounds does not require to actually quantize the parameters, as the bounds are obtained by evaluating $\mathrm{P}^{\mathcal{B}}(C,\mathbf{X})$.

3.3. Stochastic Case: Probabilistic Performance Bounds

In the last section we determined performance bounds for BNCs with reduced precision parameters $\mathcal{B}_Q$. Now, we aim at employing results from quantization theory to obtain tighter bounds on the classification performance holding with a user specified probability.

Assume that we estimate the parameters of the BN $\mathcal{B}$ from training set $\mathcal{D}$ using ML. That is, the parameters are the outcome of a random experiment, i.e. the samples in $\mathcal{D}$ are drawn i.i.d. from $\mathrm{P}^*(C,\mathbf{X})$. Consequently, if we were provided another training set $\mathcal{D}'$ and estimated the parameters of another BN $\mathcal{B}'$, then in general $\mathrm{P}^{\mathcal{B}}(C,\mathbf{X}) \neq \mathrm{P}^{\mathcal{B}'}(C,\mathbf{X})$. The estimated parameters vary around the true (optimal) parameters. The more samples are provided, the more accurate the estimated parameters are, i.e. the variance of the parameter estimators reduces. Hence, ML parameters are distributed around the optimal ML parameters. For simplicity, we assume that the estimate of the parameters of $\mathcal{B}$ is uniformly distributed in the quantization interval. However, if plenty of training data is available, the parameter estimates are accurate and will not *span* the quantization interval.

The quantized log probabilities $\hat{w}^i_{j|\mathbf{h}}$ can be written as $\hat{w}^i_{j|\mathbf{h}} = w^i_{j|\mathbf{h}} + e^i_{j|\mathbf{h}}$, where $e^i_{j|\mathbf{h}}$ is the quantization error uniformly distributed in $[-\Delta,\Delta]$. The expected value of this error is zero and its variance is $2^{-2b_f}/12$, cf. [12]. The joint probability $\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x})$ can now be written as

$$\begin{aligned}
\log \mathrm{P}^{\mathcal{B}_Q}(c,\mathbf{x}) &= \hat{w}_c + \sum_{i=1}^{L} \hat{w}^i_{j|\mathbf{h}} \qquad (12)\\
&= w_c + e_c + \sum_{i=1}^{L} (w^i_{j|\mathbf{h}} + e^i_{j|\mathbf{h}})\\
&= \log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) + E(c,\mathbf{x}),
\end{aligned}$$

where $E(c,\mathbf{x}) := e_c + \sum_{i=1}^{L} e^i_{j|\mathbf{h}}$. The term $E(c,\mathbf{x})$ is the sum of $(L+1)$ uniformly distributed RVs. Hence, it is distributed according to the mean-centered Irwin-Hall distribution [13]. If $(L+1)$ is sufficiently large, this distribution can be approximated accurately by a truncated normal distribution with zero mean, variance $(L+1)2^{-2b_f}/12$ and minimum/maximum value of $-(L+1)\Delta$ and $(L+1)\Delta$, respectively.

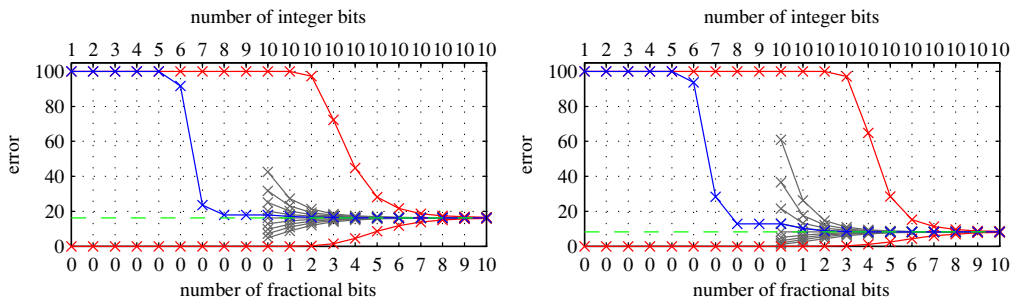
Employing the cumulative distribution function (CDF) $F_{E(c,\mathbf{x})}(\cdot)$ of $E(c,\mathbf{x})$, we can determine with a certain *confidence* level p the largest value of $E(c,\mathbf{x})$. For example, with a confidence of 100% the term $E(c,\mathbf{x})$ is smaller than $(L+1)\Delta$. We can compute similar bounds B_p on the value of $E(c,\mathbf{x})$ with other confidences like $90\%, 80\%, \ldots$. The values of B_p can be used in conjunction with (9) yielding probabilistic worst-case bounds of the form

$$\begin{aligned}
&\mathrm{Err}(h_{\mathrm{P}^{\mathcal{B}_Q}(C,\mathbf{X})}) = \qquad (13)\\
&\leq \sum_{c,\mathbf{x}} \mathrm{P}^*(c,\mathbf{x})\mathbf{1}\{\log \mathrm{P}^{\mathcal{B}}(c,\mathbf{x}) < \max_{c'\neq c} (\log \mathrm{P}^{\mathcal{B}}(c',\mathbf{x}) + 2B_p)\}.
\end{aligned}$$

The bound with confidence $p = 100\%$ corresponds to the worst-case bound, i.e. $B_p = (L+1)\Delta$, and the bound with confidence $p = 0\%$ corresponds to the best-case bound, i.e. $B_p = -(L+1)\Delta$. For $p = 50\%$, the bound $B_p = 0$, i.e. the generalization error of $\mathcal{B}_Q$ equals that of $\mathcal{B}$ with a confidence of 50 percent.

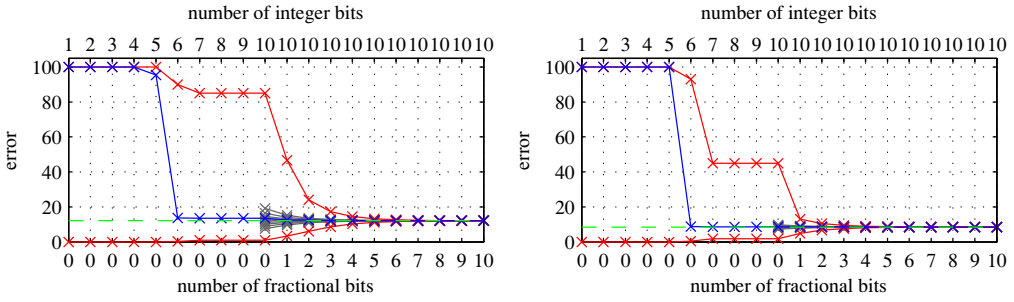
4. EXPERIMENTS

We present classification results for MNIST [4] and TIMIT data [14] using BNCs with reduced precision parameters and NB structure.



(a) MNIST data, ML parameters

(b) MNIST data, MM parameters



(c) TIMIT data, ML parameters

(d) TIMIT data, MM parameters

Fig. 2. Generalization error of BNCs with NB structure and double-precision parameters (dashed green line), reduced precision parameters (solid blue line), worst-case and best-case bounds (red lines), and $90\%, \ldots, 10\%$ confidence bounds (gray lines, from top to bottom).

The first dataset deals with handwritten digit recognition and the second with phonetic classification. Details about the data are not relevant here but provided in [1].

The experiments are performed as follows: In a first step, we learn BNCs with ML and MM parameters for each dataset using double-precision floating point numbers. Then, we quantize the determined log parameters using fixed-point numbers with varying precision, i.e. we quantize the previously determined CPTs. We select different values for the number of integer bits b_i and the number of fractional bits b_f, and evaluate the classification performance of the resulting BNCs on a separate test set. First, b_i is increased up to ten 10 bits[1], then b_f is increased up to 10 bits. All computations, i.e. the classification process, is performed using the same precision. Addition of the log probabilities is performed using saturation. Whenever $b_i = 10$, we also compute the stochastic bounds derived in the last section. Results for MNIST and TIMIT data using BNCs with NB structure and ML parameters are shown in Figures 2(a) and 2(c), and for BNCs with NB structure and MM parameters in Figures 2(b) and 2(d).

We observe that for MNIST data, 8 integer bits are required to achieve classification performance close to optimal. Additional integer bits do not increase the classification performance. When using less than 7 integer bits, classification performance degrades severely, even though the integer part of every log parameter of the classifier can be represented exactly. This is because of the saturation occurring when adding up the log probabilities. The bounds tighten with an increasing number of fractional bits and are almost tight for 3 or more fractional bits. Similar observations can be made for TIMIT.

[1]Generally we fix b_i to a maximum of 10 bits. This number could be reduced while still observing similar classification performance.

5. RELATION TO PRIOR WORK

In [3], the effect of parameter quantization in BNCs with focus on comparing the robustness of BNCs with generatively and discriminatively optimized parameters is investigated. The authors use bit-width reduced floating point parameters. Furthermore, in [5] CR performance with respect to reduced fixed point precision parameters has been analyzed.

Indirectly related work deals with (a) *sensitivity analysis* of Bayesian networks [15, 16], stating essentially that classification using BNCs is insensitive to parameter deviations whenever either these parameters are not close to zero or one, or the class posteriors are significantly different, (b) credal networks, i.e. generalizations of BNs that associate a whole set of CPDs with every node in the DAG [17], allowing for robust classification and incorporating that CPDs can often not be specified exactly.

6. CONCLUSIONS AND FUTURE WORK

We investigated quantization effects in BNCs with reduced precision parameters, i.e. the parameters were represented by fixed-point numbers of a specified precision. We determined deterministic and probabilistic performance bounds and evaluated these bounds in experiments. The bounds allow to quantify the impact of parameter quantization on classification performance.

In future work, we aim at deriving more accurate bounds by making more realistic assumptions. Further, we aim at quantifying the worst-case decrease of classification rate that can result for BNCs with reduced precision parameters for specific DAGs without considering underlying data. Additionally, we want to derive parameter learning algorithms for reduced-precision fixed-point parameters.

7. REFERENCES

[1] F. Pernkopf, M. Wohlmayr, and S. Tschiatschek, "Maximum margin Bayesian network classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 34, no. 3, pp. 521–531, 2012.

[2] M. J. Druzdzel and A. Onisko, "Are Bayesian networks sensitive to precision of their parameters?" in *Intelligent Information Systems XVI*, 2008, pp. 35–44.

[3] S. Tschiatschek, P. Reinprecht, M. Mücke, and F. Pernkopf, "Bayesian network classifiers with reduced precision parameters," in *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, 2012, pp. 74–89.

[4] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[5] F. Pernkopf, M. Wohlmayr, and M. Mücke, "Maximum margin structure learning of bayesian network classifiers," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2011, pp. 2076–2079.

[6] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.

[7] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., 1988.

[8] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

[9] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997.

[10] T. Roos, H. Wettig, P. Grünwald, P. Myllymäki, and H. Tirri, "On discriminative Bayesian network classifiers and logistic regression," *Machine Learning*, vol. 59, no. 3, pp. 267–296, 2005.

[11] Y. Guo, D. Wilkinson, and D. Schuurmans, "Maximum margin Bayesian networks," in *Uncertainty in Artificial Intelligence (UAI)*, 2005, pp. 233–242.

[12] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-time signal processing (2nd ed.)*. Prentice-Hall, Inc., 1999.

[13] N. Johnson, S. Kotz, and N. Balakrishnan, *Continuous Univariate Distributions*, 2nd ed., ser. Distributions in statistics. Wiley, 1995.

[14] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Speech Input/Output Assessment and Speech Databases*, vol. 2, 1989, pp. 61–70.

[15] H. Chan and A. Darwiche, "When do numbers really matter?" *Articial Intelligence Research*, vol. 17, no. 1, pp. 265–287, 2002.

[16] ——, "Sensitivity analysis in Bayesian networks: From single to multiple parameters," in *Uncertainty in Artificial Intelligence (UAI)*, 2004, pp. 67–75.

[17] M. Zaffalon, "Credal networks classification," Tech. Rep., 1999.