Van-Nam Huynh
Masahiro Inuiguchi
Thierry Denoeux (Eds.)

Integrated Uncertainty in Knowledge Modelling and Decision Making

LNAI 9376

4th International Symposium, IUKM 2015
Nha Trang, Vietnam, October 15–17, 2015
Proceedings

Springer

Lecture Notes in Artificial Intelligence 9376

Subseries of Lecture Notes in Computer Science

More information about this series at http://www.springer.com/series/1244

Van-Nam Huynh · Masahiro Inuiguchi
Thierry Denoeux (Eds.)

Integrated Uncertainty in Knowledge Modelling and Decision Making

4th International Symposium, IUKM 2015
Nha Trang, Vietnam, October 15–17, 2015
Proceedings

Springer

Editors
Van-Nam Huynh
Japan Advanced Institute of Science and Technology
Nomi
Japan

Thierry Denoeux
Université de Technologie de Compiègne
Compiègne
France

Masahiro Inuiguchi
Graduate School of Engineering Science
Osaka
Japan

ISSN 0302-9743 ISSN 1611-3349 (electronic)
Lecture Notes in Artificial Intelligence
ISBN 978-3-319-25134-9 ISBN 978-3-319-25135-6 (eBook)
DOI 10.1007/978-3-319-25135-6

Library of Congress Control Number: 2015951823

LNCS Sublibrary: SL7 – Artificial Intelligence

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Minimum Description Length Principle for Compositional Model Learning

Radim Jiroušek[1,2(✉)] and Iva Krejčová[1]

[1] Faculty of Management, University of Economics, Jindřichův Hradec, Czech Republic
radim@utia.cas.cz, iva.krejcova@gmail.com
[2] Institute of Information Theory and Automation, Prague Czech Academy of Sciences, Prague, Czech Republic

Abstract. Information-theoretic viewpoint at the data-based model construction is anchored on the assumption that both source data and a constructed model comprises certain information. Not having another source of information than source data, the process of model construction can be viewed at as the transformation of information representation. The combination of this basic idea with the Minimum Description Length principle brings a new restriction on the process of model learning: avoid models containing more information than source data, because these models must comprise an additional undesirable information. In the paper, the idea is explained and illustrated on the data-based construction of multidimensional probabilistic compositional models.

Keywords: Machine learning · Multidimensional models · Probability distributions · Composition · Information theory · Lossless encoding

1 Introduction

Minimum Description Length (MDL) principle has been used for model learning by a whole range of authors. In connection with Bayesian network learning let us mention for example Lam and Bacchus [10], (for general sources see also [2], and [3]). These authors regarded MDL as an application of a Occam's razor philosophical principle, which says that the best solution is more likely the simplest. In this paper we will study this approach also from another point of view, from the point of view of information theory.

Data-based model learning is usually based on the following simple idea: the data in question were generated by a generator whose probabilistic characteristics are unknown, but, in a way, stabile. If we do not have another source of information (such as, for example, some theoretical knowledge about the field of interest) all we know about this generator is encoded in the data file. So, when reconstructing the generator we should exploit as much of information contained in the data file as possible, but we should avoid adding any other undesirable information. In this sense, the process of model construction can be

V.-N. Huynh et al. (Eds.): IUKM 2015, LNAI 9376, pp. 254–266, 2015.
DOI: 10.1007/978-3-319-25135-6_25

viewed as a transformation of the data file into the constructed model. Since it is well-known that during any transformation process the amount of information cannot increase, we should check what is the amount of information before the transformation (i.e., the information contained in the input data file) and after the transformation (i.e., the information contained in the constructed model). Using this idea, models containing more information than the input data file will be considered unacceptable because, obviously, some undesirable information was added.

Accepting the above mentioned principle, a new problem arises: how to measure the above mentioned information amounts. Our proposal is to measure this information in bits necessary for the optimum lossless encoding of the data/model. In this context we take advantage of the old ideas of von Mises [13] and Kolmogorov [8] who both explored relations interconnecting randomness, complexity and information. So, we accept the principle: the more complex model, the more information it comprises. Nevertheless, realize that looking for the optimum lossless encoding would be in practical situations intractable. Therefore, we use in this paper some heuristics and also a famous Huffman's encoding [4], which is known to be optimal under some conditions. Thus, though the encoding used in this paper is only suboptimal, it will serve well to the purpose of this paper: to show that application of MDL principle is not as straightforward as it can seem at the first glance. We will show that the users should find a reasonable equilibrium balancing the complexity of the model structure and the preciseness of specified parameters.

The proposed approach is fully sensible also from the statistical point of view. The less data we have, the less amount of bits we may use to encode the model. It means, among others, that for small data files we cannot consider probability values specified with a high precision. This fully corresponds with the fact that having a small number of data, the confidence intervals for the estimates of probability parameters are rather wide. Therefore it does not have a sense to specify these estimates with a high precision, with a great number of digits.

Thus, the goal of this paper is not to introduce a new algorithm for data-based model learning. The paper presents two simple ideas that should be incorporated into any data-based learning algorithm and that we have not found in the literature. First, the amount of input data determines the upper limit to the complexity of the constructed model. It is against a common sense (and also against the information-theoretic principles) to construct a model whose encoding requires more bits than the input data file. The other idea is that the users should decide whether it is more advantageous to consider either simpler structure models with more precise parameters, or models with more complex structures, i.e., more parameters specified with lower precision.

The application of the above mentioned ideas will be illustrated on learning compositional models [6] that will be briefly introduced in the next section. Sections 3 and 4 will be devoted to the discussion of possibilities how to encode data and models, respectively, and Section 5 briefly describes two ways how to

simplify constructed models to meet the upper limit given by the size of the input data.

2 Compositional Models

As said above, in this section, we will briefly introduce the models to be constructed from data; for more details and the properties of these models the reader is refereed to [6].

In the whole paper we consider a finite set of finite-valued random variables $N = \{X_1, X_2, \ldots, X_n\}$. Probability distributions (measures) will be denoted by the characters of Greek alphabet, as e.g., $\pi(N)$. Its marginal distribution for variables $M \subseteq N$ will be denoted either $\pi(M)$, or $\pi^{\downarrow M}$. Let $\mathbb{X}_i$ denote the set of values of variable X_i. It means that a probability distribution $\pi(N) : \mathbb{X}_N \to [0, 1]$ is defined with the help $|\mathbb{X}_N|$ numbers (probabilities), where $\mathbb{X}_N$ denotes the Cartesian product $\mathbb{X}_N = \mathbb{X}_1 \times \mathbb{X}_2 \times \ldots \times \mathbb{X}_n$, e.g., the space of all states of variables N. Analogously, for a subset of variables $K \subset N$, $\mathbb{X}_K = \times_{u \in K} \mathbb{X}_u$.

For two distributions $\pi(N)$ and $\kappa(N)$, we say that κ dominates π (in symbol $\pi \ll \kappa$) if for all $x \in \mathbb{X}_N$, for which $\kappa(x) = 0$ also $\pi(x) = 0$. As a measure of similarity of two distributions we will consider their *Kullback-Leibler divergence* [9] (or crossentropy) defined

$$Div(\pi; \kappa) = \begin{cases} \sum\limits_{x \in \mathbb{X}_N : \pi(x) > 0} \pi(x) \log \frac{\pi(x)}{\kappa(x)} & \text{if } \pi \ll \kappa, \\ +\infty & \text{otherwise,} \end{cases}$$

which is known to be zero if and only if $\pi = \kappa$.

Compositional models considered in this paper are multidimensional probability distributions that are assembled (*composed*) from its low-dimensional marginals with the help of a so called *operator of composition*. This operator realizes an operation in a way inverse to marginalization. For a probability distribution $\mu(N)$ and $J, K \subset N$, such that $J \cup K = N$, the respective marginal distribution $\mu^{\downarrow J}$ and $\mu^{\downarrow K}$ are unique. On the other side, if $J \neq N$ and $K \neq N$ then there are (infinitely) many distributions $\nu(N)$ such that $\nu^{\downarrow J} = \mu^{\downarrow J}$ and $\nu^{\downarrow K} = \mu^{\downarrow K}$. All these distributions ν are called *join extensions of* $\mu^{\downarrow J}$ *and* $\mu^{\downarrow K}$. One of them can be got by the application of the following operator of composition.

Definition 1. *For two arbitrary distributions* $\pi(M)$ *and* $\lambda(L)$*, for which* $\pi^{\downarrow M \cap L} \ll \lambda^{\downarrow M \cap L}$*, their* composition *is, for each* $x \in \mathbb{X}_{L \cup M}$*, given by the following formula*[1]

$$(\pi \triangleright \lambda)(x) = \frac{\pi(x^{\downarrow M})\lambda(x^{\downarrow L})}{\lambda^{\downarrow M \cap L}(x^{\downarrow M \cap L})}.$$

In case $\pi^{\downarrow M \cap L} \not\ll \lambda^{\downarrow M \cap L}$*, the composition remains undefined.*

[1] In this paper we take $\frac{0.0}{0} = 0$ by definition.

Notice that the composition $\mu^{\downarrow J} \triangleright \mu^{\downarrow K}$ is always defined (because the marginals $\mu^{\downarrow J}$ and $\mu^{\downarrow K}$ are consistent), and that the distribution $\mu^{\downarrow J} \triangleright \mu^{\downarrow K}$ need not coincide with $\mu(N)$. It is easy to show (see [6]) that for the composed distribution $\mu^{\downarrow J} \triangleright \mu^{\downarrow K}$ variables $J \setminus K$ and $K \setminus J$ are conditionally independent[2] given variables $K \cap J$.

If we compose two general distributions $\pi(M)$ and $\lambda(L)$, and the composition $\pi \triangleright \lambda$ is defined, then the result is a distribution of variables $M \cup L$, and it is an extension of distribution π (see [6]), which is as similar as possible to a given distribution λ in the following sense (see Theorem 6.2 in[6])

$$\pi \triangleright \lambda = \arg \min_{\kappa(L \cup M):\kappa^{\downarrow M} = \pi} Div(\kappa^{\downarrow L}; \lambda).$$

Notice that if $\pi \triangleright \lambda$ is defined, then this minimum is unique. This is also the reason why we can say that $\pi \triangleright \lambda$ is a *projection* of λ into the set (space) of all the extensions of π for variables $L \cup M$ [1].

In this paper we are not interested in computational properties of distributions represented in a form of (iterative) compositions, so we need not present the algebraic properties of the operator of composition; for them, the reader is referred to [6]. Instead, let us present the definition of a compositional model.

Definition 2. *Distribution* $\kappa(N)$ *is a* compositional model *if there exists a cover* $K_1, K_2, \ldots, K_m$ *(i.e.,* $K_1 \cup \ldots \cup K_m = N$*), such that*[3]

$$\kappa(N) = \kappa^{\downarrow K_1} \triangleright \kappa^{\downarrow K_2} \triangleright \ldots \triangleright \kappa^{\downarrow K_m}. \tag{1}$$

Let us conclude this section by stating that the class of compositional models is exactly the same as the class of Bayesian networks [5].

3 Coding Data

The goal of this and the next section is not to find algorithms encoding compositional models and/or data files but just to estimate how many bits are necessary for such encodings. These numbers will be used to measure complexity of the respective models (data). More precisely, these numbers will be used when we will compare the complexity of two models, or the complexity of a model and

[2] Recall that for distribution $\kappa(N)$ variables K and L are conditionally independent given variables M ($K, L, M \subseteq N$ are assumed to be disjoint) if

$$\kappa(K \cup L \cup M) \cdot \kappa(M) = \kappa(K \cup M) \cdot \kappa(L \cup M).$$

[3] Since the operator of composition is not associative, we have to say how to understand the expression (1): If not specified otherwise by parentheses, the operator is always performed from left to right, i.e.,

$$\kappa^{\downarrow K_1} \triangleright \kappa^{\downarrow K_2} \triangleright \ldots \triangleright \kappa^{\downarrow K_m} = \Big(\ldots \big((\kappa^{\downarrow K_1} \triangleright \kappa^{\downarrow K_2}) \triangleright \kappa^{\downarrow K_3} \big) \triangleright \ldots \triangleright \kappa^{\downarrow K_{m-1}} \Big) \triangleright \kappa^{\downarrow K_m}.$$

the complexity of data. This is why we will not consider coding the number of variables, variable names and the cardinality of their value sets. Coding this information would just increase all the derived complexity measures by a constant. Therefore, without loss of generality we can assume in this paper that variable X_i is identified by its index i, and their values are $\mathbb{X}_i = \{0, \ldots, h_{i-1}\}$.

Under the above assumption when encoding a data file $\mathbf{D}$ we have to encode a matrix of nonnegative integers with d rows (records of the data file) and n columns (variables). For this we will consider several simple procedures. Let us repeat once more that we are aware of the fact that using more sophisticated types of codes, such as e.g. arithmetic codes [14], we could achieve even more economic encoding. The following codes are selected as a trade-off between precision and simplicity of the following exposition.

Direct Encoding. For a binary variable we need just one bit for each entry of the matrix. If the respective $h_i > 2$ then we use[4] $\ell_i = \lceil \log_2 h_i \rceil$ bits to encode the value of variable X_i. Therefore, for the direct encoding of the data file we need

$$c_d(\mathbf{D}) = d \times (\ell_1 + \ell_2 + \ldots + \ell_n) + c$$

bits, where c denotes the number of bits necessary to encode the number of records d (the number of rows in the matrix).

Frequency Encoding. For this coding we will take advantage of the fact that we need not consider the ordering of records in the data file. We increase the data matrix by one column into which we insert the number of repetition of each state (by *state* we understand the combination of values of all variables) in the data file. It enables us to keep in the matrix each state only once. Thus, denoting d_{red} the number of different states appearing in the original data file, and denoting f_{max} the maximal number of occurrences of the same state in the data file, then for this type of encoding we need

$$c_f(\mathbf{D}) = d_{red} \times (\ell_1 + \ell_2 + \ldots + \ell_n + \lceil \log_2(f_{max} - 1) \rceil) + 2 \times c$$

bits. $\lceil f_{max} - 1 \rceil$ appears in the formula, because all the numbers of repetition in the $(n+1)$th column are numbers from $1, \ldots, f_{max}$, and thus we can encode them as numbers from $0, \ldots, f_{max} - 1$, and $2 \times c$ bits are necessary to encode d_{red} a f_{max}.

Huffman Frequency Encoding. By this term we understand coding of the same table like in the previous case but for coding the numbers of occurrence we use the famous Huffman code [4]. The number of necessary bits for this code will be denoted by $c_{fH}(\mathbf{D})$ (see an example below).

[4] $\lceil r \rceil$ denotes the smallest integer, which is not less than r.

Lexicographic Encoding. Analogously to preceding type of encoding, consider an extended data matrix in which each state appears maximally once, and the $(n+1)$th column contains the number expressing how many times the state appears in the original data file. If the number of variables is rather small, it may happen that the following encoding of the considered matrix is more economic than that by frequency encoding: add to the matrix all states that do not appear in data (with number of repetition equaling 0), sort all the states in the lexicographic order, and then we can encode only the numbers from the $(n+1)$th column. This coding requires

$$c_l(\mathbf{D}) = |\mathbb{X}_N| \times \lceil \log_2 f_{max} \rceil + c$$

bits (realize the last c bits are used to encode f_{max}).

Huffman Lexicographic Encoding. As in the previous case we code only frequencies for all $|\mathbb{X}_N|$ combinations, for which we use the Huffman encoding. For real data files, Huffman process usually yields a code with the average length less that two bits per number (this is because in practical situations numbers of repetition greater than 1 are rare).

Naturally, the readers can extend the list of the considered data encoding possibilities by as many other approaches as they want (e.g. see [12]). In this paper we consider the complexity measure for the data file just

$$c(\mathbf{D}) = \min\{c_d(\mathbf{D}), c_f(\mathbf{D}), c_{fH}(\mathbf{D}, c_l(\mathbf{D}), c_{lH}(\mathbf{D})\}.$$

Example. The ideas presented in this paper will be illustrated by an example with artificially generated data. For the sake of simplicity we consider in this example just eight binary variables (with values $0, 1$), and a data files with 100 records (binary vectors). In spite of this we fix the number of necessary bits to encode the length of the data file to $c = 32$, because we made experiments with much bigger data files (up to 100 000 records). Recall that we neglect coding the information about the model.

To apply the direct encoding approach, when taking into account the considered small data file we need to encode the following table

$$d = 100 \left\{ \begin{array}{cccccccc} 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ & & \vdots & & & & & \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right.$$

which means that we need $c_d(\mathbf{D}) = 100 \times 8 + 32 = 832$ bits.

To encode the same data file with the frequency encoding, transform first the data file into the form, in which all the rows (states) are unique and the last column contains the number of occurrences of the respective state in the original

data file. For the considered data file we get the following table

$$d_{red} = 38 \left\{ \begin{array}{cccccccccc} 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & & 27 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & & 10 \\ & & \vdots & & & & & & & \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & & 1 \end{array} \right.$$

Thus we get $c_f(\mathbf{D}) = 38 \times (8+5) + 2 \times 32 = 558$ bits.

To get what we call Huffman version of frequency encoding we need to find Huffman code for the numbers of occurrences. In our case such a code is the following (the numbers in parentheses - the last column - read how many times the respective frequency number appears in the above table)

27	11111	(1×)
10	11110	(1×)
5	1110	(2×)
3	110	(5×)
2	10	(9×)
1	0	(20×)

Thus, using Huffman version of frequency encoding we have to encode the above coding table (which can easily be done with $6 \times (5+5) = 60$ bits, and for coding the numbers of occurrences we need only $2 \times 5 + 2 \times 4 + 5 \times 3 + 9 \times 2 + 20 \times 1 = 71$ bits (instead of $38 \times 5 = 190$, which is needed for the frequencies encoding in the previous case). So, we get $c_{fH}(\mathbf{D}) = 38 \times (8) + 60 + 71 + 2 \times 32 = 499$ bits.

To get the lexicographic encoding we have to consider all 2^8 states lexicographically ordered

$$256 \left\{ \begin{array}{cccccccc|c} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ & & \vdots & & & & & & \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 27 \\ & & \vdots & & & & & & \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{array} \right.$$

So, lexicographic encoding of the framed frequencies requires $c_l(\mathbf{D}) = 256 \times 5 + 32 = 1312$ bits. However, if we use Huffman approach to encode all the numbers appearing in the frame, i.e., if we use the following code

27	111111	(1×)
10	111110	(1×)
5	11110	(2×)
3	1110	(5×)
2	110	(9×)
1	10	(20×)
0	0	(218×)

Table 1. Requirements for coding the data files

	c_d	c_f	c_{fH}	c_l	c_{lH}
$\mathbf{D_{100}}$	832	558	499	1,312	**404**
$\mathbf{D_{1000}}$	8,032	1,680	**976**	2.080	992
$\mathbf{D_{10000}}$	80,032	4,084	2,713	3,104	**2,362**
$\mathbf{D_{100000}}$	800,032	5,676	5,800	**3,872**	4,775

we need only $7 \times (5 + 6) = 77$ bits to encode this coding table, and $c_{lH}(\mathbf{D}) = 77 + 2 \times 6 + 2 \times 5 + 5 \times 4 + 9 \times 3 + 20 \times 2 + 218 \times 1 = 404$ bits.

To illustrate the way how these complexity measures increase with the amount of the considered data we generated (using the same generator) another three data files with 1 000, 10 000 and 100 000 records. A summary of the bit requirements to encode all these data files is in Table 1.

4 Coding Models

To encode a compositional model given by Formula (1) we have to encode marginal distributions $\kappa^{\downarrow K_1}, \kappa^{\downarrow K_2}, \ldots, \kappa^{\downarrow K_m}$ in a proper order. Each of these distributions $\kappa^{\downarrow K_i}$ is described by the list of variables, i.e.,

number of variables $\lvert K_i \rvert$	$\lceil \log_2 n \rceil$ bits
list of variables	$\lvert K_i \rvert \times \lceil \log_2 n \rceil$ bits

and the respective probabilities, whose total number is $\prod_{u \in K_i} h_u$. Obviously, encoding the probabilities is, as a rule, much more space demanding that encoding the variables, for which the respective marginal is defined. The latter encoding requires, as presented above, only $(\lvert K_i \rvert + 1) \times \lceil \log_2 n \rceil$ bits.

Naturally, the space requirements for the probability encoding is closely connected with the precision with which the respective probabilities should be specified. A simple way, which is used in this paper, is the following.

Select a positive integer, denote it *base*, and express all the considered probabilities as a ratio of two nonnegative integers

$$\frac{a}{base}.$$

This means that the respective probability will be encoded by integer a. From the obvious reasons it does not have a sense to choose $base > d$ (recall that d is the number of records in the input data file). However, *base* may be much smaller than d and can be defined with respect to the size of confidence intervals computed for the probability estimates, or it can be reduced when we want to reduce the complexity of the constructed compositional model (as shown in the next section).

By employing the idea of representing probabilities by integers we get, in fact, exactly the same situation as that in the previous section: marginal distribution

$\kappa^{\downarrow K_i}$ is fully described by those states $x \in \mathbb{X}_{K_i}$, for which the probability $\kappa^{\downarrow K_i}(x)$ is positive and by the respective integer representing value $\kappa^{\downarrow K_i}(x)$. It means that for encoding the marginal distributions $\kappa^{\downarrow K_i}$ we can employ any of the techniques described in the previous section (perhaps, application of the *direct encoding* comes into consideration in very specific and unusual situations, though). As a rule, the most economic encoding is yielded by *Huffman lexicographic encoding*. *Frequency encoding* (both plain and Huffman's) may be applicable only for more-dimensional distributions, which are positive on a small part of the respective space $\mathbb{X}_{K_i}$.

Thus, when encoding compositional models we will face the only problem: whether it is more economic to construct a Huffman code specially for each marginal distribution (and thus also code the respective coding table), or construct one code for coding all the marginals from which the model is composed.

An analogous problem is connected with the selection of the number *base*. In this paper we consider only simple models and therefore we use one number *base* for the whole model. However, the reader certainly realizes that in some situations a greater chances to decrease the complexity of the model can be achieved when defining different $base_i$ for different marginals. Namely, the necessity of coding one number $base_i$ for each marginal distribution can be payed back by the savings achieved for coding the respective probabilities.

Example Continued. Let us illustrate the principles described above by coding a model

$$\mathbf{M_1}: \ \mu_1 = \kappa^{\downarrow\{1,2\}} \triangleright \kappa^{\downarrow\{3,4\}} \triangleright \kappa^{\downarrow\{3,5\}} \triangleright \kappa^{\downarrow\{1,4,5,6\}} \triangleright \kappa^{\downarrow\{5,6,8\}} \triangleright \kappa^{\downarrow\{2,5,6,7,8\}} \tag{2}$$

constructed from the considered data file **D** with 100 records. To describe a structure of the model we need to specify the number of marginal distributions $m = 6$, number $base = 100$.

Thus, the structure of the model (2) can be described with the help of $\lceil \log_2 n \rceil + c = 3 + 32 = 35$ bits, and to encode k-dimensional distribution by lexicographic encoding we need either:

number of variables k	$\lceil \log_2 n \rceil$ bits,
list of variables	$k \times \lceil \log_2 n \rceil$ bits,
frequencies (probabilities)	$2^k \times \lceil \log_2 base \rceil$ bits,

or, in the case that specification of the maximal frequencies for each marginal $f_{max,i}$ pays back by savings gained for more economic specification of all frequencies,

number of variables k	$\lceil \log_2 n \rceil$ bits,
list of variables	$k \times \lceil \log_2 n \rceil$ bits,
maximal frequency $f_{max,i}$	$\lceil \log_2 base \rceil$ bits,
frequencies (probabilities)	$2^k \times \lceil \log_2 f_{max,i} \rceil$ bits.

In our case the two approaches differ just by 18 bits, so let us consider the simpler (the former) approach. Thus we need

$$\begin{aligned}
&\text{for } \kappa^{\downarrow\{1,2\}}, \kappa^{\downarrow\{3,4\}}, \kappa^{\downarrow\{3,5\}}\text{: } 3 + 6 + 28 = 37 \text{ bits},\\
&\text{for } \kappa^{\downarrow\{1,4,5,6\}}\text{: } 3 + 12 + 112 = 127 \text{ bits},\\
&\text{for } \kappa^{\downarrow\{5,6,8\}}\text{: } 3 + 9 + 56 = 68 \text{ bits},\\
&\text{for } \kappa^{\downarrow\{2,7,5,6,8\}}\text{: } 3 + 15 + 224 = 242 \text{ bits},
\end{aligned}$$

which means that $c_l(\mathbf{M_1}) = 583$.

Taking into account the fact that among the 68 frequencies (probabilities) needed to represent the respective six marginals there appears twenty times "0" and sixteen times "1", it is not surprising that a more economic encoding is achieved by Huffman's version of lexicographic encoding, which yields for this model $c_{lH}(\mathbf{M_1}) = 423$. In any case, whatever type of encoding we may take into consideration we cannot reach the coding requirements sufficient to encode data $c_{lH}(\mathbf{D}) = 404$. This means that the model $\mathbf{M_1}$ described by formula (2) with probabilities specified with the help of $base = 100$ is unacceptable, and therefore, to meet the information-theoretic viewpoint at MDL principle described in Introduction, we have to simplify the considered model by any of the possibilities described in the next section.

5 Model Simplification

Perhaps the easiest way how to simplify the constructed model is to roughen the probability estimates by decreasing the constant *base*. Considering model $\mathbf{M_1}$ with $base = 100$ means that we take all the probability estimates with two digits of precision. Rounding these estimates to one decimal digit means to consider $base = 10$. Nevertheless, it is important to realize that we can consider finer roughening choosing any $10 < base < 100$. Denote $c_{lH}(\mathbf{M_{1:50}})$, $c_{lH}(\mathbf{M_{1:40}})$ and $c_{lH}(\mathbf{M_{1:32}})$ complexity of Huffman lexicographic encoding of model $\mathbf{M_1}$ with *base* equaling 50, 40 and 32, respectively. Then for the probability estimates got from data file $\mathbf{D}$ we have $c_{lH}(\mathbf{M_{1:50}}) = 408$, $c_{lH}(\mathbf{M_{1:40}}) = 397$, and $c_{lH}(\mathbf{M_{1:32}}) = 284$. Thus, both the latter two models are acceptable from the information-theoretic viewpoint at MDL principle. Let us also note that a greater simplification achieved when changing *base* from 40 to 32 than when changing *base* from 50 to 40 is due to the fact that $\lceil \log_2 40 \rceil > \lceil \log_2 32 \rceil$ and $\lceil \log_2 50 \rceil = \lceil \log_2 40 \rceil$.

Another way how to simplify the considered model is to simplify its structure. Obviously, in the sense of space requirements the most costly is the five-dimensional marginal $\kappa^{\downarrow\{2,5,6,7,8\}}$. Let us consider two simplifications of $\mathbf{M_1}$ consisting only of two- and three-dimensional marginals:

$$\mathbf{M_2}: \ \mu_2 = \kappa^{\downarrow\{1,2\}} \triangleright \kappa^{\downarrow\{3,4\}} \triangleright \kappa^{\downarrow\{3,8\}} \triangleright \kappa^{\downarrow\{5,8\}} \triangleright \kappa^{\downarrow\{2,7,8\}} \triangleright \kappa^{\downarrow\{1,5,6\}}, \tag{3}$$

and

$$\mathbf{M_3}: \ \mu_3 = \kappa^{\downarrow\{3,4\}} \triangleright \kappa^{\downarrow\{3,5\}} \triangleright \kappa^{\downarrow\{1,5,6\}} \triangleright \kappa^{\downarrow\{5,6,8\}} \triangleright \kappa^{\downarrow\{6,7,8\}} \triangleright \kappa^{\downarrow\{1,2,7\}}. \tag{4}$$

Repeating computations described in the preceding section we get $c_l(\mathbf{M_2}) = 306$ and $c_l(\mathbf{M_3}) = 356$ bits, and $c_{lH}(\mathbf{M_2}) = 267$ and $c_{lH}(\mathbf{M_3}) = 304$ bits. Let us

Table 2. Kullback-Leibler divergences

	$\mathbf{M_1}$	$\mathbf{M_{1:50}}$	$\mathbf{M_{1:40}}$	$\mathbf{M_{1:32}}$	$\mathbf{M_2}$	$\mathbf{M_3}$
complexity	423	408	397	284	267	304
K-L divergence	0.2736	0.2795	0.2846	0.2881	0.2964	0.3036

stress that these complexities are computed for models with with $base = 100$. So, comparing these values with $c_{lH}(\mathbf{D}) = 404$ we see that both these models are from our point of view acceptable.

Nevertheless, it is clear that we cannot evaluate models just on the basis of MDL principle, just according to the number of bits necessary for their encoding. We also need a criterion evaluating to what extent each model carries the information contained in the considered data. For this, we use the Kullback-Leibler divergence between the sample probability distribution defined by data and the probability distribution defined by the model. So, for each considered model we can compute the Kullback-Leibler divergence between the eight-dimensional sample distribution κ defined by the considered data file with 100 records, and the distribution defined by the respective model. For example, for model $\mathbf{M_1}$ it is $Div(\kappa; \mu_1)$, where κ is the sample distribution, and μ_1 is the distribution defined from κ by Formula (2). The values of these divergences for all the considered models are in Table 2.

From Table 2 we can see that the simplification of a model by decreasing the value of constant *base*, i.e., by roughening the estimates of probabilities, leads to the decrease of complexity of the model and simultaneous increase of the Kullback-Leibler divergence. The greater this type of simplification, the greater the respective Kullback-Leibler divergence. A precise version of this statement can be expressed in a form of mathematical theorems whose presentation is beyond the scope of this paper. On the other hand, from the last two columns of Table 2 the reader can see that a similar relation valid for the simplification of a model by decreasing the complexity of a model would be much more complex. This is based on the fact that though both models $\mathbf{M_2}$ and $\mathbf{M_3}$ are the simplification of $\mathbf{M_1}$, no one is a simplification of the other. This means that for structure simplification the strength of simplification cannot be measured just by one parameter, by the amounts of bits necessary for the model encoding but we have to introduce also some partial order on the set of all potential simplifications, which is a topic for future research.

6 Conclusions

The novelty of this paper lies in the detailed analysis of the complexity of probabilistic models. We do not take into account only the structure of a model but also the precision of probabilities describing the model in question. It means that the final selection of the model is based on a trade-off between the complexity of model structure and the precision of probability estimates; the simplification of a

model structure makes it possible to consider more precise probability estimates and vice versa. On the other side it also means that employing these ideas into the process of model construction substantially increases the space of possible solutions in comparison with the approaches when only the structure is optimized. Fortunately, a rather great part of the models are "forbidden" because their complexity is greater than the upper limit determined by the input data. It is a topic for the future research to design tractable algorithms taking advantage of this property.

In this paper, the new ideas are illustrated on the data based construction of probabilistic multidimensional compositional models. Naturally, it can be applied also to the construction of other probabilistic multidimensional models (like e.g., Bayesian networks), and also to construction of models in other uncertainty theories (see e.g., Shenoy's valuation based systems [7]).

Acknowledgments. This research was partially supported by GAČR under Grant No. 15-00215S.

References

1. Csiszár, I.: I-divergence Geometry of Probability Distributions and Minimization problems. Ann. probab. **3**, 146–158 (1975)
2. Grünwald, P.: A Tutorial Introduction to the Minimum Description Length Principle, p. 80 (2004). [cit. 2014–07-15] http://eprints.pascal-network.org/archive/00000164/01/mdlintro.pdf
3. Hansen, M.H., Bin, Y.U.: Minimum Description Length Model Selection Criteria for Generalized Linear Models, p. 20. http://www.stat.ucla.edu/cocteau/papers/pdf/glmdl.pdf
4. Huffman, D.A.: A Method for the Construction of Minimum-Redundancy Codes. Proceedings of the I.R.E., 1098–1102 (1952)
5. Jensen, F.V.: Bayesian Networks and Decision Graphs. IEEE Computer Society Press, New York (2001)
6. Jiroušek, R.: Foundations of compositional model theory. Int. J. General Systems **40**(6), 623–678 (2011)
7. Jiroušek, R., Shenoy, P.P.: Compositional models in valuation-based systems. Int. J. Approx. Reasoning **53**(8), 1155–1167 (2012)
8. Kolmogorov, A.N.: Tri podchoda k opredeleniju ponjatija 'kolichestvo informacii'. Problemy Peredachi Informacii **1**, 3–11 (1965)
9. Kullback, S., Leibler, R.A.: On information and sufficiency. Annals of Mathematical Statistics **22**, 76–86 (1951)
10. Lam, W., Bacchus, F.: Learning Bayesian Belief Networks: An approach based on the MDL Principle. Computational Intelligence **10**, 269–293 (1994). http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.127.5504&rep=rep1&type=pdf

11. Lauritzen, S.L.: Graphical models. Oxford University Press (1996)
12. Mahdi, O.A., Mohammed, M.A., Mohamed, A.J.: Implementing a Novel Approach an Convert Audio Compression to Text Coding via Hybrid Technique. Int. J. Computer Science **3**(6), 53–9 (2012)
13. Von Mises, R.: Probability, statistics, and truth. Courier Corporation, Mineola (1957). [Originaly published in German by Springer, 1928]
14. Witten, I.H., Neal, R.M., Cleary, J.G.: Arithmetic Coding for Data Compression. Communications of the ACM **30**(6), 520–540 (1987)