

# Chapter 4: Nonlinear signal modelling and structure selection with applications to genomics

Ioan Tabus, Jorma Rissanen, Jaakko Astola

Institute of Signal Processing  
Tampere University of Technology  
P.O Box 553, FIN-33101 Tampere, Finland

September 26, 2005

to be published in *ADVANCES ON NONLINEAR SIGNAL AND IMAGE PROCESSING*,  
Edited by Steve Marshal and Giovanni Sicuranza

## 1 Introduction

Modeling is a prerequisite for the most fundamental signal processing tasks of signal analysis, detection, classification, denoising, and compression. While linear models are widely used, and they allow elegant theoretical and algorithmic developments, their nonlinear alternatives offer advantages in terms of modeling power and improved performance, which often eclipse the extra cost due to increased complexity. In this chapter we discuss nonlinear signal modeling with two main goals. The first is to present several examples of nonlinear models arising naturally in processing genomic data. The second is to discuss methods for evaluating the complexity of these nonlinear models with information theoretic methods.

Several signal processing methods are useful for processing genomic data. Microarray data come originally in the form of microarray images, which need to be preprocessed (denoised and segmented) before the quantities of interest, the gene expression ratios, are estimated from the image. All these stages benefit greatly from nonlinear techniques. The stages involving statistical inference on the gene expression values, can also be formulated as optimal design problems for the structure and the parameters of certain nonlinear predictors. In this chapter we investigate

mainly the gene prediction problem, and we propose nonlinear predictors in which the structure is selected by the minimum description length (MDL) principle.

We present first the Boolean, ternary, and perceptron predictors. We resort to the earliest form of the MDL principle, the two part code, as a tool for selecting the proper size of the prediction window, because the selection is done on an immediate and intuitive description of the data and the model parameters. When comparing predictors of different complexities we show that the best description is achieved by the Boolean and the ternary predictors, for they give a better fit to the data with a lower model complexity. To illustrate the results both synthetic and experimental data are used.

We then introduce a more advanced way to evaluate the overall cost of describing the data by Boolean regression models. These models are useful tools for various applications in nonlinear filtering, nonlinear prediction, classification and clustering, and data compression. We discuss the normalized maximum likelihood (NML) universal model for these classes of models. Examples of the problem of the discrimination of cancer types with the universal NML model for Boolean regression demonstrate the ability of the NML model to select sets of the feature genes that are capable of discrimination at error rates significantly smaller than what achievable with other discrimination methods.

## **2 Preliminaries: modeling and predicting gene expressions**

### **2.1 An introduction to gene expression data**

One of the most important experimental advances in biology in the last decade is the development of high throughput measurement methods for simultaneously probing thousands of gene expressions in the biological probes of interest. This permits probing deep into the functioning of the cell since at any moment the fundamental processes of life deal with the production of proteins, based directly on the information contained in the genes. The abundance (also called expression) of a certain gene in a biological sample is an indication of how intense the production of the corresponding protein or proteins is (in some cases the same gene can be used to generate several different proteins). However, the mere abundance of a certain gene in the cell at a certain time is not a guarantee that the corresponding proteins are produced, because the translation from gene to protein is mediated by a number of factors, like the availability of enabling enzymes in the cell, or the absence of certain inhibiting enzymes. The enzymes involved

are proteins, produced on the information existing in other genes. This (overly) simplified image of the complex metabolic processes shows that the functioning of the cell at a certain time is related to a large extent to the amount or abundance of a number of genes that are connected functionally to enable biological processes.

The concept of the gene expression was introduced four decades ago with the discovery of messenger RNA when the theory of genetic regulation of protein synthesis was described [5]. The availability of cDNA microarrays makes it possible to measure simultaneously the expressions level for thousands of genes. Gene expression data obtained in microarray experiments may often be discretized as binary or ternary data, the values 1,0,-1 carrying the meaning of over expressed, normal, and repressed, respectively, which are the descriptors needed when defining regulatory pathways.

For quite some time the interactions of the genes, either directly or through the means of their corresponding proteins, were studied for specific biological processes, and the rules of each interaction, called a biological pathway, are usually specified in terms of enabling/inhibiting factors, sometimes represented graphically in the form of logical circuits. Studying the pathways is of fundamental importance in biology and medicine since understanding the mechanisms of a biological process allows one to influence the process by modifying some of the enabling factors.

The new microarray technologies are able to provide measurements of thousands of gene expressions, sometimes by a differential method, where the expression of a gene in a sample of interest is measured against the expression in a reference sample. This is because the resulting gene expression ratio is less subject to the variation of the experimental and processing conditions. There are several alternative technologies for obtaining the gene expressions. Most of them are automated involving a robot arm which deposits the substances on the allocated spots on a microarray slide. The microarray slides are rectangular of dimensions in the order of centimeters having a large number of spots or wells arranged in a regular grid, each spot being the place where the abundance of a certain gene in a certain biological sample is measured. During the technology phases the genetic material from the probe at each spot is hybridized to the substance deposited on the spot, and by means of some color markings a high abundance of a gene in the probe is transformed into a high intensity color component. In the case of a differential measuring system the probe and the reference are marked with red and green colors, respectively, resulting in a composite color of the spot image. What is called a gene expression results finally from processing the image of the spot with the goal of integrating the contribu-

tion of the hybridization process in the area of the spot, while taking into account the particular features of the technological process.

After the creation and processing a microarray slide the net result is in the form of one value of the expression for each of the thousands of genes represented in the array (or in case of differential measurements, two values, one in the probe of interest and the other in the reference probe). In a complete experiment many microarray slides are produced and measured, for example, one slide for each patient.

## 2.2 Prediction of the gene expression values

In the present chapter we concentrate on processing the outcomes of such complete experiments, where we have a number of  $N$  patients, and for each we have the measurement of  $p$  gene expressions. In the following we denote by  $x_{i,j}$  the expression of gene  $i$  for patient  $j$ .

One of the main research avenues opened by the existence of various microarray technologies is a generalization of the biological pathways. These are used to describe the activation of certain biological functions in terms of the interaction between the genes and the proteins from the explanations based only on a handful of genes or on virtually all the known genes. In recent years a main research goal turned out to be uncovering the network of interactions between the genes. Analogously to the measurement of gene expressions, new technologies begin to be introduced for the measurement of protein expressions, see e.g. [10]. It will soon become possible to study experimentally the interaction of the gene network in connection with the protein network. Since it is still believed that the interactions in the network are sparse; i.e. each gene interacts only with a small number of other genes, the methodologies for uncovering the network still rely mostly on local modeling, where individual functions are established for the regulation of every gene. However, a number of approaches exist for modeling the full network or large parts of it, when enough information exists about the temporal behavior of the genes [3][22][23].

The methods described in this chapter aim at identifying the nonlinear relationships that are able to predict the expression of one gene, given the expressions of other genes and possibly the values of environmental factors measured. We define a nonlinear gene predictor as

$$\hat{x}_{ij} = f(x_{i_1j}, x_{i_2j}, \dots, x_{i_kj}), \quad (1)$$

where  $f$  is a nonlinear function,  $k$  the order of the predictor, and  $\{i_1, i_2, \dots, i_k\}$  the structural

indices of the predictor. The task of determining a predictor refers to an ensemble of problems: first, select the model structure (find the order  $k$  and the structural indices), and then specify the nonlinear function  $f(\cdot)$ .

A good prediction model is desired to make the predictions themselves, but it also should be able to uncover the biological relationships between genes. For the former goal the parameters in the model will necessarily have to be tuned to their optimal values, and the structure of the model will also have to be selected optimally to prevent overfitting.

### 2.3 Classification based on gene expression values

A second large class of applications, where the methods presented in this chapter are relevant, is the classification of diseases based on the transcriptome information [11][21]. Since the gene expression information may play an essential role in the characterization of the metabolic processes in cells and tissues, it is tempting to attempt at profiling the type of a disease based on the gene expressions of the biological sample studied. Most work has been done in this respect for identifying the types of the various forms of cancer disease. Again, apart from the interest in the performance of such a classifier, which soon promises to complement the traditional methods of diagnosis, there is also a separate interest of its own in the structure of the optimal classifier for the genes involved in separating the types of disease. This is because knowing the genes responsible for each type may be of help in designing specific treatments for each type of disease. Associating a class label  $y$  with each type of a disease, and having available a training sample formed of the expressions  $x_{ij}$  for each patient  $j$  and gene  $i$ , the design problem for the classifier consists of determining the structure and the parameters of the nonlinear function

$$\hat{y}_j = f(x_{i_1j}, x_{i_2j}, \dots, x_{i_kj}) \quad (2)$$

to be found to minimize a criterion based on the classification error.

## 3 Several classes of nonlinear functions and associated design methods

When selecting classes of nonlinear methods for modeling the gene interactions two important features of gene expression data need to be taken into account. On the one hand, the number of measured gene expressions in each slide is extremely high, even tens of thousands of genes.

Therefore selecting the sparse structure of the predictor; i.e. selecting a small number of genes out of all the existing genes raises the issue of having to compare a combinatorially very large number of candidates. On the other hand, the amount of microarray slides in almost all biological experiments is not particularly large, rarely exceeding 100 slides, which makes fitting the parameters in very complex models difficult due to the lack of enough observed cases. As a consequence, the models employed in gene expression predictors and classifiers are usually simple like the perceptron models, the Boolean models, and some simple neural networks. As hinted in the previous section, the selection of the structure of the predictor is of main interest in many applications; i.e. one wants to find the genes that are involved in a certain biological process or in profiling a certain disease, while the detailed functional form of the predictor/classifier is of lesser importance.

### 3.1 The Boolean model with binary inputs

The simplest model that biologists found appealing [6] from the earliest times is the Boolean model. Such a model operates on binary inputs, which creates the need to quantize the gene expressions before they can be used in the model. An advantage of the model is its one-to-one compatibility with the existing pathway representations based on Boolean schemes. As an example, Figure 1 illustrates the FAS signaling pathway, (obtained from <http://www.biocarta.com/>) which is involved in immune surveillance to remove transformed cells and virus infected cells. One of the interactions shows that *Caspase8* is produced when *Pro-caspase8* is on and inhibited when both *FAP* and *I-FLICE* are on. These models have the additional advantage of being simple to understand and manipulate by biologists with an easy integration of various local functions into an overall function.

Denoting the quantized binary expressions of *Pro-caspase8*, *FAP*, *I-FLICE* and *Caspase8* as  $x_1, x_2, x_3$  and  $x_4$ , respectively, the Boolean model can be expressed as

$$x_4 = x_1 \bar{x}_2 \bar{x}_3, \quad (3)$$

where by over bar we denote the negation of a binary variable and by the product the logical ‘and’ function.

In the following we use  $x_{ij}$  for the expression value and sometimes the quantized expression value, and we specify in each context the number of the quantization levels and the centers of the Voronoi cells (hence  $x_{ij}$  may be binary, ternary or even continuous valued, depending on the

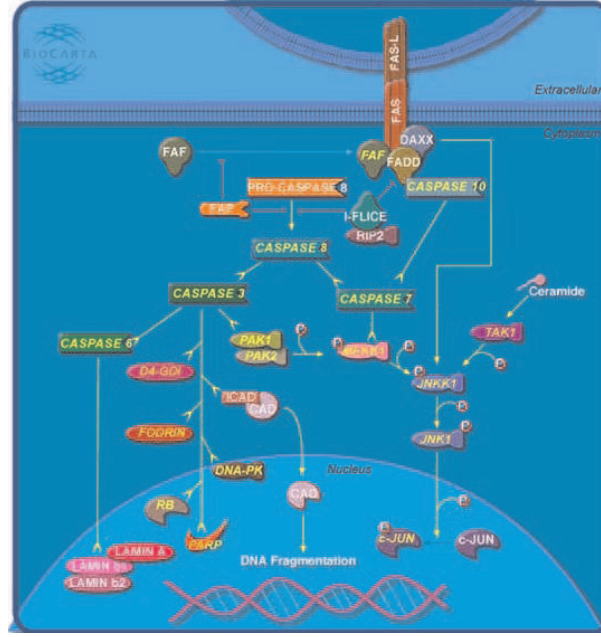


Figure 1: A biological pathway where gene and protein interactions are represented by logical operators. (reproduced from Biocarta, <http://www.biocarta.com/>)

context).

An optimal design of a Boolean predictor is straightforward, and we describe it here just for the sake of completeness and ease of comparison with the design of more complex models. We illustrate the design for a given structure, say  $\hat{x}_4 = f(x_1, x_2, x_3)$ , from a training set of quadruples  $\{(x_{1j}, x_{2j}, x_{3j}; x_{4j}), j = 1, \dots, N\}$ . In general the prediction given by the model  $\hat{x}_{4j}$  and the measured value  $x_{4j}$  are different, and we refer to their difference as modeling error

$$e_j = \hat{x}_{4j} \oplus x_{4j} = f(x_{1j}, x_{2j}, x_{3j}) \oplus x_{4j}, \quad (4)$$

where  $\oplus$  denotes the ‘exclusive or’ operator. We get that  $\hat{x}_{4j} = e_j \oplus x_{4j}$  and  $x_{4j} = e_j \oplus \hat{x}_{4j}$ . The Boolean function minimizing the error criterion

$$\sum_{j=1}^N (f(x_{1j}, x_{2j}, x_{3j}) \oplus x_{4j}) \quad (5)$$

can easily be found by counting. For each triplet of binary values  $(b_1, b_2, b_3) \in \mathcal{B}^3$  we have the count  $n_0(b_1, b_2, b_3)$  of the number of times  $x_{4j}$  is 0 when the input variables have values  $x_{1j} = b_1, x_{2j} = b_2, x_{3j} = b_3$ , and similarly the count  $n_1(b_1, b_2, b_3)$  of the number of times  $x_{4j}$  is

1. Then the optimal Boolean function minimizing the criterion (5) is given by

$$f(b_1, b_2, b_3) = \begin{cases} 0 & \text{if } n_0(b_1, b_2, b_3) \geq n_1(b_1, b_2, b_3) \\ 1 & \text{otherwise.} \end{cases} \quad (6)$$

If not enough cases are observed it is clear that for some combinations  $(b_1, b_2, b_3) \in \mathcal{B}^3$  both counts  $n_0(b_1, b_2, b_3)$  and  $n_1(b_1, b_2, b_3)$  may be zero (or equal), and no clear cut decision can be made. For the criterion (5) the undefined values do not matter, but when the model is applied to data outside the training set the non-definiteness of the model for some input combinations should be solved in a principled way. These issues will be discussed later in Section 4.3.1.

The solution to the prediction problem presented is immediately applicable to the classification problem, where the training set is of the form  $\{(x_{1j}, x_{2j}, \dots, x_{kj}; y_j), j = 1, \dots, N\}$  and the class predictor is  $\hat{y}_j = f(x_{1j}, x_{2j}, \dots, x_{kj})$ .

### 3.2 Three predictors for ternary data

We consider next three predictor classes for ternary valued gene expressions: a hard Boolean predictor, a hard ternary predictor, and finally, a perceptron predictor. We discuss the parameter estimation for all of them, and we present later the selection of the predictor order based on the MDL principle. The ternary valued data are written as  $\{0, 1, 2\}$ . For  $x \in \{-1, 0, 1\}$  use first the mapping  $x \leftarrow x + 1$  to transform  $x$  to the set  $\{0, 1, 2\}$ .

#### 3.2.1 The specification and design of the Hard Boolean predictor

Denote by  $\underline{x} = [x_1, \dots, x_k]$  the prediction window of dimension  $k$ , where  $x_i \in \{0, 1, 2\}$ . Define the thresholded vectors  $\underline{x}^{b_1} = [x_1^{b_1}, \dots, x_k^{b_1}]$  and  $\underline{x}^{b_2} = [x_1^{b_2}, \dots, x_k^{b_2}]$ , where

$$x_i^{b_1} = \begin{cases} 0 & \text{if } x_i = 0 \\ 1 & \text{if } x_i \geq 1 \end{cases} \quad (7)$$

$$x_i^{b_2} = \begin{cases} 0 & \text{if } x_i \leq 1 \\ 1 & \text{if } x_i = 2. \end{cases} \quad (8)$$

The predictor is defined as

$$\hat{y} = f(\underline{x}^{b_1}) + f(\underline{x}^{b_2}), \quad (9)$$

where  $f(\cdot)$  is a Boolean function with  $k$  variables.

The design of the Boolean predictor proceed as described in Section 3.1, where the training set is given by  $\{[x_{1j}^{b_1}, \dots, x_{kj}^{b_1}, y_j], [x_{1j}^{b_2}, \dots, x_{kj}^{b_2}, y_j] : j = 1, \dots, N\}$ , and the corresponding optimal Boolean function results from the counts as in (6).

### 3.2.2 The specification and design of the Ternary predictor

The optimal Ternary predictor is found by quantization of the conditional expectation to three intervals:

$$\hat{y} = h^*(\underline{x}) = \begin{cases} 0 & \text{if } E(y|\underline{x}) \leq 0.5 \\ 1 & \text{if } 0.5 < E(y|\underline{x}) \leq 1.5 \\ 2 & \text{if } 1.5 < E(y|\underline{x}). \end{cases} \quad (10)$$

### 3.2.3 The specification and design of the Perceptron predictor

The perceptron is found by quantization the best linear combination of samples in the predictor window to three intervals :

$$\hat{y} = \begin{cases} 0 & \text{if } \underline{w}^T + w_0 \underline{x} \leq 0.5 \\ 1 & \text{if } 0.5 < \underline{w}^T \underline{x} + w_0 \leq 1.5 \\ 2 & \text{if } 1.5 < \underline{w}^T \underline{x} + w_0. \end{cases} \quad (11)$$

The parameters  $\underline{w}$  and  $w_0$  can be obtained by the Perceptron algorithm [7],[8].

## 3.3 Model selection based on a two part code length

Our approach to prediction aims at finding flexible classes of models with good predictive properties, but we also consider the complexity of the models, the balance being set by the minimum description length (MDL) principle, as described in the next section.

### 3.3.1 Modeling and data compression

Does a model capture the data generation mechanism? If yes, we can describe the data more concisely by describing the model and listing the errors than by copying the data in ‘raw’ point by point; i.e. encoding without a model or, perhaps, with the trivial identity model.

A well established fact is that both the complexity of models and the data can be measured by the universal yardstick of code length. The MDL principle [1][12][13][14] considers the following axiom as a basis for a theory of modeling: given the data and a model class, select the model in

the class with which both the data and the model can be encoded with the shortest code length. This means that the code length needed results from a two-part encoding process: Encode first the model parameters and then the residuals, given the model defined by the parameters. When the parameters range over the reals they must be optimally quantized. In reality we only need the code length rather than the actually encoded strings for the data and the quantized parameters, by a deeper theory of coding the shortest code length can be computed from a special parameter-free model that is *universal* for the model class. The same approach can be applied to different model classes, which may be compared by calculating the shortest code length for each and selecting the class with the shortest overall code length.

To apply the technique to our data we postulate a parametric model for the dependency  $\hat{x}_\ell = f(x_1, \dots, x_k, \theta)$ , fit its parameters  $\theta$  to the available data, and find the code length for the parameters and the residuals. If we get an important reduction in the code length with a model class we claim that the dependency modeled is significant. If different model classes are consistently capturing dependencies, they are very likely to be rules of the nature.

### Encoding without a model

We take as an example the case of  $N = 30$  patients, which we use in our experiments. Encoding the expressions for the gene  $q_\ell$  without a model requires  $L = \log_2(3^{30}) = 47.55$  bits. Other codes for the target gene alone may be used, but we find it convenient to make no assumption about the distribution of  $-1, 0, 1$  at this point.

### Encoding the residuals

When we select a predictive model we have to decide on a code for the residuals. At a first glance the residuals may be more difficult to encode than the original sequence since the dynamic range is now expanded to  $\{-2, -1, 0, 1, 2\}$ . However, there is a reason why the residuals may be easier to encode: a good predictor will cause the sequence of the residuals to contain the symbol 0 many times. With arithmetic coding, or simply with run length coding, this can be taken advantage of, and we get a smaller code length than that of the original sequence. A less optimal result can be obtained with an even simpler code as described next.

The residuals will be encoded in a way to penalize the nonzero errors. For each error we send the actual (correct) value encoded as "0" = 01, "1" = 10, " - 1" = 11, the codeword 00 signaling the end of the residual sequence. After the actual value we also transmit the location

of the error with  $\lceil \log_2 N \rceil = \lceil \log_2 30 \rceil = 5$  bits. An error will therefore be transmitted with 7 bits.

### 3.3.2 The hard Boolean predictor: the description length

We consider the two-part code: first we encode the model “parameters” needed to specify the function  $f^*(\cdot)$ , and then the residuals (prediction errors).

The model cost for encoding the function  $f^*(\cdot)$  is

$$C_M(k) = 2^k \text{ bits} \quad (12)$$

if we assume a uniform apriori distribution for all the Boolean functions. Other selections are also possible; i.e. to favor the more active Boolean functions (e.g. according to the absolute difference between the number of ones and zeros in  $f$ ). Therefore, if there are  $N_e$  nonzero prediction errors, the overall code length is the model length plus the prediction error length:

$$L(k) = 2^k + 7N_e + 2 \text{ bits.} \quad (13)$$

Starting from a data set of several possible gene factors we can select the best combination by designing sequentially the predictors with the window size  $k = 2, 3, 4, 5$  and computing  $L(k)$  for each predictor. The MDL principle selects as the optimal window size  $k$  for which  $L(k)$  is minimum.

### 3.3.3 The Ternary predictor: the description length

To encode the model we have two possibilities.

1. The values of the prediction window  $\underline{x}$  can be ordered lexically as  $[0\ 0\ 0], [0\ 0\ 1], [0\ 0\ 2], \dots$ . We may specify the optimal ternary predictor by the string of values  $h^*([0\ 0\ 0]), h^*([0\ 0\ 1]), h^*([0\ 0\ 2]), \dots$ , which needs  $2 \cdot 3^k$  bits. Therefore the cost of encoding the ternary model is  $C_M = 2 \cdot 3^k$  bits, which is the same for all models, irrespective of the data. However, note that we do not know the optimal values  $h^*(\underline{x})$  for the prediction windows  $\underline{x}$  which have not been seen.

2. Therefore, in the second alternative the cost of the model depends on the actual data and we have to specify the optimal predictions only for the seen prediction windows. Denoting by  $n_x(k)$  the number of different prediction windows found in the data set the cost of encoding the model becomes  $C_M = 2n_x(k)$  bits, which is upper bounded by  $2N$ , where  $N$  is the number

of measurements. This variant is used in our experimental determinations. Note that a similar variant can be used also for the Boolean predictor.

The prediction errors will be encoded the same way as in the case of the Boolean prediction. Therefore, if  $N_e$  nonzero prediction errors are obtained with the Ternary predictor, the code length for the prediction residuals is  $7N_e + 2$ .

The overall code length is then the model length plus the prediction error length, or

$$L(k) = 2n_x(k) + 7N_e + 2 \text{ bits.} \quad (14)$$

Apparently the ternary predictor is defined only for the prediction windows seen in the experiment. There is an obvious way to use the observed data to define  $h^*(\underline{x})$  for the unseen prediction windows  $\underline{x}$ . A simple way is to take  $h^*(\underline{x}) = h^*(\underline{x}')$ , where  $\underline{x}'$  is a seen prediction window of size smaller than  $\underline{x}$ . However, in the current gene expression problem the goal is to identify outstanding gene dependencies, given the observed data, and it is unnecessary to specify the predictor for the unseen windows.

### 3.3.4 The Perceptron predictor: the description length

For ternary valued data the number of all distinct perceptrons with two or three inputs are known, [24]. A perceptron model with two input genes requires  $C_M = \log_2 471 = 8.88$  bits, while a perceptron model with three input genes requires  $C_M = \log_2 85629 = 16.38$  bits.

## 3.4 An artificial example

We take first an artificial example, where, knowing the true state of nature, we will be able to check the success of our procedure.

We assume  $N = 30$  measurements (patients), at a patient  $i$  40 variables (genes)  $x_{1,i}, \dots, x_{40,i}$  and one target, denoted  $y(i)$ . Suppose the measurements are quantized to three levels: 0, 1, 2. The number of the sequences of 30 symbols with ternary values is  $3^{30} = 2 \cdot 10^{14}$ , which exceeds hugely the 40 sequences we have in the experiment. Suppose that there is a function  $f^* : \{0, 1, 2\}^4 \rightarrow \{0, 1, 2\}$ ,  $y(i) = f^*(x_{t_1,i}, x_{t_2,i}, x_{t_3,i}, x_{t_4,i})$ , which describes the target exactly.

The data were generated by choosing randomly 40 sequences of 30 measurements from all the possible  $2 \cdot 10^{14}$  ternary sequences, with a uniform prior on them. The target function  $f^* : \{0, 1, 2\}^4 \rightarrow \{0, 1, 2\}$  is selected in the following way: for any input window  $(j_1, j_2, j_3, j_4)$  the value  $f^*(j_1, j_2, j_3, j_4)$  is obtained by sampling a random variable over the values 0, 1, 2 with

the probability distribution  $p = [\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$ . The target gene is then constructed as the exact function  $y(i) = f^*(x_1(i), x_2(i), x_3(i), x_4(i))$ . Therefore, the true window for the target is  $[x_1(i), x_2(i), x_3(i), x_4(i)]$ , referred [1, 2, 3, 4] for short. We show in the following that the Boolean and the Ternary predictors can be used in conjunction with the MDL principle to select the prediction window candidates.

**The Target alone:** The complexity of the target without conditioning on other genes is evaluated by three methods: (a) A prior uniform distribution for all sequences, which gives  $L(y^N) = -\lceil \log_2 3^{30} \rceil = 48$  bits; (b) An adaptive arithmetic coding, which gives the value  $L(y^N) = 47$  bits; (c) Move-to-Front coding (which favors correlated sequence), giving the length  $L(y^N) = 52$  bits.

Since the target was generated as an uncorrelated sequence of ternary values, the Move-to-Front procedure gives rather different results from the first two methods, but we can safely assume that the complexity of the target alone is in the range of 47–52 bits.

**The window size=2:** No predictor with only two genes can achieve a description length lower than the complexity of the target. In the left of Table 1 we list the best models of order two, which give a description length less than 72, the total of 17 windows. The "correct" window, which is of order four containing gene<sub>1</sub>, gene<sub>2</sub>, gene<sub>3</sub>, and gene<sub>4</sub>, has a "trace" in the list of the best models, which occupies the second position with the window (gene<sub>1</sub>, gene<sub>3</sub>).

**The window size=3:** The predictors with three genes are more successful in describing the target. In the middle of Table 1 we list the best models of order three, which give a description length less than 55, making the total of 28 windows.

**The window size=4:** We list the best models of order four (the ones having length less than 49, making the total of 23 windows). We show in bold the "true" model, which was ranked the 19'th out of 91390 possible prediction windows. Observe that the variations of the true window have been selected frequently as winners in the final list.

### 3.5 A comparison of the three classes of models in a small scale experiment

We consider here a comparison of the three ternary predictors by the MDL criterion on the experimental data from [8] having  $N = 30$  and  $p = 13$ , presented in Table 7 of the appendix.

<i>Descr.Length</i>	$g_1$	$g_2$	<i>Descr.Length</i>	$g_1$	$g_2$	$g_3$
60.00	15	40	44.00	1	3	12
65.00	<b>1</b>	<b>3</b>	47.00	10	32	36
67.00	4	13	48.00	16	20	31
67.00	9	40	48.00	30	32	38
67.00	10	11	49.00	12	15	40
67.00	10	34	49.00	14	19	23
67.00	14	20	50.00	2	10	32
67.00	20	31	50.00	10	31	34
67.00	20	32	50.00	15	20	32
67.00	20	40	50.00	22	33	34
67.00	26	34	52.00	6	20	25
67.00	27	40	52.00	8	10	12
67.00	38	40	52.00	8	20	32
72.00	3	34	52.00	13	18	19
72.00	11	34	53.00	<b>1</b>	<b>2</b>	<b>3</b>
72.00	27	32	53.00	10	11	34
72.00	32	36	53.00	10	15	40

<i>Descr.Length</i>	$g_1$	$g_2$	$g_3$	$g_4$
46.00	1	2	3	12
46.00	1	3	12	38
46.00	12	30	32	38
46.00	30	32	34	38
46.00	8	10	31	34
46.00	1	12	32	38
46.00	12	30	32	38
48.00	1	3	12	18
48.00	10	27	32	36
48.00	4	16	20	31
48.00	2	10	25	32
48.00	2	10	29	32
48.00	10	25	31	34
48.00	15	20	25	32
48.00	15	20	32	40
48.00	6	20	25	40
48.00	8	20	32	34
48.00	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
48.00	10	11	13	34
48.00	15	20	32	40
49.00	1	3	12	34
49.00	12	16	20	31

Table 1: **Artificial data.** The prediction windows of the best predictors of lengths two, three and four (the framed one of length four was used to generate the data)

The target gene is *AHA*.

In Table 2 we present the values of the description length and also the number of the errors, obtained with the best predictors in each of the three classes for various sizes of the prediction window, when the genes are restricted only to the set  $(RCH1, p53, PC-1)$ . With the Boolean predictor the best description length is obtained with the model  $HB(RCH1, p53)$ . Appending  $PC-1$  to the prediction window does not improve the descriptive power of the model. The same conclusion can be drawn for the hard ternary predictors, and the perceptrons. The MDL criterion makes therefore a clear and consistent selection of the best prediction window, while the coefficient of the determination  $c_d$  [8] simply shows a better fit of the model to the data with an increasing window size. The problem with  $c_d$  is that it expresses only the fit to the training data. To avoid overtraining one needs another (validation) data set. The MDL principle is a method, which implicitly penalizes too large models, and which can be shown to provide consistent order estimations unlike the several cross-validation methods.

We extend now the experiment to predicting the values of the target *AHA* based on the combinations of two, three, or four of the genes in the first 12 columns of Table 7. In Tables 3, 4, 5 we show the best predictors for each window size. The best overall predictor is seen to be  $AHA = HB(p53, ATF3, RCH1)$ , which has the Boolean function  $f(x_1, x_2, x_3) = x_1x_2 + x_2x_3$ , and the description length  $L = 10$ . This is much lower than what can be obtained by any other window size or a combination of genes. By comparing the description length of the predictors of different sizes, we can now conclude that most of the "good" predictors of order 3 or 4 are worse than the corresponding predictors of lower order; that is, all predictors of size three with the description length 24 are actually worse than the predictor of order two  $AHA = HB(p53, RCH1)$ , which has the description length 20.

We are therefore capable of organizing the huge number of "good predictors" in Tables 3, 4, 5 and discarding all the overly complex ones, to conclude that the relevant predictors in light of the data are only:  $HB(p53, RCH1)$ ,  $HB(p53, ATF3, RCH1)$ , and  $HT(p53, ATF3, BCL3)$ .

Hard Boolean Predictor			
	AHA=HBP(RCH1,p53,PC-1)	AHA=HBP(RCH1,p53)	AHA=HBP(p53)
$N_e$	2	2	12
$Length$	24	20	88
Hard Ternary Predictor			
	AHA=HT(RCH1,p53,PC-1)	AHA=HT(RCH1,p53)	AHA=HT(p53)
$N_e$	1	2	7
$Length$	25	24	55
Perceptron Predictor			
	AHA=Per(RCH1,p53,PC-1)	AHA=Per(RCH1,p53)	AHA=Per(p53)
$N_e$	2	2	7
$Length$	30.38	22.87	53.24
$c_d$ (MSE)[8]	0.946	0.785	0.624

Table 2: **“3 Gene” Experiment.** The description length for three predictors : Hard Boolean Predictors (HBP), Hard Ternary Predictor(HT) and the Perceptron Predictor (Per)

Length	20	38	45
Gene <sub>1</sub>	p53	p53	p53
Gene <sub>2</sub>	RCH1	BCL3	ATF3
Method	HB	HT	HT

Table 3: **“12 Gene” Experiment.** The description length of the best predictors with window size 2.

Length	10	24	24	24	24	24	24	24	24	24	30	39	42
Gene <sub>1</sub>	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53
Gene <sub>2</sub>	ATF3	BCL3	FRA1	RELB	IAP1	PC-1	MBP1	SSAT	MDM2	p21	ATF3	ATF3	RELB
Gene <sub>3</sub>	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	BCL3	RELB	BCL3
Method	HB	HB	HB	HB	HB	HB	HB	HB	HB	HB	HT	HT	HT

Table 4: **“12 Gene” Experiment.** The description length of the best predictors with window size 3.

Length	18	18	18	18	18	18	18	18	18	27	27	29	29
Gene <sub>1</sub>	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53
Gene <sub>2</sub>	ATF3	ATF3	ATF3	IAP1	PC-1	MBP1	SSAT	MDM2	p21	PC-1	MBP1	PC-1	MBP1
Gene <sub>3</sub>	BCL3	FRA1	RELB	ATF3	ATF3	ATF3	ATF3	ATF3	ATF3	BCL3	PC-1	IAP1	FRA1
Gene <sub>4</sub>	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1
Method	HB	HB	HB	HB	HB	HB	HB	HB	HB	HT	HT	HT	HT

Length	29	29	29	30	30	30	30	31	31	31	32	32	32
Gene <sub>1</sub>	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53	p53
Gene <sub>2</sub>	SSAT	SSAT	MDM2	MBP1	SSAT	MDM2	p21	FRA1	PC-1	p21	RELB	RELB	IAP1
Gene <sub>3</sub>	FRA1	PC-1	PC-1	IAP1	MBP1	MBP1	MBP1	BCL3	RELB	PC-1	BCL3	FRA1	BCL3
Gene <sub>4</sub>	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1	RCH1
Method	HT	HT	HT	HT	HT	HT	HT	HT	HT	HT	HB	HB	HB

Table 5: “12 Gene” Experiment. The description length of the best predictors with window size 4.

## 4 Normalized maximum likelihood models for a class of Boolean regressor models

The selection of the model order based on the two-part code discussed in the previous sections is widespread for a number of reasons. First, it is easily interpretable and intuitive in that it separates nicely the costs of the model and the remaining non-modeled part of the string (erroneous according to the model). Secondly, one of the asymptotic forms of the two-part code has been shown in many applications to be quite successful, and it was further shown to be equivalent with the Bayesian information criterion. However, in the recent years further advances in the MDL principle have been made, and a specific code, induced by the so-called normalized maximum likelihood (NML) model, was shown to possess important optimality properties. We illustrate in the rest of the chapter the derivation of the NML model and its application to a class of nonlinear models, first introduced by us in [19]. We follow it closely in this presentation.

The NML model for the linear regression problem was introduced and analyzed recently, [15]. We restate classification as a modeling problem in terms of a class of parametric models, for which the maximum likelihood parameter estimates can be easily computed. We review first the NML model for Bernoulli strings as the solution to a minmax optimization problem. We then introduce a model class for the case where the binary strings to be modeled are observed jointly with several other binary strings (regression variables). We derive the NML model for this model class. We further provide a fast evaluation procedure and apply it to a classification problem.

### 4.1 The NML model for Bernoulli strings

In this section we assume that a Bernoulli variable  $Y$  with  $P(Y = 0) = \theta$  is observed repeatedly  $n$  times, generating the string  $y^n = y_1, \dots, y_n$ . We look for a distribution  $q(y^n)$  over all strings of length  $n$  such that the ideal code length  $\log \frac{1}{q(y^n)}$  assigned to a particular string  $y^n$  by this distribution, is as close as possible to the ideal code length  $\log \frac{1}{P(y^n|\hat{\theta}(y^n))}$ , obtainable with the Bernoulli models. The words ‘ideal code length’ are used because they need not be integer valued as real code lengths must be. For long strings they differ from the real code lengths by at most unity. In the coding scenario the decoder is allowed to use a predefined distribution,  $q(\cdot)$ , but he cannot use  $P(y^n|\hat{\theta}(y^n))$  because it is not a distribution. The latter, however, will be a target which no model in the class can beat since it maximizes  $P(y^n|\theta)$  and therefore minimizes

the ideal code length  $\log \frac{1}{P(y^n|\theta)}$ . The distribution  $q(y^n)$  is selected such that the "regret" of using  $q(y^n)$  instead of  $P(y^n|\hat{\theta}(y^n))$ , namely,

$$\log \frac{1}{q(y^n)} - \log \frac{1}{P(y^n|\hat{\theta}(y^n))} = \log \frac{P(y^n|\hat{\theta}(y^n))}{q(y^n)}, \quad (15)$$

is minimized for the worst case  $y^n$ ; i.e.

$$\min_q \max_{y^n} \log \frac{P(y^n|\hat{\theta}(y^n))}{q(y^n)}. \quad (16)$$

**Theorem 1** (*Shtarkov[17]*) *The minimizing distribution is given by*

$$q(y^n) = \frac{P(y^n|\hat{\theta}(y^n))}{C_n}, \quad (17)$$

where

$$C_n = \sum_{m=0}^n \binom{n}{m} \left(\frac{m}{n}\right)^m \left(1 - \frac{m}{n}\right)^{n-m}. \quad (18)$$

A strong optimality property of the NML models was recently proven in [16], where the following minmax problem was formulated: find the (universal) distribution which minimizes the average regret

$$\min_q \max_g E_g \log \frac{P(Y^n|\hat{\theta}(Y^n))}{q(Y^n)}, \quad (19)$$

where  $g(\cdot)$ , the generating distribution of the data, and  $q(\cdot)$  run through any sets that include the NML model.

**Theorem 2** (*[16]*) *The minimizing distribution  $q(\cdot)$  in the minmax problem (19) is given by (17) and (18).*

If we replace 'minmax' by 'maxmin' the worst case distribution is unique and also given by (17) and (18).

## 4.2 The NML model for a Boolean class

We consider a binary random variable  $Y$ , which is observed jointly with a binary regressor vector  $\underline{X} \in \mathcal{B}^k$ . In a useful model class a carefully selected Boolean function  $f : \mathcal{B}^k \rightarrow \{0, 1\}$  should provide a reasonable prediction  $f(\underline{X})$  of  $Y$  in the sense that the absolute error  $\mathcal{E} = |Y - f(\underline{X})|$  has a large probability of being 0. Since  $\mathcal{E}, Y, f(\underline{X})$  are binary-valued we have  $\mathcal{E} = |Y - f(\underline{X})| = Y \oplus f(\underline{X})$ , which also implies  $Y = f(\underline{X}) \oplus \mathcal{E}$ , where  $\oplus$  is modulo 2 sum.

We therefore consider a corruption model defined as follows:

$$Y = f(\underline{X}) \oplus \mathcal{E} = \begin{cases} f(\underline{X}) & \text{if } \mathcal{E} = 0 \\ \overline{f(\underline{X})} & \text{if } \mathcal{E} = 1, \end{cases} \quad (20)$$

where  $f(\cdot)$  is a Boolean function and the error  $\mathcal{E}$  is independently drawn from a Bernoulli source with parameter  $\theta$ ; i.e.,  $P(\mathcal{E} = 1) = 1 - \theta$  and  $P(\mathcal{E} = 0) = \theta$ , or for short

$$P(\mathcal{E} = b) = \theta^{1-b}(1 - \theta)^b, \text{ for } b \in \{0, 1\}. \quad (21)$$

Denote by  $\underline{b}_i \in \{0, 1\}^k$  the vector having as entries the bits in the binary representation of integer  $i$ , i.e.,  $\underline{b}_0 = [0, \dots, 0, 0]$ ,  $\underline{b}_1 = [0, \dots, 0, 1]$ , etc. Further, define by (20) and (21) the conditional probability for code  $\underline{b}_i \in \{0, 1\}^k$ ,

$$P(Y = y | \underline{X} = \underline{b}_i) = \theta^{1-y \oplus f(\underline{b}_i)}(1 - \theta)^{y \oplus f(\underline{b}_i)}. \quad (22)$$

The Boolean regression problem will be stated as finding the optimal universal model (in a minmax sense to be specified shortly) for the following class of models:

$$\mathcal{M}(\theta, k, f) = \{P(y|f, \underline{b}_i, \theta) = \theta^{(1-y \oplus f(\underline{b}_i))}(1 - \theta)^{(y \oplus f(\underline{b}_i))}\}, \quad (23)$$

where  $y \in \{0, 1\}, \theta \in [0, 1], \underline{b}_i \in \{0, 1\}^k$ .

When the sequence  $y^n = y_1 \dots y_n$  and the sequence of binary regressor vectors  $\underline{b}^n = \underline{b}_{i_1}, \dots, \underline{b}_{i_n}$  are observed, a member of the class  $\mathcal{M}(\theta, k, f)$  assigns to the sequence  $y^n$  the following probability

$$\begin{aligned} P(y^n | \theta, k, f, \underline{b}^n) &= \prod_{j=1}^n \theta^{(1-y_j \oplus f(\underline{b}_{i_j}))}(1 - \theta)^{(y_j \oplus f(\underline{b}_{i_j}))} \\ &= \theta^{n_0}(1 - \theta)^{n - n_0}, \end{aligned} \quad (24)$$

where  $n_0$  is the number of zeros in the sequence  $\{\varepsilon_j = y_j \oplus f(\underline{b}_{i_j})\}_{j=1}^n$ . The ML estimate of the model parameters,

$$(\hat{\theta}(y^n), \hat{f}_{y^n}) = \arg \max_{\theta, f} P(y^n | \theta, k, f, \underline{b}^n), \quad (25)$$

can be obtained in two stages, first by maximizing with respect to  $f$ ,

$$\max_f P(y^n | \theta, k, f, \underline{b}^n), \quad (26)$$

and observing that the optimal  $f(\cdot)$  does not depend on  $\theta$ . For a fixed  $\theta > 0.5$ , the function  $P(y^n|\theta, k, f, \underline{b}^n) = \theta^{n_0}(1-\theta)^{n-n_0}$  decreases monotonically with  $n_0$ , and (26) is maximized by maximizing  $n_0$ , or, equivalently, by minimizing  $n - n_0$

$$\begin{aligned} \min_f (n - n_0) &= \min_f \sum_{j=1}^n |y_j - f(\underline{b}_{i_j})| \\ &= \min_f \sum_{\ell=0}^{2^n} m_{\ell_0} f(\underline{b}_\ell) + m_{\ell_1} (1 - f(\underline{b}_\ell)), \end{aligned} \quad (27)$$

where  $m_{\ell_0}$  and  $m_{\ell_1}$  denote the number of times  $y_j = 0$  and  $y_j = 1$ , respectively, have been seen at the regressor vector  $\underline{b}_{i_j} = \underline{b}_\ell$ .

Equation (27) shows that  $f$  is optimal for the mean absolute error (MAE) criterion. It can also be seen that the assignment of  $f(\underline{b}_\ell)$  depends only on  $m_{\ell_0}, m_{\ell_1}$ , and the solution is

$$\hat{f}_{y^n}(\underline{b}_\ell) = \begin{cases} 0 & \text{if } m_{\ell_0} \geq m_{\ell_1} \\ 1 & \text{if } m_{\ell_0} < m_{\ell_1} \end{cases}, \quad (28)$$

which can be readily computed from the data set. Denote by  $n_0^*(y^n)$  the number of zeros in the sequence  $\{\varepsilon_j = y_j \oplus \hat{f}_{y^n}(\underline{b}_{i_j})\}_{j=1}^n$ . To completely solve the ML estimation problem we have to find

$$\max_{\theta} P(y^n|\theta, k, \hat{f}_{y^n}, \underline{b}^n), \quad (29)$$

for which the maximizing parameter is  $\hat{\theta}(y^n) = \frac{n_0^*(y^n)}{n}$ . Therefore

$$\begin{aligned} &P(y^n|\hat{\theta}(y^n), k, \hat{f}_{y^n}, \underline{b}^n) \\ &= \left(\frac{n_0^*(y^n)}{n}\right)^{n_0^*(y^n)} \left(1 - \frac{n_0^*(y^n)}{n}\right)^{n-n_0^*(y^n)}. \end{aligned} \quad (30)$$

We need to define a distribution  $q(y^n)$  over all possible sequences  $y^n$  which is the best in the minmax sense

$$\min_q \max_{y^n} \frac{P(y^n|\hat{\theta}(y^n), k, \hat{f}_{y^n}, \underline{b}^n)}{q(y^n)}. \quad (31)$$

This is clearly given by the NML model,

$$q(y^n) = \frac{P(y^n|\hat{\theta}(y^n), k, \hat{f}_{y^n}, \underline{b}^n)}{C_n(k, \underline{b}^n)}, \quad (32)$$

where

$$C_n(k, \underline{b}^n) = \sum_{y^n} \left(\frac{n_0^*(y^n)}{n}\right)^{n_0^*(y^n)} \left(1 - \frac{n_0^*(y^n)}{n}\right)^{n-n_0^*(y^n)}. \quad (33)$$

Note that  $n_0^*$  depends on  $y^n$  through  $\hat{f}_{y^n}$  in a complicated manner. When  $k = 0$ , the normalization factor is  $C_n(0, \underline{b}^n) = C_n$ , given in (18).

Alternative expressions for the coefficient  $C_n(k, \underline{b}^n)$  provide faster evaluation. Let  $\{\underline{b}_{j_1}, \dots, \underline{b}_{j_K}\}$  be the set of the distinct elements in the set  $\{\underline{b}_\ell | \underline{b}_\ell \in \underline{b}^n\}$ , and let  $K = K(\underline{b}^n)$  be the number of the distinct regressor vectors. Denote by  $z^q$  the subsequence of  $y^n$  observed when the regressor vector is  $\underline{b}_{j_q}$ . Let  $n_q$  be the length of the subsequence  $z^q$  having  $m_q$  zeros.

We observe that (33) can be alternatively expressed as

$$C_n(k, \underline{b}^n) = \sum_{n_1^*=0}^n \binom{n_1^*}{n} \left(1 - \frac{n_1^*}{n}\right)^{n-n_1^*} S_{K, n_1, \dots, n_K}(n_1^*),$$

where  $S_{K, n_1, \dots, n_K}(n_1^*)$  is the number of sequences  $y^n$  having  $n_1^* = \sum_{q=1}^K \min(m_q, n_q - m_q)$  ones in the residual sequence. The numbers  $S_{K, n_1, \dots, n_K}(n_0^*)$  can be easily computed recursively in  $K$ . Denote first

$$h_\ell(m) = \begin{cases} 0 & \text{if } m > \frac{n_\ell}{2} \\ \binom{n_\ell}{m} & \text{if } m = \frac{n_\ell}{2} \\ 2 \binom{n_\ell}{m} & \text{else} \end{cases}, \quad (34)$$

which is the number of sequences of length  $n_\ell$  having either  $m$  bits set to 1 or  $n_\ell - m$  bits set to 1, for  $0 \leq m \leq \frac{n_\ell}{2}$ . By combining each of the  $S_{K-1, n_1, \dots, n_{K-1}}(n_1^* - m_K)$  sequences having  $n_1^* - m_K$  ones in the residual sequence with each of the  $h_K(m_K)$  sequences having either  $m_K$  bits set to 1 or  $n_K - m_K$  bits set to 1, we get sequences having  $(n_1^* - m_K) + \min(m_K, n_K - m_K) = n_1^*$  occurrences of 1 in their residual sequence. Therefore the following recurrence relation holds:

$$S_{K, n_1, \dots, n_K}(n_1^*) = \sum_{m_K=0}^{n_K} h_K(m_K) S_{K-1, n_1, \dots, n_{K-1}}(n_1^* - m_K), \quad (35)$$

where by convention  $S_{K-1, n_1, \dots, n_{K-1}}(n_1^* - m_K) = 0$  for negative arguments,  $n_1^* - m_K < 0$ .

We note that the recurrence is simply a convolution sum,  $S_{K, n_1, \dots, n_K} = h_K \otimes S_{K-1, n_1, \dots, n_{K-1}}$ , and we conclude that

$$S_{K, n_1, \dots, n_K} = h_1 \otimes h_2 \otimes \dots \otimes h_K. \quad (36)$$

We can easily see that  $S_{K_1, n_1, \dots, n_{K_1}}(i) = 0$  for  $i > \frac{\sum_{q=1}^{K_1} n_q}{2}$  due to the fact that the optimal residual sequence cannot have more than  $\frac{\sum_{q=1}^{K_1} n_q}{2}$  ones. Also, from (34) we note that only  $\frac{1}{2^K} \prod_{q=1}^K n_q$  terms have to be added when all the convolution sums in (36) are evaluated.

Table 6: The best 18 triplets of genes for predicting the class label according to the NML model for the class  $\mathcal{M}(\theta, 3, f)$ .

Codelength	Classification error [%]	Triplet of Genes			Gene accession numbers		
6.9	0.912	1834	2288	5714	M23197	M84526	HG1496-HT1496
<b>7.9</b>	<b>0.010</b>	<b>1834</b>	<b>3631</b>	<b>6277</b>	<b>M23197</b>	<b>U70063</b>	<b>M30703</b>
7.9	0.891	758	4250	4342	D88270	X53586	X59871
8.0	0.652	2288	4847	6376	M84526	X95735	M83652
<b>8.7</b>	<b>0.008</b>	<b>1834</b>	<b>3631</b>	<b>5373</b>	<b>M23197</b>	<b>U70063</b>	<b>S76638</b>
<b>8.7</b>	<b>0.007</b>	<b>1834</b>	<b>3631</b>	<b>6279</b>	<b>M23197</b>	<b>U70063</b>	<b>X97748</b>
8.7	0.910	1144	1217	1882	J05243	L06132	M27891
8.8	0.649	302	2288	6376	D25328	M84526	M83652
8.8	0.055	1144	1834	1882	J05243	M23197	M27891
8.8	0.063	1834	1882	6049	M23197	M27891	U89922
<b>8.8</b>	<b>0.004</b>	<b>1144</b>	<b>1882</b>	<b>5808</b>	<b>J05243</b>	<b>M27891</b>	<b>HG2981-HT3127</b>
8.8	0.584	2288	3932	6376	M84526	U90549	M83652
8.9	0.558	2288	5518	6376	M84526	X95808	M83652
8.9	0.560	1399	2288	6376	L21936	M84526	M83652
8.9	0.620	1241	2288	6376	L07758	M84526	M83652
8.9	0.605	2288	3660	6376	M84526	U72342	M83652
8.9	0.582	2288	4399	6376	M84526	X63753	M83652
8.9	0.556	2288	4424	6376	M84526	X65867	M83652

### 4.3 An experiment from cancer genomics

We illustrate the classification procedure based on the NML model for classes of Boolean regression models for the microarray DNA data Leukemia (ALL/AML) of [4], publicly available at <http://www-genome.wi.mit.edu/MPR/>. The microarray contains 6817 human genes, sampled from 72 cases of cancer, of which 47 are of ALL type and 25 of AML type. The data is pre-processed as recommended in [4] and [2]. The resulting data matrix  $\tilde{X}$  has 3571 rows and 72 columns.

We design a two level quantizer by the LBG algorithm [9], and the decision threshold results at 2.6455. All the entries in the matrix  $\tilde{X}$  are used as a training set but we note that no information about the true classes is used during the quantization stage. The entries in the matrix  $\tilde{X}$  are quantized to binary values, which results in the binary matrix  $X$ .

#### 4.3.1 Extending the classification for unseen cases of the Boolean regressors

The Boolean regressors observed in the training set may not span all the  $2^k$  possible binary vectors. If the binary vector  $\underline{b}_q$  is not observed in the training set, the classification decision  $f^*(\underline{b}_q)$  remains undecided during the training stage. We select the value of  $f^*(\underline{b}_q)$  by use of the nearest neighbor voting, which amounts to the decision by the majority vote of the neighbors  $\underline{b}_\ell$  situated at Hamming distance 1, for which  $f^*(\underline{b}_\ell)$  was decided during the training stage. If after voting there still is a tie we take the majority vote of the neighbors at Hamming distance 2, and continue if necessary until a clear decision is reached.

#### 4.3.2 Estimation of classification errors achieved with Boolean regression models with $k = 3$

The Leukemia data set was considered recently in a study for comparing several classification methods [2]. The evaluation of the performance is there based on the classification error as estimated in a crossvalidation 2:1 experiment. In order to compare our classification results with the results in [2], we estimate the classification error in the same way, namely, by dividing at random the 72 patient set into a training set of  $n_T = 48$  patients and a test set of  $n_s = 24$  patients, finding the optimal predictor  $f^*(\cdot)$  over the training set, classifying the test set by use of the predictor  $f^*(\cdot)$  (the extension for the unseen cases in the training set is done as in Section 4.3.1), and counting the number of classification errors produced in the test set. The random split is repeated  $n_r = 10000$  times, and the estimated classification error is computed

as the percentage of the total number of errors observed in the  $(n_r \cdot n_s)$  test classifications. For comparison, we mention that the best classification methods tested in [2] have classification errors higher than 1%. As we can observe in Table 6 there are several predictors with three genes, achieving classification rates as low as 0.004%. We note a remarkable consensus in ranking of the gene triplets, according to the NML code length and the estimated classification error rates.

As to the genes involved in the optimal predictors in Table 6, we note that five genes belong to the set of 50 “informative” genes selected in [4], namely  $M23197$ ,  $M84526$ ,  $M27891$ ,  $M83652$ ,  $X95735$ .

## 5 Summary

The Boolean regression classes of models are powerful modeling tools with the associated NML models, which can be easily computed and used in MDL inference, in particular for factor selection.

The MDL principle for classification with the class of Boolean models provides an effective classification method, as seen in the important application of cancer classification based on gene expression data. The NML model for the class  $\mathcal{M}(\theta, k, f)$  was used for the selection of informative feature genes. With the sets of feature genes selected by the NML model we achieved classification error rates significantly lower than those reported recently for the same data set.

## Appendix

In the Table 7 we present the data of a toy experiment from [8]. Note that the table is transposed according to the convention used in text for  $x_{ij}$ .

## References

- [1] A. Barron, J. Rissanen, Y. Bin. The minimum description length principle in coding and modeling. *IEEE Trans. on Information Theory*, vol.44, no. 6, 2743–2760, Oct. 1998.
- [2] S. Dudoit, J. Fridlyand, T. P. Speed. Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Dept. of Statistics University of California, Berkeley, Technical Report 576, 2000.

RCH1	BCL3	FRA1	REL-B	ATF3	IAP-1	PC-1	MBP-1	SSAT	MDM2	p21	p53	AHA
-1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	0	1	0	0	0	0	1	1	1	1
-1	0	0	1	1	0	1	0	0	1	1	1	1
0	0	1	0	1	0	0	0	0	0	1	1	1
-1	0	0	1	1	1	1	1	0	1	1	1	1
0	0	0	0	1	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	1	0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1	1	1
-1	0	1	1	0	0	0	0	0	1	1	1	0
0	0	1	0	1	0	0	0	0	1	1	1	1
0	0	1	1	1	0	0	0	0	1	1	1	1
0	1	0	1	1	1	1	0	0	1	1	1	1
0	0	0	0	1	0	0	0	0	0	1	1	1
0	0	0	0	1	0	0	0	0	0	1	1	1
-1	1	1	1	1	0	1	0	0	0	0	-1	-1
0	0	0	0	1	0	0	0	0	0	0	-1	0
-1	1	0	1	1	0	1	0	1	0	1	-1	-1
0	0	1	0	1	0	0	0	0	1	1	-1	0
0	0	0	0	0	0	0	0	0	0	0	-1	0
0	0	0	0	1	0	0	0	0	0	0	-1	0
0	0	0	1	0	0	1	0	0	0	0	-1	0
0	0	0	0	1	0	0	0	0	0	1	-1	0
0	0	0	0	1	0	1	0	0	0	1	-1	0
-1	1	0	1	0	1	1	0	0	0	0	-1	-1
-1	0	0	0	1	0	0	0	0	0	1	-1	-1
-1	0	0	0	1	0	0	0	0	0	1	-1	-1
0	0	0	1	0	0	0	0	0	0	1	-1	0
0	0	0	0	1	0	0	0	0	0	1	-1	0
0	0	0	0	1	0	0	0	0	0	1	-1	0

Table 7: A small scale experiment where gene expressions are quantized to ternary values. The columns represent different genes, while the rows represent the various conditions (patients).

- [3] C.D. Giurcaneanu, I. Tabus, J. Astola. Clustering time series gene expression data based on sum-of-exponentials fitting. *Journal of Applied Signal Processing*, Volume 2005, No. 8, 2005.
- [4] T.R. Golub, D.K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J.P. Mesirov, H. Coller, M.L. Loh, J.R. Downing, M.A. Caligiuri, C.D. Bloomfield, E.S. Lander. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. *Science*, Vol. 286, pp. 531-537, Oct. 1999.
- [5] F. Jacob, J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, Vol. 3, 318-356, 1961.
- [6] S.A. Kauffman. *Origins of Order: Self-Organization and Selection on Evolution*. Oxford University Press, Oxford, 1993.
- [7] S. Kim, E.R. Dougherty. Coefficient of determination in nonlinear signal processing. *Signal Processing*, 80:2219–2235, 2000.
- [8] S. Kim, E.R. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J.M. Trent, M. Bitnner. Multivariate measurement of gene expression relationships. *Genomics*, 67, 201–209, 2000.
- [9] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantization design. *IEEE Transactions on Communications*, 28:84–95, Jan. 1980.
- [10] C. Mircean , I. Shmulevich , D. Cogdell , W. Choi , Y. Jia , I. Tabus , S.R. Hamilton, W. Zhang. Robust estimation of protein expression ratios with lysate microarray technology. *Bioinformatics*, Advance Access published on January 12, 2005, DOI 10.1093/bioinformatics/bti258, 2005.
- [11] C. Mircean, I. Tabus, T. Kobayashi, M. Yamaguchi, H. Shiku, I. Shmulevich, W. Zhang. Pathway analysis of informative genes from microarray data reveals that metabolism and signal transduction genes distinguish different subtypes of lymphomas. *International Journal of Oncology*, 24(3):497-504, 2004.
- [12] J. Rissanen. Modelling by shortest data description. *Automatica*, vol. 14, 465–471, 1978.
- [13] J. Rissanen. Universal coding, information, prediction and estimation. *IEEE Trans. on Information Theory*, vol.30, 629–636, Jul. 1984.

- [14] J. Rissanen. Stochastic complexity and modelling. *Ann. Statist.*, vol. 14, pp. 1080-1100, 1986.
- [15] J. Rissanen. MDL Denoising. *IEEE Trans. on Information Theory*, vol. IT-46, No. 7, 2537–2543, Nov. 2000.
- [16] J. Rissanen. Strong optimality of the normalized ML models as universal codes and information in data. *IEEE Trans. on Information Theory*, vol.IT-47, No. 5, 1712–1717, July 2001.
- [17] Y.M. Shtarkov. Universal sequential coding of single messages. Translated from *Problems of Information Transmission*, Vol. 23, No. 3, 3–17, July-September 1987.
- [18] I. Tabus and J. Astola. On the use of MDL principle in gene expression prediction. *Journal of Applied Signal Processing*, Volume 2001, No. 4, December 2001.
- [19] I. Tabus, J. Rissanen, J. Astola. A classifier based on normalized maximum likelihood model for classes of Boolean regression models. *Proceedings of EUSIPCO 2002, XI European Signal Processing Conference*, September 3-6, 2002, Toulouse, France, Vol.1, pp. 119-122, 2002.
- [20] I. Tabus, J. Rissanen and J. Astola. Classification and feature gene selection using the normalized maximum likelihood model for discrete regression. *Signal Processing, Special issue on Genomic Signal Processing*, Vol. 83, No.4, pp. 713-727, April, 2003.
- [21] I. Tabus, C. Mircean, W. Zhang, I. Shmulevich, J. Astola. Transcriptome-based glioma classification using informative gene set. In "*Genomic and molecular neuro-oncology*" (W. Zhang and G. Fuller, eds), Jones and Bartlett Publishers, pp. 205-220, 2003.
- [22] I. Tabus, J. Astola. Clustering the non-uniformly sampled time series of gene expression data. *Proceedings of ISSPA 2003, International Symposium on Signal Processing and Applications*, Paris, July 2-5, p. 61–64, 2003.
- [23] I. Tăbuș, C.D. Giurcăneanu, and J. Astola. Genetic networks inferred from time series of gene expression data. In *ISCCSP 2004, First International Symposium on Control, Communications and Signal Processing*, pages 755–758, Hammamet, Tunisia, Mar. 21-24 2004.

- [24] Y. Yamamoto and M. Mukaidono. Meaningful special classes of ternary logic functions—Regular ternary logic functions and ternary majority functions. *IEEE Trans. Computers*, Vol.37,no.7,799–806, July 1988.