Bayesian Network Classifiers with Reduced Precision Parameters

Sebastian Tschiatschek[1], Peter Reinprecht[1],
Manfred Mücke[2,3], and Franz Pernkopf[1]

[1] Signal Processing and Speech Communication Laboratory
Graz University of Technology, Graz, Austria
[2] University of Vienna, Research Group
Theory and Applications of Algorithms, Vienna, Austria
[3] Sustainable Computing Research, Austria
http://www.spsc.tugraz.at

Abstract. Bayesian network classifiers (BNCs) are probabilistic classifiers showing good performance in many applications. They consist of a directed acyclic graph and a set of conditional probabilities associated with the nodes of the graph. These conditional probabilities are also referred to as parameters of the BNCs. According to common belief, these classifiers are insensitive to deviations of the conditional probabilities under certain conditions. The first condition is that these probabilities are not too extreme, i.e. not too close to 0 or 1. The second is that the posterior over the classes is significantly different. In this paper, we investigate the effect of precision reduction of the parameters on the classification performance of BNCs. The probabilities are either determined generatively or discriminatively. Discriminative probabilities are typically more extreme. However, our results indicate that BNCs with discriminatively optimized parameters are almost as robust to precision reduction as BNCs with generatively optimized parameters. Furthermore, even large precision reduction does not decrease classification performance significantly. Our results allow the implementation of BNCs with less computational complexity. This supports application in embedded systems using floating-point numbers with small bit-width. Reduced bit-widths further enable to represent BNCs in the integer domain while maintaining the classification performance.

Keywords: Bayesian Network Classifiers, Custom-precision Analysis, Discriminative Classifiers.

1 Introduction

Pattern recognition is about identifying patterns in input data and assigning labels to this data. Examples of pattern recognition are regression and classification. A classifier has to be learned from a set of training samples by identifying discriminative properties such that new unlabeled samples can be correctly classified. Many approaches and algorithms for this purpose exists. Some of the most

P. Flach et al. (Eds.): ECML PKDD 2012, Part I, LNCS 7523, pp. 74–89, 2012.

competitive approaches are support vector machines [16], neural networks [7] and Bayesian network classifiers (BNCs) [5].

BNCs are probabilistic classifiers that assume a joint probability distribution over the input data and the class labels. They classify new input data as the maximum a-posteriori estimate of the class given this data using the assumed probability distribution. The probability distribution is represented by a Bayesian network (BN). BNs consist of a directed acyclic graph, i.e. the structure, and a set of local conditional probability densities, i.e. the parameters. The classification performance of a BNC is determined by the assumed probability distribution. Finding probability distributions that result in good classifiers is addressed by the tasks of structure [1,5,8,14] and parameter learning [6,8,12,15,16]. Structure learning is not considered in this paper and we assume fixed graph structures. In detail, we consider BNCs with naive Bayes (NB) structures, cf. Figure 1, and tree augmented network structures (TAN) [5].

Parameter learning in BNCs resorts to identifying a probability distribution over the input data and the class labels. This distribution must be compatible with the assumed structure of the BNC. For learning these distributions, we use the maximum likelihood (ML), the maximum conditional likelihood (MCL), and the maximum margin (MM) objectives.

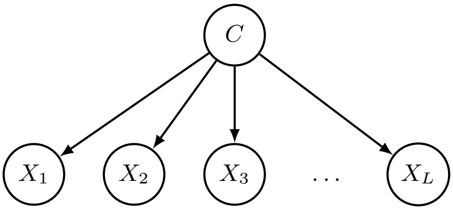


Fig. 1. Naive Bayes structure

The process of parameter learning and classification is typically performed on a computer using high numerical precision, i.e. double-precision floating-point calculations. However, this high precision causes large storage requirements, cf. Table 1. Additionally, the necessary calculations depend on complex computer architectures to be performed efficiently. In contrast to up-to-date computers this requirements are often not met by embedded systems, low energy computers or integrated solutions that need to optimize the used hardware resources. To aid complexity reduction we investigate the performance of BNCs with reduced precision probability parameters. Especially, we are interested in comparing the robustness of generatively (ML) and discriminatively (MCL, MM) optimized probability distributions with respect to precision reduction of their parameters using various BN structures.

Some of our findings can be related to results from sensitivity analysis of BNs [3, 4]. Amongst others, the framework of sensitivity analysis describes the

Table 1. Number of probability parameters (# parameters) and the storage requirements (storage) for these parameters in BNCs with different graph structures (for different datasets). Each parameter is assumed to be stored in double-precision floating-point format, i.e. 64 bits are required for each parameter. Details on the structures and datasets are provided in Section 5.

data	structure	# parameters	storage [kB]
USPS	NB	8650	67.6
	TAN-CR	20840	162.8
MNIST	NB	6720	52.5
	TAN-CR	39980	312.3
TIMIT (4 classes)	NB	1320	10.3
TIMIT (6 classes)	NB	1998	15.6

dependency of inference queries to variations in the local conditional probability parameters. The precision reduction of the probability parameters resorts to such variations and can, therefore, be interpreted in this framework. However, the focus in this paper is different. We are particularly interested in analyzing the classification performance of BNCs when reducing the bit-width of all parameters simultaneously. Additionally, we are interested in comparing the robustness of the classification of BNCs with generatively and discrimatively optimized parameters with respect to this precision reduction. As the local conditional probability parameters of discriminatively optimized BNCs tend to be more extreme, we suspected classification rates of these classifiers to depend stronger on the used precision than the classification rates of BNCs with generatively optimized parameters. Nevertheless, our results demonstrate that this is not true.

Our main findings are:

- The number of extreme conditional probability values, i.e. probabilities close to 0 or 1, in BNCs with discrimatively optimized parameters is larger than in BNCs with generatively optimized parameters, cf. Section 5.1. Using results from sensitivity analysis, this suggests that BNCs with discrimatively optimized parameters might be more susceptible to precision reduction than BNCs with generatively optimized parameters. Nevertheless, we observed in experiments that BNCs with both types of parameters can achieve good classification performance using reduced precision floating-point parameters. In fact, the classification performance is close to BNCs with parameters represented in full double-precision floating-point format, cf. Section 5.2.
- The reduction of the precision allows for mapping the classification process of BNCs to the integer domain, cf. Section 4. Thereby, exact computation in that domain, reduced computational complexity and implementation on simple embedded hardware is supported. In fact, some of the considered BNCs can perform classification using integer arithmetic without significant reduction of performance.

The outline of this paper is as follows: In Section 2 we provide a motivating example demonstrating that there is large potential in reducing the precision of the parameters of BNCs. Afterwards, we introduce probabilistic classification, BNCs, and the sensitivity of BNs to changes of their parameters in Section 3. An approach for mapping the parameters of BNCs to the integer domain is presented in Section 4 and various experiments are provided in Section 5. Finally, we conclude the paper in Section 6 and provide a perspective on future work.

2 Motivating Example

In this section we provide an example demonstrating that the parameters of BNs employed for classification do not require high precision. They can be approximated coarsely without reducing the classification rate significantly. In some cases, only a few bits for representing each probability parameter of a BNC are necessary to achieve classification rates close to optimal.

The probability parameters of BNCs, these classifiers are introduced in detail in Section 3, are typical stored in double-precision floating-point format [10, 11]. We use logarithmic probability parameters $w = \log(\theta)$, with $0 \leq \theta \leq 1$, represented as

$$w = (-1)^s \left(1 + \sum_{k=1}^{52} b^k 2^{-k}\right) 2^{\left(\sum_{l=0}^{10} e^l 2^l - 1023\right)}, \tag{1}$$

where $s \in \{0, 1\}$, $b^k \in \{0, 1\}$ for all k, and $e^l \in \{0, 1\}$ for all l. The term

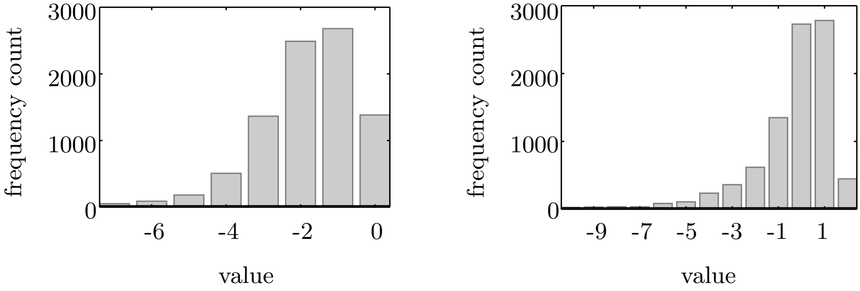
- $(-1)^s$ is the *sign*,
- $(1 + \sum_{i=1}^{52} b^k 2^{-k})$ is the *mantissa*, and
- $(\sum_{l=0}^{10} e^l 2^l - 1023)$ is the *exponent*

of w, respectively. In total 64 bits are used to represent each log-parameter. Processing these parameters on desktop computers does not impose any problems. However, this large bit-width of the parameters can be a limiting factor in embedded systems or applications optimized for low run-times or low energy-consumption.

The range of the parameters using double-precision floating-point format is about $\pm 10^{300}$ and by far larger than required; The distribution of the log-parameters of a BNC with maximum likelihood parameters for handwritten digit data (USPS data, details are provided in Section 5) is shown in Figure 2(a). Additionally, the distribution of the values of the exponent is shown in Figure 2(b). All the log-parameters are negative and their range is $[-7; 0]$. The range of the exponent of the logarithmic parameters is $[-10; 2]$.

The required bit-width to store the logarithmic parameters in a floating-point format, cf. Equation (1), can be reduced in three aspects:

1. **Sign bit.** Every probability θ satisfies $0 \leq \theta \leq 1$. Therefore, its logarithm is in the range $-\infty \leq w \leq 0$. Consequently, the sign bit can be removed without any change in the represented parameters.



(a) logarithmic probabilities

(b) exponent of logarithmic conditional probabilities in double-precision

Fig. 2. Histograms of (a) the log-parameters, and (b) the exponents of the log-parameters of a BNC for handwritten digit data with ML parameters assuming NB structure.

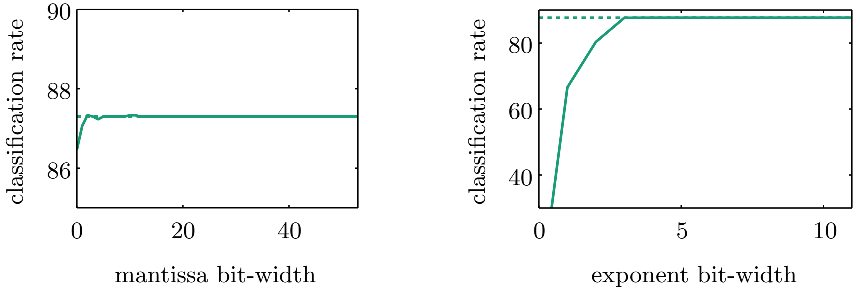
2. **Bit-width of the mantissa.** We varied the bit-width of the mantissa of the log-parameters while keeping the exponent unchanged. As a result, we observed that this does not influence the classification rate significantly when using ML parameters, cf. Figure 3(a). When using 4 or more bits to represent the mantissa, the performance is almost the same as when using the full double-precision floating-point format, i.e. 53 bits for the mantissa.
3. **Bit-width of the exponent.** Changing the bit-width of the exponent has the largest impact on the classification performance. A change of the exponent of a parameter results in a change of the scale of this parameter. The classification rates resulting from reducing the bit-width of the exponent are shown in Figure 3(b). Note that we reduced the bit-width starting with the most significant bit (MSB). Only a few bits are necessary for classification rates on par with the rates achieved using full double-precision floating-point parameters.

Based on this motivating example demonstrating the potential of precision reduction we can even map BNCs to the integer domain, cf. Section 4. Further experimental results are shown in Section 5.

3 Background

3.1 Probabilistic Classification

Probabilistic classifiers are embedded in the framework of probability theory. One assumes a random variable (RV) C denoting the class and RVs $X_1, \ldots, X_L$ representing the attributes/features of the classifier. These RVs are related by a joint probability distribution $\mathrm{P}^*(C, \mathbf{X})$, where $\mathbf{X} = [X_1, \ldots, X_L]$ is a random vector consisting of $X_1, \ldots, X_L$. In typical settings, this joint distribution is unknown and a limited number of samples drawn from true distribution $\mathrm{P}^*(C, \mathbf{X})$,



(a) varying mantissa bit-width, using full bit-width for exponent

(b) varying exponent bit-width, using full bit-width for mantissa

Fig. 3. Classification rate over varying bit-width of (a) the mantissa, and (b) the exponent, for handwritten digit data, NB structure, and log ML parameters. The classification rates using full double-precision logarithmic parameters are indicated by the horizontal dotted lines.

i.e. a training set $\mathcal{D}$, is available. This set $\mathcal{D}$ consists of N i.i.d. labeled samples, i.e. $\mathcal{D} = \{(c^{(n)}, \mathbf{x}^{(n)}) | 1 \leq n \leq N\}$, where $c^{(n)}$ denotes the instantiation of the RV C and $\mathbf{x}^{(n)}$ the instantiation of $\mathbf{X}$ in the n^{th} training sample. The aim is to induce *good* classifiers provided the training set, i.e. classifiers with low generalization error. Formally, a classifiers h is a mapping

$$\begin{aligned} h: \quad & \text{sp}(\mathbf{X}) \to \text{sp}(C), \\ & \mathbf{x} \mapsto h(\mathbf{x}), \end{aligned} \tag{2}$$

where sp($\mathbf{X}$) denotes the set of all assignments of $\mathbf{X}$ and sp(C) is the set of classes. The generalization error of this classifier is

$$\text{Err}(h) := \mathbb{E}_{\text{P}^*(C,\mathbf{X})} \left[\mathbf{1}\{C \neq h(\mathbf{X})\} \right], \tag{3}$$

where $\mathbf{1}\{A\}$ denotes the indicator function and $\mathbb{E}_{\text{P}^*(C,\mathbf{X})}[\cdot]$ is the expectation operator with respect to the distribution $\text{P}^*(C, \mathbf{X})$. The indicator function $\mathbf{1}\{A\}$ equals one if statement A is true and zero otherwise. Typically, the generalization error can not be evaluated because $\text{P}^*(C, \mathbf{X})$ is unknown but is rather estimated using cross-validation [2].

BNCs with generatively optimized parameters are based on the idea of *approximating* $\text{P}^*(C, \mathbf{X})$ by a distribution $\text{P}^{\mathcal{B}}(C, \mathbf{X})$ and using the induced classifier $h_{\text{P}^{\mathcal{B}}(C,\mathbf{X})}$, given as

$$\begin{aligned} h_{\text{P}^{\mathcal{B}}(C,\mathbf{X})}: \quad & \text{sp}(\mathbf{X}) \to \text{sp}(C), \\ & \mathbf{x} \mapsto \arg\max_{c \in C} \text{P}^{\mathcal{B}}(C = c | \mathbf{X} = \mathbf{x}), \end{aligned} \tag{4}$$

for classification. 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 $\text{P}^{\mathcal{B}}(C, \mathbf{X})$. BNCs with discriminatively optimized parameters do not approximate $\mathrm{P}^*(C, \mathbf{X})$ but rather determine $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X})$ such that good classification performance is achieved. Discriminative learning of BNCs is advantageous in cases where the assumed model distribution $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X})$ can not approximate $\mathrm{P}^*(C, \mathbf{X})$ well, for example because of a too limited BN structure. Several approaches for optimizing $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X})$ are discussed in the next section after introducing the concept of Bayesian networks in more detail.

3.2 Bayesian Networks and Learning Bayesian Network Classifiers

Bayesian Networks (BNs) [8,12] 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.

$$\mathrm{P}^{\mathcal{B}}(C, X_1, \ldots, X_L) = \mathrm{P}(C) \prod_{i=1}^{L} \mathrm{P}(X_i|Pa(X_i)). \tag{5}$$

BNs for classification 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. The structures considered throughout this paper are fairly simple. In detail, we used naive Bayes structures, cf. Figure 1, and tree augmented network structures (TAN) [5].

For learning the parameters $\mathcal{P}_{\mathcal{G}}$ of a BN two paradigms exist, namely generative parameter learning and discriminative parameter learning:

- 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

$$\mathcal{P}^{\mathrm{ML}}_{\mathcal{G}} = \arg\max_{\mathcal{P}_{\mathcal{G}}} \prod_{n=1}^{N} \mathrm{P}^{\mathcal{B}}(c^{(n)}, \mathbf{x}^{(n)}), \tag{6}$$

where $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X})$ is the joint distribution in (5) 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 from $\mathrm{P}^*(C, \mathbf{X})$. Several objectives for this purpose are known in the literature. Throughout this paper, we consider the maximum conditional likelihood (MCL) [15] objective and the maximum margin (MM) [6,13] objective.
  MCL parameters $\mathcal{P}_{\mathcal{G}}^{\mathrm{MCL}}$ are obtained as
  $$\mathcal{P}_{\mathcal{G}}^{\mathrm{MCL}} = \arg\max_{\mathcal{P}_{\mathcal{G}}} \prod_{n=1}^{N} \mathrm{P}^{\mathcal{B}}(c^{(n)}|\mathbf{x}^{(n)}), \tag{7}$$
  where again $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X})$ is the joint distribution induced by the BN $(\mathcal{G}, \mathcal{P}_{\mathcal{G}})$ and $\mathrm{P}^{\mathcal{B}}(C|\mathbf{X})$ denotes the conditional distribution of C given $\mathbf{X}$ determined from $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X})$ as $\mathrm{P}^{\mathcal{B}}(C, \mathbf{X}) = \mathrm{P}^{\mathcal{B}}(C|\mathbf{X}) \cdot \mathrm{P}^{\mathcal{B}}(\mathbf{X})$. Thus, MCL parameters maximize the conditional probability of the class instantiations given the instantiations of the attributes.
  MM parameters $\mathcal{P}_{\mathcal{G}}^{\mathrm{MM}}$ are found as
  $$\mathcal{P}_{\mathcal{G}}^{\mathrm{MM}} = \arg\max_{\mathcal{P}_{\mathcal{G}}} \prod_{n=1}^{N} \min\left(\gamma, d^{(n)}\right), \tag{8}$$
  where $d^{(n)}$ is the margin of the n^{th} sample given as
  $$d^{(n)} = \frac{\mathrm{P}^{\mathcal{B}}(c^{(n)}|\mathbf{x}^{(n)})}{\max_{c \neq c^{(n)}} \mathrm{P}^{\mathcal{B}}(c|\mathbf{x}^{(n)})}, \tag{9}$$
  and $\gamma > 1$ is a parameter controlling the margin. In this way, the margin *measures* the ratio of the likelihood of the n^{th} sample belonging to the correct class $c^{(n)}$ to belonging to the strongest competing class. The n^{th} sample is correctly classified if $d^{(n)} > 1$ and vice versa.

3.3 Sensitivity of Bayesian Networks

The *sensitivity* of a BN $\mathcal{B} = (\mathcal{G}, \mathcal{P}_{\mathcal{G}})$ describes the change in a query with respect to changes in the local conditional probabilities in $\mathcal{P}_{\mathcal{G}}$. For example, a query is the calculation of a posterior probability of the form $\mathrm{P}^{\mathcal{B}}(\mathbf{X}_q|\mathbf{X}_e)$, with $\mathbf{X}_q, \mathbf{X}_e \subseteq \{C, X_1, \ldots, X_L\}$ and $\mathbf{X}_q \cap \mathbf{X}_e = \emptyset$. Several results on estimating and bounding this sensitivity exist in the literature, cf. for example [3,17]. The results therein essentially state that the sensitivity of BNs depends mainly on probability parameters being close to 0 or 1 and queries being close to uniform.

In this context, consider the following theorem:

Theorem 1 (from [3]). *Let X_i be a binary RV in a BN $\mathcal{B} = (\mathcal{G}, \mathcal{P}_{\mathcal{G}})$, then*

$$\left|\frac{\partial \mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e)}{\partial \tau_{X_i|Pa(X_i)}}\right| \leq \frac{\mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e) \cdot (1 - \mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e))}{\mathrm{P}^{\mathcal{B}}(X_i|Pa(X_i)) \cdot (1 - \mathrm{P}^{\mathcal{B}}(X_i|Pa(X_i))}, \tag{10}$$

where $\tau_{X_i|Pa(X_i)}$ *is a meta-parameter such that* $\mathrm{P}^{\mathcal{B}}(X_i = 0|Pa(X_i)) = \tau_{X_i|Pa(X_i)}$ *and* $\mathrm{P}^{\mathcal{B}}(X_i = 1|Pa(X_i)) = 1 - \tau_{X_i|Pa(X_i)}$.

The theorem states that the magnitude of the partial derivative of $\mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e)$ with respect to $\tau_{X_i|Pa(X_i)}$ is bounded above. The bound depends on the query under the current parameters $\mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e)$ and on the conditional probabilities $\mathrm{P}^{\mathcal{B}}(X_i|Pa(X_i))$. The partial derivative is large whenever $\mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e)$ is close to uniform and whenever $\mathrm{P}^{\mathcal{B}}(X_i = 0|Pa(X_i))$ is close to 0 or 1. In classification the query of interest is the probability of the class variable given the features, i.e. $\mathrm{P}^{\mathcal{B}}(X_i|\mathbf{X}_e) = \mathrm{P}^{\mathcal{B}}(C|\mathbf{X})$. Discriminative objectives for parameter learning in BNs aim at good class separation, i.e. $\mathrm{P}^{\mathcal{B}}(C|\mathbf{X})$ or $1 - \mathrm{P}^{\mathcal{B}}(C|\mathbf{X})$ is typically large. However, also the parameters tend to be extreme, i.e. $\mathrm{P}^{\mathcal{B}}(X_i|Pa(X_i))$ is close to 0 or 1 (some empirical results supporting this are shown in Section 5.1). We expect the bound to be large for discriminatively optimized parameters, as the denominator in the above theorem scales the bound inversely proportional [3]. Hence, either the bound is loose or the partial derivative is actually large resulting in high sensitivity to parameter deviations. This could be the tripping hazard for BNCs with discriminatively optimized parameters. However, experimental observations in Section 5.2 show a robust classification behavior using discriminatively optimized small bit-width parameters.

The above Theorem only describes the sensitivity with respect to a single parameter. There are some extensions of sensitivity analysis describing the sensitivity of queries with respect to changes of many parameters [4]. However, to the best of the authors knowledge, these do not extend to changes of all parameters, which is the focus of this paper. Furthermore, in classification we are not directly interested in the sensitivity of certain queries. The focus is rather on the maximum of a set of queries, i.e. the sensitivity of the MAP classification. Further analytical analysis is intended for future work.

4 BNCs in the Integer Domain

In this section we present how to cast classification using BNCs to the integer domain. This is possible when using reduced precision log-parameters for the BNCs. Without reduced precision, the mapping can not be achieved considering the large range of numbers representable by double-precision floating-point numbers.

Remember, a BNC given by the BN $\mathcal{B} = (\mathcal{G}, \mathcal{P}_{\mathcal{G}})$ assigns an instantiation $\mathbf{x}$ of the attributes to class

$$\begin{aligned} c &= \arg \max_{c' \in \mathrm{sp}(C)} \mathrm{P}^{\mathcal{B}}(c', \mathbf{x}) && (11) \\ &= \arg \max_{c' \in \mathrm{sp}(C)} \mathrm{P}(C = c') \prod_{i=1}^{L} \mathrm{P}(X_i = \mathbf{x}(X_i)|Pa(X_i) = \mathbf{x}(Pa(X_i))), && (12) \end{aligned}$$

where $\mathbf{x}(X_k)$ denotes the entry in $\mathbf{x}$ corresponding to X_k. This classification rule can be equivalently stated in the logarithmic domain, i.e. $\mathbf{x}$ is assigned to class

$$c = \arg \max_{c' \in \mathrm{sp}(C)} \left[\log \mathrm{P}(C = c') + \sum_{i=1}^{L} \log \mathrm{P}(X_i = \mathbf{x}(X_i) | Pa(X_i) = \mathbf{x}(Pa(X_i))) \right]. \quad (13)$$

As shown in Sections 2 and 5 the logarithmic probabilities in the above equation can often be represented using only a few bits without reducing the classification rate significantly. In many cases, 2 bits for the mantissa and 4 bits for the exponent are sufficient to achieve good classification rates. Using these 6 bits, the logarithmic probability $w^i_{j|\mathbf{h}} = \log \theta^i_{j|\mathbf{h}}$ is given as

$$w^i_{j|\mathbf{h}} = -(1 + b^{i,1}_{j|\mathbf{h}} \cdot 2^{-1} + b^{i,2}_{j|\mathbf{h}} \cdot 2^{-2}) \cdot 2^{\left(\sum_{k=0}^{3} e^{i,k}_{j|\mathbf{h}} \cdot 2^k - 7\right)}. \quad (14)$$

Hence,

$$\begin{aligned} c &= \arg \max_{c' \in \mathrm{sp}(C)} \left[w^0_{c'} + \sum_{i=1}^{L} w^i_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} \right] && (15) \\ &= \arg \min_{c' \in \mathrm{sp}(C)} \left[-w^0_{c'} - \sum_{i=1}^{L} w^i_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} \right] && (16) \\ &= \arg \min_{c' \in \mathrm{sp}(C)} \left[(1 + b^{0,1}_{c'} \cdot 2^{-1} + b^{0,2}_{c'} \cdot 2^{-2}) \cdot 2^{\left(\sum_{k=0}^{3} e^{i,k}_{c'} \cdot 2^k - 7\right)} + \right. && (17) \\ &\left. \sum_{i=1}^{L} (1 + b^{i,1}_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} 2^{-1} + b^{i,2}_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} 2^{-2}) \cdot 2^{\left(\sum_{k=0}^{3} e^{i,k}_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} 2^k - 7\right)} \right]. \end{aligned}$$

Multiplying (17) by the constant 2^9 does not change the classification. Hence, classification can be performed by

$$\begin{aligned} c = \arg \min_{c' \in \mathrm{sp}(C)} &\left[(4 + b^{0,1}_{c'} \cdot 2 + b^{0,2}_{c'}) \cdot 2^{\left(\sum_{k=0}^{3} e^{i,k}_{c'} \cdot 2^k\right)} + \right. && (18) \\ &\left. \sum_{i=1}^{L} (4 + b^{i,1}_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} \cdot 2 + b^{i,2}_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))}) \cdot 2^{\left(\sum_{k=0}^{3} e^{i,k}_{\mathbf{x}(X_i)|\mathbf{x}(Pa(X_i))} \cdot 2^k\right)} \right] \end{aligned}$$

which resorts to integer computations only. Furthermore, no floating-point rounding errors of any kind are introduced during computation when working purely in the integer domain. Integer arithmetic is sufficient for implementation.

5 Experiments

In this section we present classification experiments using reduced precision log probability parameters of BNCs. Throughout this section we consider the following three datasets:

- **TIMIT-4/6 Data.** This dataset is extracted from the TIMIT speech corpus using the dialect speaking region 4. It consists of 320 utterances from 16 male and 16 female speakers. Speech frames are classified into either four or six classes using 110134 and 121629 samples, respectively. Each sample is represented by 20 mel-frequency cepstral coefficients (MFCCs) and wavelet-based features [13]. We perform classification experiments on data of both genders (Ma+Fe).
- **USPS Data.** This dataset contains 11000 uniformly distributed handwritten digit images from zip codes of mail envelopes. Each digit is represented as a 16 × 16 grayscale image, where each pixel is considered as feature.
- **MNIST Data [9].** This dataset contains 70000 samples of handwritten digits. The digits represented by gray-level images were down-sampled by a factor of two resulting in a resolution of 16 × 16 pixels, i.e. 196 features.

Some of the experiments are performed using different BN structures. In detail, we considered the naive Bayes (NB) structure, the generative TAN-CMI structure [5] and the discriminative TAN-OMI-CR and TAN-CR structures [14]. The discriminative structures are determined by search-and-score heuristics using the classification rate (CR) as score.

5.1 Number of Extreme Parameter Values in BNCs

We determined BNCs with ML, MCL and MM parameters. For calculating the MCL and MM parameters we used the conjugate gradient based approaches proposed in [13]. However, we did not use the proposed early-stopping heuristic for determining the number of conjugate gradient iterations but rather performed up to 200 iterations (or until there was no further increase in the objective). We then counted the number of conditional probability parameters with a maximal distance of ϵ to the extreme values 0 and 1, i.e. the count is given as

$$M_\epsilon = \sum_{i,j,\mathbf{h}} \mathbf{1}\{(1 - \theta^j_{j|\mathbf{h}}) < \epsilon\} + \sum_{i,j,\mathbf{h}} \mathbf{1}\{\theta^j_{j|\mathbf{h}} < \epsilon\}. \tag{19}$$

The results for USPS and MNIST data are shown in Tables 2(a) and 2(b), respectively. The number of extreme parameter values in BNCs with MCL parameters is larger than in BNCs with MM parameters, and the number of extreme parameter values in BNCs with MM parameters is larger than in BNCs with ML parameters. This suggests that classification using MCL parameters is more sensitive to parameter deviations than classification with MM parameters, and classification using MM parameters is more sensitive to deviations than classification with ML parameters.

5.2 Reduced Precision Classification Performance

We evaluated the classification performance of BNCs with ML, MCL and MM parameters on the USPS, MNIST and TIMIT data. Results are shown in

Table 2. Number of probability parameters $\theta^i_{j|\mathbf{h}}$ close to the extreme values 0 and 1. Additionally, the total number of parameters (# par.) and classification rates (CR) on the test set using parameters in full double-precision floating-point format on (a) USPS data and (b) MNIST data are shown.

(a) USPS

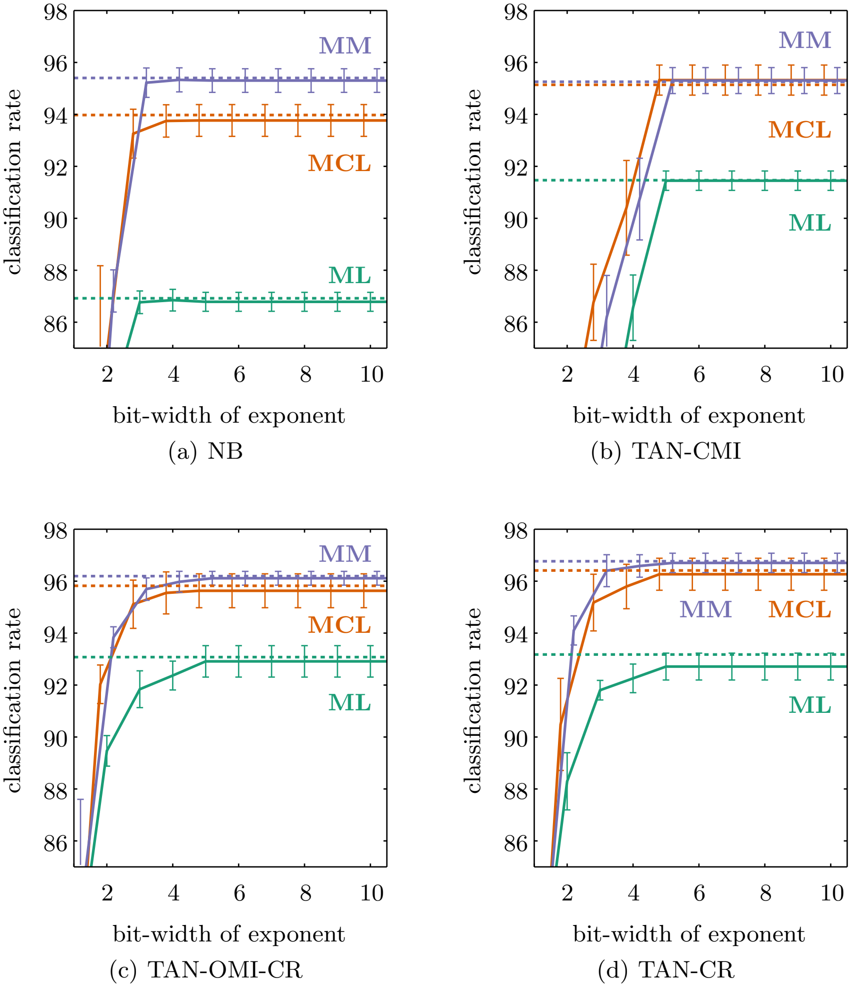
		$M_{0.05}$			$M_{0.01}$			CR		
structure	# par.	ML	MCL	MM	ML	MCL	MM	ML	MCL	MM
NB	8650	1478	4143	1837	364	2134	446	87.10	93.93	95.00
TAN-CMI	33040	12418	14712	13002	8271	9371	8428	91.90	95.70	95.37
TAN-OMI-CR	25380	6677	8167	7441	3486	3937	3624	92.40	95.73	95.40
TAN-CR	20840	5405	7344	6519	2666	3503	3009	92.57	95.97	95.87

(b) MNIST

		$M_{0.05}$			$M_{0.01}$			CR		
structure	# par.	ML	MCL	MM	ML	MCL	MM	ML	MCL	MM
NB	6720	3252	3289	3170	1784	1513	1520	83.73	92.00	91.97
TAN-CMI	38350	15772	25327	16790	8603	18647	9448	91.28	92.91	94.21
TAN-OMI-CR	44600	22488	29159	24048	13615	20419	15147	92.01	93.59	94.60
TAN-CR	39980	19557	25733	23308	11794	17702	16020	92.58	93.72	95.02

Figures 4, 5, and 6, respectively. Classification rates using full double-precision floating-point parameters are indicated by the dotted lines. The classification performance resulting from BNCs with reduced precision ML, MCL, and MM parameters are shown by the solid lines. Reduced precision parameters were determined by firstly learning parameters in double-precision, and secondly reducing the precision of these parameters. Even when using only 4 bits to represent the exponent and 1 bit to represent the mantissa, the classification rates are close to full-precision performance on USPS data. On MNIST and TIMIT data the results are similar when 4 and 2 bits are used to represent the mantissa, respectively.

Furthermore, we evaluated the classification performance of BNCs with reduced precision parameters using a varying size of the training set. The training sets were obtained by selecting the desired number of samples randomly from all available samples. The remaining samples were used as test set. For every sample size, 5 different training/test splits were evaluated. Results on USPS data are shown in Figure 7. Classification performance using reduced precision parameters is close to optimal for all sample sizes.



(a) NB

(b) TAN-CMI

(c) TAN-OMI-CR

(d) TAN-CR

Fig. 4. Classification rates of BNCs with (a) NB, (b) TAN-CMI, (c) TAN-OMI-CR, and (d) TAN-CR structures using reduced precision ML, MCL, and MM parameters on USPS data. The bit-width of the mantissa was fixed to 1 bit and the bit-width of the exponent was varied. The classification rates for full double-precision floating-point parameters are indicated by the horizontal dotted lines. Error bars indicate the 95 % confidence intervals of the mean classification rate over 5 different training/test splits.

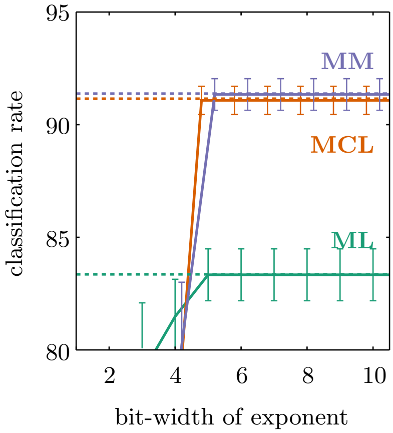
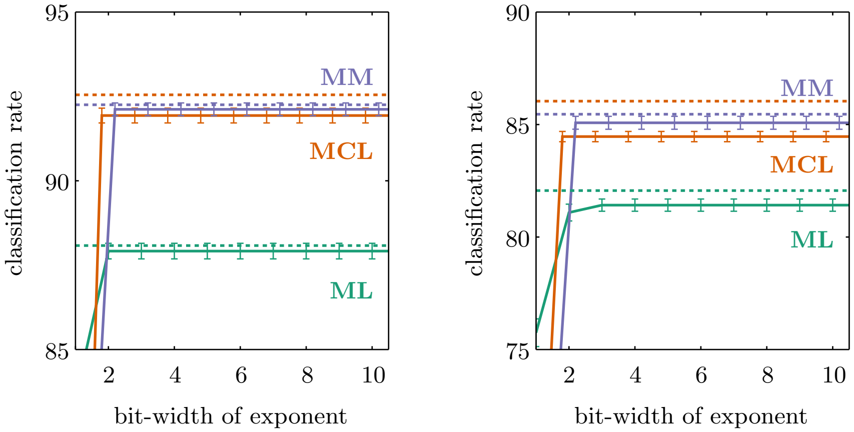


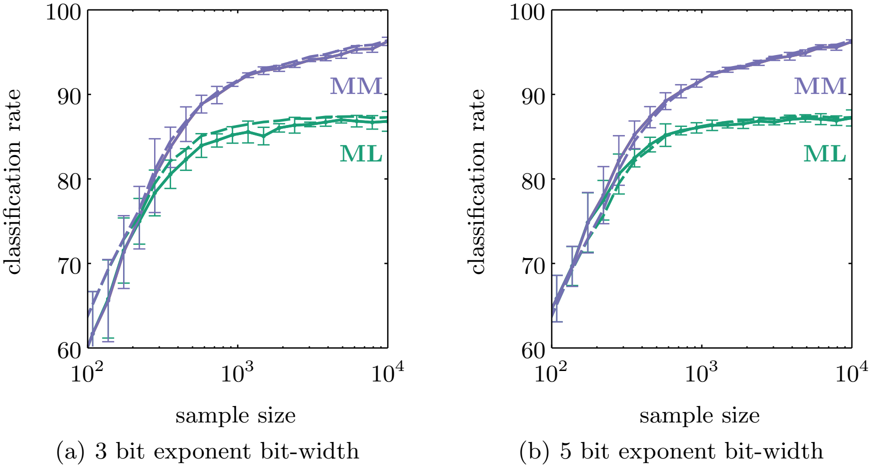
Fig. 5. Classification rates of BNCs with NB structure using reduced precision ML, MCL, and MM parameters on MNIST data. The bit-width of the mantissa was fixed to 4 bits and the bit-width of the exponent was varied. The classification rate for full double-precision floating-point parameters is indicated by the horizontal dotted lines. Error bars indicate the 95 % confidence intervals of the mean classification rate over 5 different training/test splits.



(a) 4 classes

(b) 6 classes

Fig. 6. Classification rates of BNCs with NB structure using ML, MCL, and MM parameters with reduced precision on TIMIT data with (a) 4 classes and (b) 6 classes. The bit-width of the mantissa was fixed to 2 bits and the bit-width of the exponent was varied. The classification rates for full double-precision floating-point parameters are indicated by the horizontal dotted lines. Error bars indicate the 95 % confidence intervals of the mean classification rate over 5 different training/test splits.



(a) 3 bit exponent bit-width

(b) 5 bit exponent bit-width

Fig. 7. Classification rates of BNCs with NB structures using reduced precision ML and MM parameters on USPS data. The parameters were learned from training sets with varying sizes. The bit-width of the mantissa was fixed to 1 bit. The bit-width of the exponent is 3 bits in (a) and 5 bits in (b). The classification rates for full double-precision floating-point parameters using the same training data are indicated by the dashed lines. Error bars indicate the 95 % confidence intervals of the mean classification rate over 5 different training/test splits.

6 Conclusion and Further Work

In this paper, we presented classification results of BNCs when reducing the precision of the probability parameters. Contrary to the authors' expectation, even discriminatively optimized BNCs are robust to distortions in the parameters resulting from the bit-width reduction. About 6 to 10 bits are necessary to represent each probability parameter while maintaining classification rates close to full-precision performance. This allows either to implement BNCs with reduced precision floating point arithmetic or to cast the classification to the integer domain. In both cases, computational and run-time benefits arise when implementing BNCs on embedded systems or low-power computers.

Future work aims to address the following issues:

1. Analytical determination of the minimum bit-width of the probability parameters of BNCs such that classification rates are close to full-precision performance. Results from sensitivity analysis are to be used. The analysis will be performed for different datasets and classifier structures.
2. Implementation of BNCs in the integer domain and measuring the computational complexity reduction.

Acknowledgments. This work was supported by the Austrian Science Fund (project numbers P22488-N23, S10608-N13 and S10610-N13).

References

1. Acid, S., Campos, L.M., Castellano, J.G.: Learning Bayesian network classifiers: Searching in a space of partially directed acyclic graphs. Machine Learning 59, 213–235 (2005)
2. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer (2007)
3. Chan, H., Darwiche, A.: When do numbers really matter? Artificial Intelligence Research 17(1), 265–287 (2002)
4. Chan, H., Darwiche, A.: Sensitivity analysis in Bayesian networks: From single to multiple parameters. In: Uncertainty in Artificial Intelligence (UAI), pp. 67–75 (2004)
5. Friedman, N., Geiger, D., Goldszmidt, M.: Bayesian network classifiers. Machine Learning, 131–163 (1997)
6. Guo, Y., Wilkinson, D., Schuurmans, D.: Maximum margin Bayesian networks. In: Uncertainty in Artificial Intelligence (UAI), pp. 233–242 (2005)
7. Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall, Upper Saddle River (1998)
8. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
9. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
10. Muller, J.M., Brisebarre, N., de Dinechin, F., Jeannerod, C.P., Lefèvre, V., Melquiond, G., Revol, N., Stehlé, D., Torres, S.: Handbook of Floating-Point Arithmetic. Birkhäuser Boston (2010)
11. Overton, M.L.: Numerical computing with IEEE floating point arithmetic - including one theorem, one rule of thumb, and one hundred and one exercices. Society for Industrial and Applied Mathematics (SIAM) (2001)
12. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
13. Pernkopf, F., Wohlmayr, M., Tschiatschek, S.: Maximum margin Bayesian network classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 34(3), 521–531 (2012)
14. Pernkopf, F., Bilmes, J.A.: Efficient heuristics for discriminative structure learning of Bayesian network classifiers. Journal of Machine Learning Research (JMLR) 11, 2323–2360 (2010)
15. Roos, T., Wettig, H., Grünwald, P., Myllymäki, P., Tirri, H.: On discriminative Bayesian network classifiers and logistic regression. Machine Learning 59(3), 267–296 (2005)
16. Vapnik, V.N.: Statistical Learning Theory. Wiley (1998)
17. Wang, H.: Using sensitivity analysis for selective parameter update in Bayesian network learning. In: Association for the Advancement of Artificial Intelligence, AAAI (2002)